Semantic relations in ontology mediated linguistic data integration[*]

Alexis Dimitriadis[1], Adam Saulwick[2], Menzo Windhouwer[2]

[1] Utrecht Institute of Linguistics, OTS
Trans 10, 3512 JK Utrecht, The Netherlands
alexis.dimitriadis@let.uu.nl

[2] Department of Theoretical Linguistics, University of Amsterdam
Spuistraat 210, NL-1012 VT Amsterdam, The Netherlands
{a.g.saulwick, m.a.windouwer}@uva.nl

Abstract

In developing a system for integrated access to diverse typological databases, we rely on the formulation of general principles by which linguistic concepts and database fields and their relationships can be appropriately modeled. By modeling these entities and relationships, representing them in an ontology and linking them to a merged hierarchical data structure, heterogeneous perspectives on linguistic phenomena can be managed in a unified way without distorting the source data.
In this paper we enumerate these principles and present our approach to data integration with a range of specific examples from component databases, which illustrate the relationship types and integration methods employed.

Keywords: ontology, linguistics, linguistic typology, database integration, semantic integration.

1 Introduction

The Typological Database System (TDS)[1] is an ongoing research project to develop a web-based service for unified querying of multiple independently created typological databases, supported by an integrated ontology. The TDS currently contains information on circa 1,000 languages from five integrated databases. Its component databases contain data on a range of linguistic topics including agreement, parts of speech, word order, stress placement and predication phenomena. Some yet-to-be integrated databases also contain primary linguistic data in the form of lexicons and glossed sentences. The TDS project has developed an ontology of linguistic concepts to facilitate data integration and management. This paper discusses some of the key features of the current TDS ontology (TDSO). These include the purpose of the ontology, the types of semantic concepts encoded in it, its structure, and its content.

The TDSO constitutes an “inclusive” framework of linguistic concepts and terms. By this, we mean that it provides a non-prescriptive frame of reference into which the potentially idiosyncratic perspective of the component databases can be integrated.

Another important feature of our ontology is the differentiation of several types of relations between linguistic concepts. For instance, the relation between the categories ‘case’ and ‘grammatical case’ is not the same the relation between the categories ‘vowel length’ and ‘syllable weight’. The former represents an instance of “specialization” whereas the latter is

[*] The authors would like to thank Rob Goedemans for his contribution to aspects of this paper discussing the StressTyp database.

[1] The TDS project is being carried out by a research group of the Netherlands Graduate School of Linguistics (LOT). In addition to the three authors, the project participants include Rob Goedemans (Leiden University) and Eugenie Stapert (research assistant, UvA). The creators of the component databases contribute data and explanatory information. The TDS project is supported through funding from the Netherlands Organization for Scientific Research (NWO) and participating Universities.

an instance of "determination". We argue that by distinguishing these and other relationships, the ontology can provide a means of reasoning and thus the basis for "smart searching" and display. To this end, we present a range of domain-specific semantic relations that are considered both distinct and basic, and discuss how they are expressed in the language of the ontology. These include subsumption, meronymy, (loose) synonymy, equivalence, determinants, and the form–function relation (see §2.2.2).

The TDS provides access to data from a number of heterogeneous databases, organized in various forms and expressed in ways that reflect different theoretical perspectives. To support a uniform means of access to this information, the ontology is used as a blueprint for bringing the heterogeneous contents of the component databases into a consistent form. Explicit links between the unified data and ontology concepts facilitate searching through the integrated database fields.

Searching is a two-step process. First, the user discovers fields relevant to the topics of interest, by using one of the search or browser options (which use the ontology in various ways) in the TDS interface. These fields are accumulated, forming a pre-query. In the second step, the user refines this pre-query and executes it. Diverse aspects of the system's behavior are controlled by the ontology, directly or indirectly, supporting a degree of "smart searching" (see §4.2).

In the remainder of this section we give a brief overview of the system's goals, fundamental concepts, and architecture.

1.1 Goals of the system

The main purpose of the system is to provide integrated access to the data in a collection of typological databases. The *data,* therefore, is the central resource that the system must manage. The goals of the system are (a) to provide an interface that will help users *find* relevant data, and (b) to allow users to *interpret* the data they are presented with. The linguistic ontology is utilized in both tasks: Searching for data relevant to a topic is mediated by links between database fields and values and concepts in the ontology. Interpretation of the data is supported by the concept documentation, which may be presented alongside a database field's own documentation. As data may be presented out of their original context, in all cases the interface must provide the provenance of the data along with database-specific description; this allows users to properly evaluate the information presented.

