Distributed tasking in ontology mediated integration of typological databases for linguistic research

Adam Saulwick[1], Menzo Windhouwer[1], Alexis Dimitriadis[2] and Rob Goedemans[3]

[1] Department of Theoretical Linguistics, University of Amsterdam
Spuistraat 210, NL-1012 VT Amsterdam, The Netherlands
{a.g.saulwick, m.a.windouwer}@uva.nl
[2] Utrecht Institute of Linguistics, OTS
Trans 10, 3512 JK Utrecht, The Netherlands
alexis.dimitriadis@let.uu.nl
[3] Phonetics Lab, PO Box 9515, 2300 RA Leiden, The Netherlands
R.W.N.Goedemans@let.leidenuniv.nl

Abstract. Developing a unified interface for typological databases with diverse theoretical perspectives of certain crosslinguistic phenomena poses problems for ontology-mediated integration, where multiple potentially theory-specific concepts need to be associated. This paper outlines a bidirectional approach to unbiased domain conceptualization and unification of database notions. At the global level, we describe the use of ontology for effective domain specification. At the local level we are concerned with an appropriate unification description, called Data Transformation Language (DTL), containing rules for normalization of values and the addition of semantic structures. The complete approach is justified and illustrated with a partial case study taken from the Typological Database System (TDS) project.[1]

1 Introduction

It seems to be a part of human nature that people form diverse viewpoints of phenomena in the world. Scientific databases reflect this by modelling an observed phenomenon in different ways. The possibility of doing crossdatabase searches can facilitate greater access to empirical data for the testing of hypotheses. However, the integration of diverse models is not a trivial task. For integration to be coherent, not only do syntax and structure need alignment, but also the precise semantic relationships between modelled concepts need to be made explicit.

In this paper we describe an approach aimed at the construction of a domain ontology, which links diverse theoretical models. The ontology grows out of a

[1] TDS gratefully acknowledges the financial support of the Netherlands Organization for Scientific Research (NWO). The TDS project is being carried out by a research group of the Netherlands Graduate School of Linguistics (LOT), with members representing the University of Amsterdam, Leiden University, Radboud University Nijmegen, and Utrecht University.

bidirectional approach to domain modelling. First, component database models are inspected for key concepts of the subdomain. This is done by the construction of a local specification which encapsulates the database's theoretical bias by transforming or grouping fields. This enables identification of and mapping to general domain concepts within the subdomain. Second, these subdomains are incorporated into a meaningful network, the global domain ontology. These local and global specifications each assume a distinct role in the task of unification. We call this "distributed tasking" and justify and illustrate it with a partial case study taken from the Typological Database System (TDS) project, briefly outlined in §2.

2 Typological Database System

The Typological Database System[2] is a web-based service (currently in development) hosting an integrated ontology for unified querying of multiple independently developed typological databases.

In very general terms, linguistic typology studies structural and semantic variation across languages of the world (sometimes as distinct from variation found within linguo-genetically affiliated languages), and then strives to categorize and hypothesize over linguistic types on the bases of empirically observed linguistic phenomena. This kind of research involves the collection of information about linguistic phenomena from a representative sample of the world's languages.

Researchers contributing to the TDS project have (on the whole) collected source information from published material (such as grammars) on particular languages and stored this in digital form (see Table 1 for those currently integrated). The purpose of the TDS project is to make these diverse typological databases available through a unified interface and to enable sophisticated searches across database boundaries.

Table 1. Component databases currently integrated

Database name and reference to source data	TDS appellation
StressTyp [4]	D-StressTyp
Person Agreement Database [1]	D-Agreement
Typological Database Nijmegen [16]	D-TDN
SCALA/Spinoza database [11]	D-SCALA
TDA Parts-of-speech systems [7]	D-TDA

The creator of each source database may have a particular theoretical perspective—implicitly or explicitly—encoded in a database. This will be called the *theoretical bias*. For instance, the use of specific terminology, or the choice of features encoded in a database, may reflect the creator's theoretical bias.

It is a tenet of the TDS project to respect the theoretical bias of component databases, without imposing another layer of interpretation. Further, it is felt that reinterpretation of data should be left open to the end-user. In terms of the design of the system this leads to at least two consequences. First, it is important to provide rich metadata on the provenance of information so that the end-user can take into account

[2] See http://languagelink.let.uu.nl/tds/ for the current online version.

its origin. Second, specification of the domain in the ontology should allow for, or encompass, variant and theoretically biased terminology. The ontology provides an overt specification of the domain by naming linguistic concepts, displaying them in a network, and giving each a description. Thus, TDS values current recommendations for semantic integration [5](p.38ff) [6], namely: clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment.

Moreover, the stated model for specifying the domain provides a means of intuitive querying. Stuckenschmidt and van Harmelen [17] state: 'Using an ontology as a query model has the advantage that the structure of the query model should be more intuitive for the user because it corresponds more to the user's appreciation of the domain' (p.35).

The goal of providing a single interface for unified querying has required tackling problems of data mapping across linguistic subdomains, each with theory-specific terminology. For instance, information on key concepts in component databases may need to be displayed in the user interface (UI). It may constitute the only information on a database field or it may be displayed together with a concept in the ontology. The conditions on mapping database fields to concepts in the ontology are discussed in §4 below. Without an ontology integration of multiple databases would result in no or little domain coherence.

Having identified the need for ontology mediated semantic integration, we now discuss the construction of the domain ontology.

3 Domain ontology

The TDS uses an integrated ontology to describe linguistic (and related) concepts, within a meaningful network and enable mapping of database fields to searchable notions in the user-interface. The ontology is built using the Web Ontology Language (OWL) [12]. The motivation for the use of OWL is multi-faceted, and includes the wish to:

i. facilitate integration at a technical level;
ii. permit compatibility with web standards for accessibility and internationalization;
iii. be open and extensible; and
iv. allow for the possibility to dock into other relevant ontologies (such as SUMO [13] and GOLD [3]).[3]

3.1 Conceptualization of the domain

The 'domain' is the subfield of linguistics, namely linguistic typology, as further specified by the vocabulary used by the creators of the participating electronic databases. As with many other fields of linguistics, this subfield is not immune to

[3] Note that points (ii) and (iii) are stated capabilities of OWL, see http://www.w3c.org/2004/OWL/.

divergence or disagreement over the application of terminology. Conceptualization of the domain as expressed in the ontology is formulated to encompass the diverse theoretical perspectives represented by the individual nomenclatorial practice of linguistic phenomena in component databases. Thus, an important principle of depicting the domain is to provide explicit and clear descriptions of each contributor's intended interpretation of particular linguistic concepts.

The aim of the ontology is emphatically *not* an attempt to prescribe crosslinguistically valid definitions of linguistic notions.[4]

3.2 Unification of concepts

As stated above, concepts in the ontology describe notions resident in the databases. Theoretically biased perspectives on subdomains use specific vocabulary which may not be (universally) accepted or adhered to by other members of the (sub)domain community. A case in point is word order phenomena, which are variously represented in multiple databases. In this instance, "word order phenomena" refers to variety in the linear order of certain (basic) words in sentences. For instance, in one database (D-TDN) Basic Word Order (BWO) variation is discussed in terms of the patterns: SOV, SVO, VSO, VOS, OVS, (that is, specification of certain linear orderings of the linguistic entities labelled 'subject', 'object', 'verb', commonly abbreviated to S, O & V). In contrast, in another database (D-TDA) BWO is discussed in terms of 'predicate initial', 'predicate medial' or 'predicate final'. (For explanation of the terms 'BWO' and 'predicate' see [15] and [8] respectively.) These differences in terminology highlight two issues for the conceptualization of the domain. The first issue concerns the intended meaning of the terms. The second issue concerns the nature of the correspondence between them.

TDS handles these issues by adopting what has been called a 'hybrid approach' to information sharing [17] (p.32ff). In this approach 'the semantics of each source is described by its own ontology' [17] (p.33). In the TDS architecture, although source descriptions contain semantic information we refer to them as *annotated schemata* rather than local ontologies. This is because we wish to differentiate between the complex semantic network of the TDS's global ontology [17] 'shared vocabulary' (p.34)) and the relatively structurally shallow local descriptions.

In accord with the hybrid approach to information sharing, the intensional meaning of notions in each database is specified at a local level, whereas unification of related notions across databases is carried out at a global level using the shared vocabulary in the ontology. We now detail the precise nature of how we handle some aspects of hybrid information sharing.

For expository and integration purposes we give different names to ideas depending on the level at which they occur. The three terms are *notion*, *derived notion* and *concept*. *Notion* refers to an idea named and described of a database field or value. *Derived notion (dNotion)* refers to an idea which is not described of a database

[4] No claim is made as to the validity or indeed possible success of such a venture. Moreover, it is seen as irrelevant for the current project, which simply aims to present the domain in a discipline-coherent manner.

but which is derived through the unification of multiple fields (either within a single, or across multiple, databases). Note that the derivation of a notion involves some form of semantic enrichment of the source data.[5] The term *concept* refers to a (generally linguistic) idea named in the ontology regardless of whether it is named in the local annotated schema.