1.2 Nomenclature

In order to make explicit the principles underlying our use of the ontology, it is useful to define some basic terms. The information in a database is stored in *records,* which contain the *values* for a number of *fields.* Since the databases may (and frequently do) assume different versions of the linguistic concepts they are concerned with, we utilize a "hybrid", or two-level, model of the semantic domain (Stuckenschmidt & van Harmelen, 2005). The TDS defines a *global ontology* of linguistic concepts, whose contents are meant to be as far as possible compatible with diverse conceptualizations of the domain (more about this in §2). The meanings applicable to individual component databases form a set of *local ontologies,* whose categories correspond in a stricter or looser way to those in the global ontology. For convenience, we refer to categories in the global ontology as *Concepts,* and to categories in the local ontologies as *Notions*. In our system, Notions are not just the local equivalent of Concepts in the global ontology: they also serve other functions, such as containing data and documentation imported from the databases, or grouping other Notions. Notions corresponding to a data field are called *field Notions.*

1.3 Overview of the system architecture

An overview of the TDS architecture is shown in Figure 1. The system operates by periodically importing the contents and metadata of the component databases. These are restructured, merged and transformed into a single hierarchical data structure, *i.e.,* a tree. The user interface interacts with this tree, and with its associated specification schemas, extracting information through queries. To create the global tree, each database is first addressed on its own, *i.e.,* a collection of trees is built with each containing the data of one component database. The schema for each of these trees is described using a special-purpose language we have developed, called the *Data Transformation Language* (DTL). A DTL specification describes Notions and their relationships, and the nodes in the tree are thus instantiations of these Notions.

The internal organization of the DTL specification is designed to parallel, as far as possible, relevant parts of the (global) linguistic ontology. Notions can be shared by the various DTL specifications, and their instantiations, *i.e.,* the nodes (where needed identified by a key2), can be shared. This allows the collection of trees to be merged into a single one. This tree may also be enriched with *derived Notions,* which are field Notions computed on data from one or more component databases. The result of this series of steps is a single hierarchy containing all the data from the integrated databases, shown in Figure 1 as the *TDS data*.

The query system interacts with both the TDSO and the DTL specifications, allowing a user to discover database fields of interest and thus to formulate and execute a query. Selected fields are collected into a pre-query that can be further refined and carried out. The query process, and the role of the ontology in it, are described in §4.

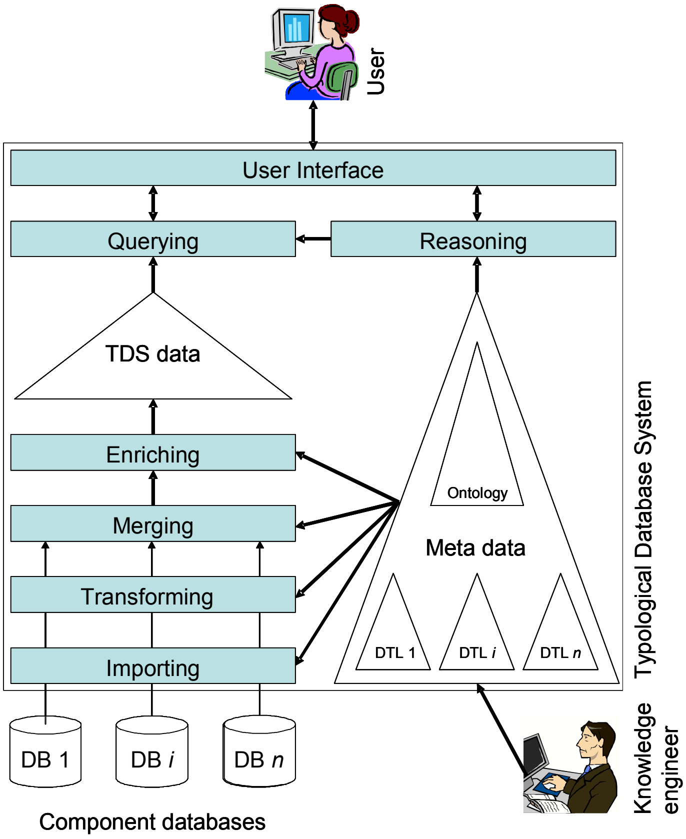

Figure 1 TDS architecture

2 Design of the linguistic ontology

The task of the ontology is to provide a basis for the integration of diverse databases containing linguistic information. In order to do this, the linguistic ontology must specify domain vocabulary supported with descriptions, and information about the logical structure of complex Concepts. To date the thematic range of the ontology covers information on agreement, parts of speech, word order, stress placement and predication phenomena. The