Part of the TDS architecture provides a means for mapping between notions, derived notions and concepts. Thus, we can identify a clear division of labour between its components, namely:

i. It is the task of local annotated schema to provide explicit descriptions of the intensional meaning of both underived and derived database notions.

ii. It is the task of the ontology to provide explicit descriptions of the intensional meaning of linguistic concepts as relevant to notions in the component databases.

Examples of each type of domain idea (*Notion*, *dNotion* and *Concept*) are given in Table 2.

Table 2. Nomenclature of ideas at each level of semantic integration

Type	Source information →	notions in local schemata →	concepts in the ontology
1	Agreement Marker	*Underived:* Agreement Marker	Agreement Marker
2	Rhythm weight plus Stress weight	*Derived:* Weightful	Weightful
3	BWO clause	BWO clause	predicate-based word order

Having differentiated database notions from ontology concepts and outlined a situation where unification of notions is possible, we give some strategies for performing crossdatabase ontology mediated unification. As an example of semantic unification we take an instance of Type 3 from Table 2. The ontology identifies a concept called *core constituent word order*. This is a hypernym, i.e. subsumes two subconcepts *basic word order* and *predicate-based word order*. In the ontology both of these concepts each contain daughter nodes which map to fields in two separate databases (as stated in §3.2 above). A representation of this section of the ontology hierarchy is Figure 1.

[5] Note that we do not discuss the precise nature of semantic enrichment here as it is beyond the scope of the present discussion.

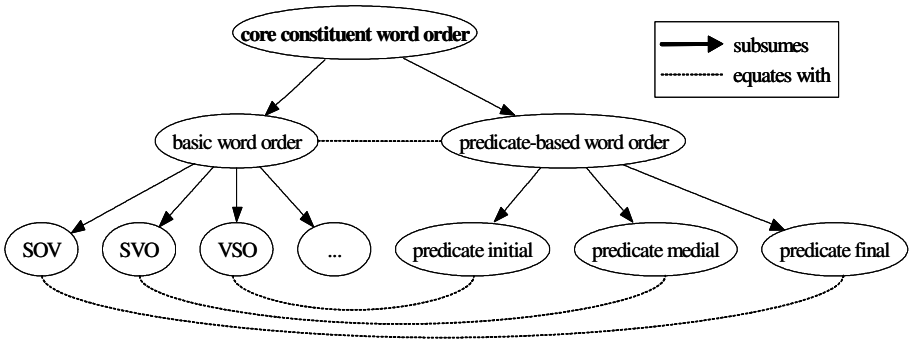

Figure 1. Section of ontology hierarchy showing the unifying concept in bold

There are two types of relationships between concepts depicted in this figure. Vertical arrows indicate the relationship of hypernym to hyponym, where the arrow originates from the semantically more general term and points to the more specific term. Horizontal lines connect concepts that have been coded in the ontology as having some degree of semantic equivalence in the domain.

For researchers interested in the general phenomena of word order of core constituents, the ability to perform a single query at the concept level *core constituent word order*, which incorporates both perspectives (encapsulated by the concepts *BWO* and *predicate-based word order*), will result in a greater number of data matches and thus output richer data than a standard query performed at a lower level in the ontology. This represents an important means by which 'unification' of diverse terminology is accomplished while at the same time maintaining unbiased domain specification. Terms in source databases remain unchanged. The term 'unification' here is intended to represent the process of structural integration of related concepts within the ontology. It is important to point out that unification should not be seen as an attempt to gloss over differences in the intensional meaning of terminology but rather a means of hypernymic grouping or synonym association. A further advantage of this approach to semantic integration is that no information is lost from either of the sources. Rather, diverse source information and nomenclature is presented to the user. In fact, as will be shown in §5 below, queries can result in increased access to information.

Through unification just outlined, the TDS seeks to permit unbiased yet explicit domain conceptualization and provide end-users with access to a greater number of database correspondences. Thus, the ontology is not merely a listing of isolated concepts, but a meaningful conceptualization of the domain *and* a means through which unification is mediated. It provides domain-specific knowledge in the form of an annotated hierarchy of concepts and relations which are mapped to notions and linked to database fields, thereby furnishing the end-user with a structured semantic network. On a purely practical level, this can be used as a navigational tool with embedded domain-specific explanatory matter. The end-user can traverse the ontology with the aid of vocabulary associated with alternate theoretical stand-points and call on typological data mapped to nodes representing linguistic concepts at various degrees of granularity.

Having outlined the domain ontology, we now describe the specification of the local annotated schemata.