[2] For example, *language* nodes are identified by their Ethnologue code where possible (Gordon, 2005).

ontology organizes the relevant linguistic Concepts into a coherent network. Where information in more than one database relates to the same topic, it is the task of the ontology to establish a valid conceptual structure that will enable the integration of diverse conceptualizations.

The TDS project takes a neutral standpoint towards the incoming (meta-)data as regards its validity with respect to the phenomena described. That is, we do not consider it the job of the TDS project to make value judgments on the quality of the analyses expressed in the contributing databases, but rather to provide access to information through cross-database querying. As long as the provenance of information is made explicit, end users will be able to make their own judgments as to the usefulness and/or reliability of query results. It should be added that contributing developers have spent considerable time and resources encoding information in their databases; hence, in database terms, the databases represent very high-value information, created through considerable human effort and utilizing extensive domain expertise. In short, each contributing database represents an extremely valuable resource.

We now discuss some of the principles concerning the integration of concepts in the ontology.

2.1 Conceptual principles underlying ontology development

The TDS project is built using one of the industry-standard languages, Web Ontology Language (OWL). The choice of OWL is motivated by (among other things) the requirement for extensibility, ease of integration in our XML-based system, web-based user interface querying and the availability of development tools. As discussed in Saulwick et al. (2005), our methodology follows current recommendations for ontology building (Gruber, 1993; Gómez-Pérez, Fernández-López, & Corcho, 2004), namely: *clarity*, *coherence*, *extendibility* [sic], *minimal encoding bias*, *minimal ontological commitment*, *representation of disjoint and exhaustive knowledge*, *minimization of syntactic difference in encoding* and *standardization in naming conventions*. Important features of ontology-driven integration are the use of shared vocabulary in a coherent and consistent manner (Gruber & Olsen, 1994) and where possible the standardization of naming conventions. In the following paragraphs we will discuss the conceptual principles guiding TDS ontology development.

A bottom-up approach. A fundamental design principle of the TDSO concerns the basis for the postulation and establishment of Concepts (*i.e.,* OWL classes, properties or individuals). It is a design and methodological principle of the TDS that ontology Concepts are only established on the basis of information *existing* in component databases, thereby constraining the ontology to a range of relevant concepts. This is motivated by the desire to ground the ontology in empirical data-based theory, and thus acts as a limiting device on otherwise unconstrained ontology growth. However, a concept *may* be established for which there is no database mapping if the concept is syntagmatically or paradigmatically relevant. For instance, the TDSO Concept `Transitive Object` (the second argument of a transitive verb) is not linked to any currently integrated database fields, but it exists in the ontology based on the occurrence of the Concepts `Intransitive Argument` (the sole argument of an intransitive verb) and `Monotransitive Argument` (the first argument of a transitive verb) which do figure in links. In other words, a Concept may be established if it fills a paradigmatic or syntagmatic gap in the network thematic domain.