4 Data Transformation Language (DTL)

The domain ontology described in the previous section lays the basis for semantic integration of the source databases. However, these databases not only show semantic differences, but also great variety in structure. This is a well known and extensively studied problem in the world of data warehousing [see for example overviews in 9, 10, 14]. The approach outlined in this section differs in the fact that the normalization process of the source data is also taken as an opportunity to add additional, and semantic-laden, structure.

The restructuring, or transformation, process needed by the imported data is described in a language we call the Data Transformation Language (DTL); an annotated schema language. It is used to describe the desired (hierarchical) data structure, and to annotate those structures with descriptions of their semantic meaning. DTL functions as an intermediary between source databases and the domain ontology. In addition, the DTL specifies how instantiations of this schema are built from the source data, *i.e.* the data transformed. This language is formulated to allow non-programmers to describe mappings, and to abstract away from the low-level physical details of the data format and in- and output handling.

Table 3 shows a description of the types of mappings expressed in the DTL, along the lines of the distinction between underived and derived notions described in §3.2 above.

Table 3. DTL mapping types

Type	Mapping	Source information →	notions in DTL →		Ontology concepts
Underived notions	Direct	Db_1:$field_1$[=$condition_1$]	DB_1:$Notion_a$		$Concept_a$
			DB_1:$Notion_b$		–
	Indirect	Db_1:$field_1$[=$condition_1$] and/or Db_1:$field_n$[=$condition_n$] [and/or …]	DB_1:$Notion_a$		$Concept_a$
			DB_1:$Notion_b$		–
Derived notions	Local semantic enrichment		Db_1: [d]$Notion_1$[=$condition_1$] [and/or Db_1:[d]$Notion_n$[=$condition_n$] [and/or …]]	DB_1:$dNotion_a$	$Concept_a$
				DB_1:$dNotion_b$	–
	Global semantic enrichment		Db_1:[d]$Notion_1$[=$condition_1$] and/or Db_j:[d]$Notion_n$[=$condition_n$] [and/or …]	TDS:$dNotion_a$	$Concept_a$
				TDS:$dNotion_b$	–

Legend: [x] = optional, … = repetition, x/y = x or y, subscripts indicate a specific instantiation, – = no related concept.

We now give an example of a DTL specification. Figure 2 shows a section of DTL for transforming one of the source databases (D-TDN) for integration into TDS. This example is used in the following paragraphs to illustrate various types of mappings.

```
1.  IMPORT "TDS.dtl";
2.
3.  DECLARE tdn="http://languagelink.let.uu.nl/tds/ns/TDS/D-TDN";
4.
5.  MAP {
6.    False   FOR 0;
7.    True    FOR 1;
8.    MISSING FOR 9;
9.    NULL    FOR 99;
10. }
11.
12. MAP code (code,name) {
13.   IMPORT MAP SIL (code);
14.   "x-obsolete-sil-MEX" FOR ("MEX","Malagasy");
15.   "x-tds-031"          FOR ("YOV","Yokuts");
16.   OTHERWISE ERROR "No unique language code available";
17. }
18.
19. NOTION language LOOKUP MAP code (ethnologue_code,language_name)
    GROUPS {
20.   NOTION name   IS language_name USE MAP text;
21.   NOTION tdn:id IS language_ID   USE MAP text MEANS "TDN language
    ID";
22.
23.   NOTION BWO IS {
24.    "SOV"     MAP AS CONCEPT SOV                      FOR v146 = 1;
25.    "SVO"     MAP AS CONCEPT SVO                      FOR v147 = 1;
26.    "VSO"     MAP AS CONCEPT VSO                      FOR v148 = 1;
27.    "VOS"     MAP AS CONCEPT VOS                      FOR v149 = 1;
28.    "OVS"     MAP AS CONCEPT OVS                      FOR v150 = 1;
29.    "N - GEN" MAP AS CONCEPT nounGenitiveWordOrderNGEN FOR v151 = 1;
30.    "GEN - N" MAP AS CONCEPT genitiveNounWordOrderNGEN FOR v152 = 1;
31.    CLASH ERROR "enumeration conflict for BWO value"
32.   }
33.
34.   NOTION tdn:VerbalMorphology GROUPS {
35.     NOTION SubjectAgreement GROUPS {
36.       NOTION tdn:SubjectFlectionVerb
37.         MAP TO CONCEPT agreementMarkerOnVerb
38.         MEANS "subject flection on verb"
39.         IS v456;
40.       NOTION tdn:SubjectMarkerPossessivePronoun
41.         MEANS "subject marker = possessive pronoun"
42.         IS  v460;
43.     }
44.     NOTION ObjectAgreement GROUPS {
45.       NOTION tdn:ObjectAgreementOnVerb
46.         MAP TO CONCEPT agreementMarkerOnVerb
47.         MEANS "object agreement on verb"
48.         IS v458;
49.       NOTION tdn:ObjectAgreementOnCopula
50.         MAP TO CONCEPT agreementMarkerOnCopula
51.         MEANS "object agreement on copula"
52.         IS v459;
53.     }
54.   }
55. }
```

Figure 2. D-TDN DTL specification

The DTL specification starts with a general preamble: the import of the generic TDS DTL description and the declaration of the database-specific D-TDN namespace. In lines 5 to 10 the default value mapping is defined. In the D-TDN database most variables can take one of the values: 0, 1, 9 and 99. These are all translated into TDS standard values: booleans and some special values. By making a distinction between a `MISSING` and a `NULL` value TDS can show incomplete instantiations as possible results of a query, because the `MISSING` explicitly indicates a possible value. Naturally, this feature relies on the database developer to have already made the distinction between missing values and no or irrelevant values, *i.e.* the `NULL` value should not be overloaded with several meanings [2]. A second map (lines 12 to 17), named `code`, is defined next. This is used to map D-TDN primary keys into global TDS primary keys. The map is partially filled by importing an existing map, *i.e.* SIL[6] (defined in TDS.dtl). When no mapping can be found an error is raised.

From line 19 onwards the DTL statements start transforming the universal D-TDN table into a hierarchical structure of notions. A DTL notion may have a one-to-one relationship with a concept from the ontology, *i.e.* they share a vocabulary. Notions without such a relationship exist specifically to group other notions together. The mappings start with the root notion `language` and concept *language*. The `LOOKUP` operation uses the `code` map to find the unique language id, based on the `ethnologue_code` and `language_name` fields from D-TDN. Note in Table 3 that all following types of mappings may be subject to selection restrictions in the form of a condition on a database field or notion (see lines 24 to 30).

Direct mappings. The following two statements are examples of direct mappings from the database to underived notions in DTL. Underived DTL notions can be created through a direct mapping of a database field to a notion. The mappings of `language_ID` to `tdn:id` is an example of a direct mapping. As demonstrated in this instance, notions that do not correspond to a concept from the ontology are annotated with a description of their meaning for display to the end-user.

Indirect mappings. Notions can also be the result of an indirect mapping. In such cases multiple fields of one database are combined as one notion. On line 23 the notion `BWO` is defined, which has a one-to-one correspondence with a concept in the ontology. This notion derives its value from several D-TDN fields. Each of its values is also related to an ontology concept, *e.g.* the value "SOV" for the notion `BWO` relates to the concept *SOV* (this is also an exact one-to-one relationship as is indicated by the use of `MAP AS CONCEPT`). Only one value is permitted, so when more then one of the D-TDN fields `v146` to `v152` has value 1 an error is raised to indicate an inconsistency in the source database.

From line 27 on, a deeper structure is built. This groups several variables on verbal morphology, which are once more grouped into notions named `SubjectAgreement` and `ObjectAgreement` (both known in the ontology). The database fields mapped in this case do not correspond one-to-one to a concept in the ontology, but can still be mapped to related concepts. This distinction is indicated by the use of `MAP TO CONCEPT` instead of `MAP AS CONCEPT`.

---

[6] This map contains codes for each language; taken from Ethnologue http://www.ethnologue.org/ (where these exist), otherwise alternate codes are assigned in the DTL (see lines 14 and 15).

Semantic enrichment. Derived notions (shown as *dNotions* in Table 3) are formed from the amalgamation of component database fields. They do not exist as "mappable" notions in the source data, but are relevant for the subdomain. We see this as a form of semantic enrichment. This can happen at two distinct levels. The first level is local to a database. In this case notions from one database are used to derive a new notion. The second level on which semantic enrichment can take place is the global one. This allows new notions to be created on the basis of (locally or globally derived) notions from several databases. Lines 34 to 54 show how the D-TDN DTL uses locally derived notions for the introduction of hierarchical grouping, like `SubjectAgreement` and `ObjectAgreement`. Another example of local semantic enrichment is taken from the D-StressTyp DTL and is shown in Figure 3.

```
1.  NOTION weightful IS {
2.    true FOR stress:Rhythm_weight=true OR stress:Stress_weight=true;
3.  }
```

Figure 3. Section of D-StressTyp DTL specification

The D-StressTyp database describes stress phenomena in the languages of the world. Although it contains information on whether the rules for the placement of primary or secondary stress (rhythm) are sensitive to syllabic weight, it does not contain a single value giving information on whether a stress system as a whole can be considered "weightful". The DTL statement in Figure 3 derives this new notion from the previously created D-StressTyp notions, and thus enriches the local D-StressTyp structure. The enrichment process can be applied at a global TDS level for creating crossdatabase derived notions.

Adding a new database to the TDS requires the construction of a new DTL specification. The entry cost of this can be reduced by generating a bare-bones specification, *e.g.* creating underived notions by analyzing the fields and their instantiations occurring in the database. The addition of extra indirect mappings, derived notions and the identification of related concepts is the task of a knowledge engineer with in-depth knowledge about both the database and the global domain ontology.

Although this makes the integration of each new data source in the DTL approach mostly a manual, and thus, labour-intensive process, it results in a set of reliable mappings. We believe this approach is suited to the integration of certain database types, such as in the case exemplified here of TDS where component databases are of high value but relatively low volume.

Having described semantic and structural integration for both database models and data, we now outline the architecture for its implementation.

5 TDS System Architecture

As outlined in §2, the TDS project's major goal is to implement a system which will enable unified querying of multiple typological databases. The architecture of this special-purpose database system is shown in Figure 4 below.

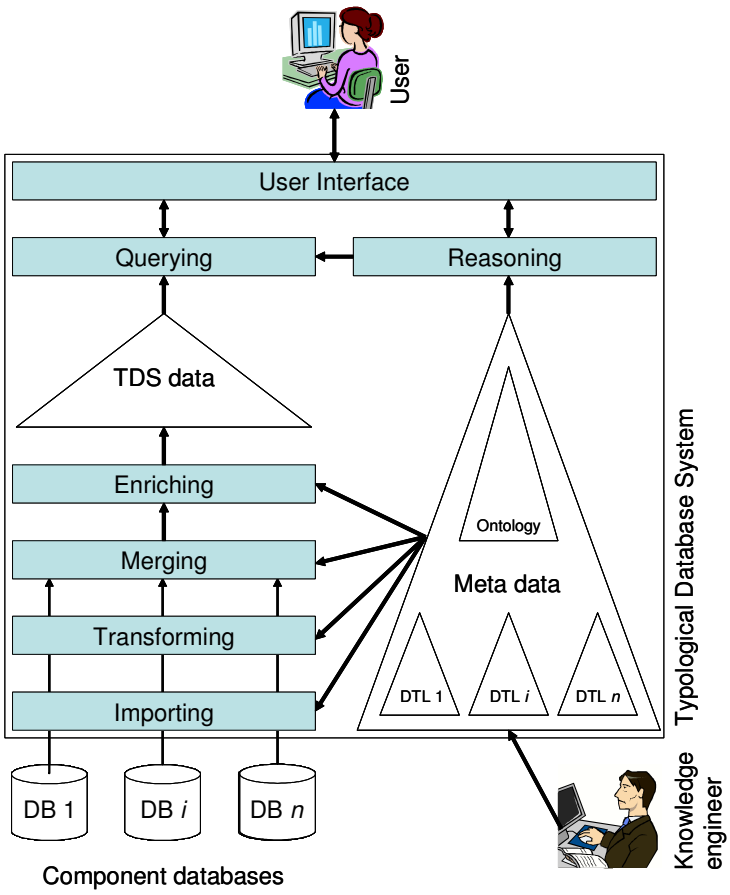

Figure 4. Typological Database System architecture

Various component databases provide the typological data for import into the system. The TDS view of their data and the flow of it through the system are controlled by the metadata. This means that when a new component database is added (or a known one has substantially changed), a knowledge engineer with extensive understanding of the new database needs to make a DTL specification to enable incorporation into the TDS database.

An import step allows the system to load data from the component databases. An import plug-in accesses the native storage structure of the component database, *e.g.* a Microsoft Access database or a set of flat CSV files. The result of the import step is an XML document containing a basic dump of the database.

This basic XML dump is transformed into a TDS normalized structure by the DTL processor. As shown in line 19 of the DTL example in Figure 2, this process also makes sure that local primary keys are translated into global primary keys. This allows the merging of the component data into one TDS data structure. All notions are

available for manipulation and global semantic enrichment can take place. This is specified in the DTL. The result is an XML document, now containing all the data in a standardized structure which is accessible from the TDS user interface.

The web interface allows the user to construct a query in various ways, *e.g.* by filling out forms on specific typological topics or by navigating the ontology in various ways. The ontology navigator allows the user to traverse the concept hierarchy and see which database notions are linked to a specific concept, as specified in the DTL. Multiple concept queries, possibly spanning data originally from multiple component databases, can easily be constructed. Queries are answered on the basis of the global TDS XML document. A reasoning component interprets the metadata and makes the embedded relationships available in a format easily interpretable by the user or other system components.

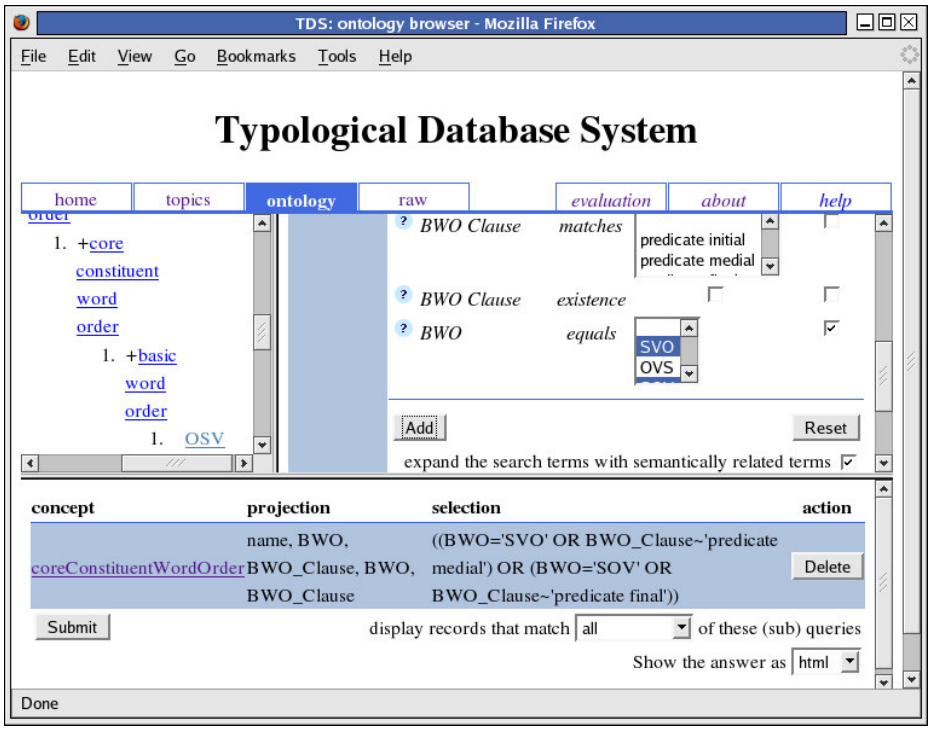

Figure 5. User interface showing a query on BWO

Figures 5 and 6 show a query and answer session. In the query session, the user selected the concept *core constituent word order* from the ontology browser (the left-hand frame of the web page shown in Figure 5). The right-hand frame displays a description of the concept (not shown) plus any database mappings directly associated with it. An additional option (also not shown, but active) aggregates and shows all mappings related to descendant concepts. One of the aggregated mappings shown on the right-hand frame is the notion `BWO` (described in §4). The interface allows the user to select one or more of the values assigned to this notion as a query criterion (here

SVO & SOV are selected). This is added to the collection of selection criteria and notions selected for projection at the bottom frame.

Through ontology mediated integration it is possible to request a search of semantically related terms. In this example, the user chose to expand the condition on the notion `BWO`. This resulted in additional search criteria, namely on the notion `BWO_clause`. TDS found these extra criteria by searching the ontology for equivalence relationships between concepts, as illustrated in Figure 1. The answer page shown in Figure 6 illustrates the value of this expansion. For example the language Alamblak only has a value in the related notion `BWO_Clause` (and its source database D-TDA), and would thus never have been found with a query over `BWO` only. Crucially, this additional information would not have been furnished to the end-user without the ontology mediated integration.

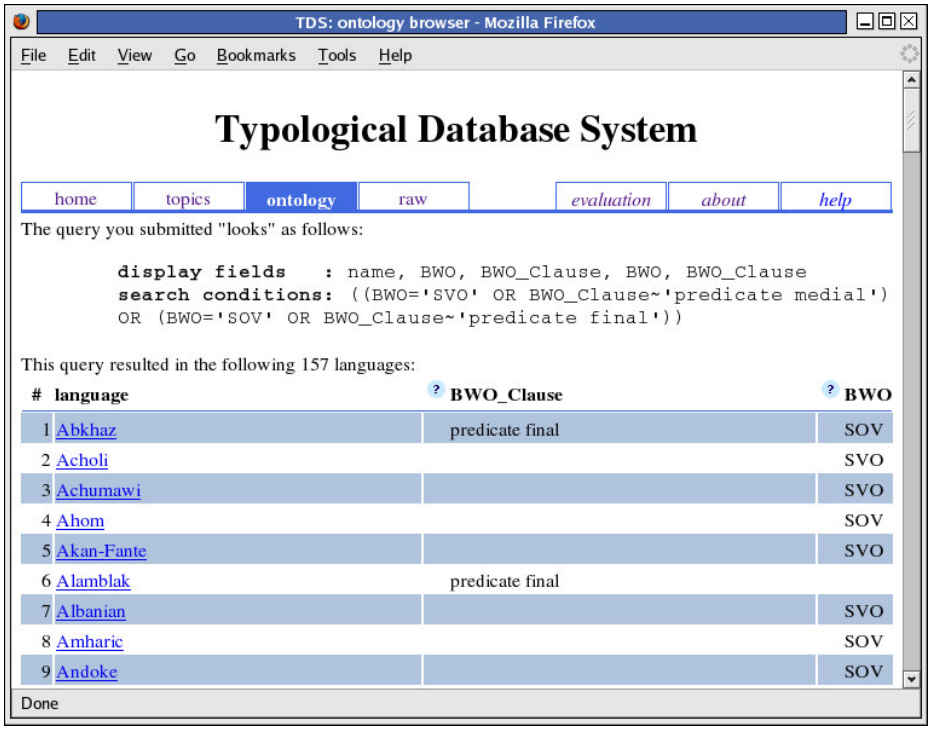

Figure 6. User interface showing the answer on the BWO query of Figure 5

6 Conclusions and future work

In this paper we outlined a distributed approach to the task of semantic integration. On one hand a domain ontology is constructed as a conceptual network, on the other hand low-level annotated schemata are described for component databases. These two semantic descriptions of the global and local modelling approaches are tied together

using a shared vocabulary, *i.e.* notions of the DTL correspond to concepts of the ontology. An advantage of this structure is that it allows crossdatabase searches, while at the same time respects both the global domain overview and local, theory-specific, database notions.

There remain a number of problems still to be tackled. These include the appropriate handling of a sub-class of semantically related values (including unknown, missing or irrelevant) versus NULL (see §4 above). A further issue in research relevant to the linguistic domain is an additional possibility for "fuzzy" queries. A case currently being addressed concerns unified searches of phyla, languages and dialects. For instance, language may have dialects each with typologically interesting features. And languages themselves can be grouped in phyla. We envisage that the ability to search within this hierarchy of generalization and specialization will enable the TDS to show answers in which different descendants are only partially matched by the query, but as a group are completely matched. By implementing solutions for these issues the semantic integration of the databases can be fully exploited.

Acknowledgements

We would like to thank the participating contributors of the component databases.

References

1. Bakker, D. and Siewierska, A., *Person Agreement Database*, University of Amsterdam, Lancaster University, (2004).
2. Date, C. J., *Null Values in Database Management*, in C. J. Date, ed., *Relational Database: Selected Writings*, Addison-Wesley, Readings, Mass., (1986).
3. Electronic Metastructure for Endangered Languages Data, *Terminology Mapping*, (2005).
4. Goedemans, R., van der Hulst, H. and Visch, E., *Stress Patterns of the World*, Holland Academic Graphics, The Hague, (1996).
5. Gómez-Pérez, A., Fernández-López, M. and Corcho, O., *Ontological Engineering*, Springer, London, (2004).
6. Gruber, T. R., *Toward principles for the design of ontologies used for knowledge sharing*, in N. Guarino and R. Poli, eds., *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, Deventer, (1993).
7. Hengeveld, K., *Typological Database Amsterdam*, in TDS project, ed., Amsterdam, (2005).
8. Hengeveld, K., Rijkhoff, J. and Siewierska, A., *Parts-of-speech systems and word order*, Journal of Linguistics, 40 (2004), pp. 527-570.
9. Inmon, W. H., *Building the Data Warehouse*, Wiley, (2002).
10. Jarke, M., Lenzerini, M., Vassiliou, Y. and Vassiliadis, P., *Fundamentals of data warehouses*, Springer, Berlin; New York, (2000).
11. Klamer, M., Musgrave, S. and van Halteren, H., *Spinoza Project Lexicon and Syntax*, Leiden University, (2002).

12. Miller, E. and Hendler, J., *Web Ontology Language (OWL)*, W3C World Wide Web Consortium, (2005).
13. Niles, I. and Pease, A., *Towards a Standard Upper Ontology*, in C. Welty and B. Smith, eds., *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS 2001)*, Ogunquit, Maine, (2001).
14. Pyle, D., *Data Preparation for Data Mining*, Morgan Kaufmann, (1999).
15. Stassen, L., *Intransitive predication*, Oxford University Press, Oxford, (2002).
16. Stassen, L., *Typological Database Nijmegen: Comparison and intransitive predication*, Radboud University Nijmegen, (2004).
17. Stuckenschmidt, H. and van Harmelen, F., *Information Sharing on the Semantic Web*, Springer, Berlin, (2005).