Prototypes. As is well known in prototype theory in linguistics (Rosch & Lloyd, 1978; Taylor, 1989; Varela, Thompson, & Rosch, 1991), the set of entities subsumed by a category may have greater or fewer features/attributes associated with the category than a single exponent of the class, depending on whether they represent more or less prototypical exponents of the class. A classic example is the class *chair*. We know that ideas vary concerning the requirement of specific features such as number of legs, occurrence of a back rest, and so on. The TDSO adopts a prototype approach to the classification of linguistic categories. For instance, the class `free pronoun` subsumes the classes `Cardinal Pronoun`, `Demonstrative Pronoun`, `Emphatic Pronoun`, `Personal Pronoun`, `Possessive Pronoun`, `Reflexive Pronoun`, and `Weak Form Of Person Marker`. Subsumption represents the standardly used 'is-a (kind of)' relation, where the subordinate entities represent specializations of the category. It is clear that the entities *cardinal*, *demonstrative*, *emphatic*, *personal* and *possessive pronouns* are each a special type of the superordinate class *free pronoun*. That is, each of these classes has at least one additional feature that is the basis for its specialization. We could label each of these features, respectively, as +value cardinal, +value demonstrative, +value emphatic and so on. Now, in terms of classification one could argue that the class `weak form of person marker` is an invalid specialization of the class `free pronoun` because it is not necessarily *free* or unbound. That is, exponents may be free, cliticized or bound depending on the language. Thus in the strictest sense the class `weak form of person marker` is not a specialization of the `free pronoun`. However, adopting a prototype analysis allows for a subordinate class (in this case `weak form of person marker`) to have features in apparent conflict with the superordinate class if certain core features of the specialized class are consonant with the superordinate category. In this case we could describe some of these as: ‘deictic marker referencing person referents’.[3] By permitting the kind of prototype classification presented here, a richer and thus more fine-grained network of associations between categories is provided. Arguably, this results in the possibility of more extensive cross-data mappings and thus facilitates more effective resource discovery.

Theory-neutral perspective. Each of the component databases reflects the theoretical stance of its creator, both in the way linguistic phenomena are conceptualized and in the terminology used to describe them; when diverse databases provide information about the same topic or use the same term, there is the potential for mismatch. As stated in the Introduction, the TDSO is an *inclusive* ontology of linguistic concepts: it provides a common vocabulary that serves as a non-prescriptive basis for the integration of database-specific categories. The TDSO is by design maximally compatible with different conceptualizations of linguistic phenomena. It represents crucial concepts but attempts to refrain from incorporating details peculiar to a particular theoretical orientation. This does not mean that the ontology itself consists of Concepts that are “a-theoretic”. Indeed we hold that such a pursuit is unattainable for the simple reason that all terms bear the hall-marks of their particular theoretical orientation. Rather the ontology is “inclusive” in the sense that it can accommodate the variety and richness of individual theoretical orientations with all their idiosyncrasies. Where appropriate, variant and potentially conflicting orientations/conceptualizations are included in the global ontology and are unified under broader categories.

The decision whether to include a Concept in the ontology is dependent on how commonly accepted a linguistic category is, within or across (conflicting) linguistic theories. A linguistic category that is not included in the global ontology is treated as a concept in a local ontology: it is represented as a DTL Notion (a *Concept Notion)*, as discussed in Saulwick et al. (2005) and in §3.2 below.

In this way, the ontology strives to achieve a variation on the principle of Gruber’s (1993) minimal ontological commitment, namely *minimal orientation commitment*. This is the inclusion of diverse theoretical orientations without fear or favour, *i.e.,* without ascribing to any one of them a favoured status. Where a single database adopts a certain theoretical orientation, this may temporarily skew the relevant section of the ontology toward that particular orientation. Where a subdomain is covered by multiple databases, which at least partially overlap, this will require the establishment of a richer vocabulary. This principle should not be understood as tolerance for the use of vague or inadequately defined terminology.

Having described the theory-neutral perspective of the TDSO, we now briefly exemplify it. Saulwick et al. (2005) discussed the case of the variant Concepts `Basic Word Order` and `Predicate-Based Word Order`. In the TDSO these are unified under the supercategory `Core Constituent Word Order`. We call this semantic unification; not an ironing out or watering down of theoretical orientation, but the introduction of an inclusive superconcept for the purposes of information integration. A query over any one of these Concepts allows the end-user access to the others. In adherence to the principle of

[3] The degree to which a weak form is able to encode referential specificity is not at issue here, see Siewierska (2004, 9, 124ff).

clarity (Gómez-Pérez et al., 2004), the TDS will ensure that the intention behind each database contributor's use of terminology is faithfully represented in the ontology (supplemented as necessary by database-specific documentation of Notions in the DTL).

Now that some of the important principles constraining the TDSO have been discussed, we can present some design and implementation details of the linguistic ontology.

2.2 Linguistic Concepts

The linguistic ontology describes a number of linguistic objects, relationships and other linguistics-related ideas, collectively referred to as (*linguistic*) *Concepts.* Concepts are labeled and described with a short explanation, and possibly references. In this way the intended extension of the concept can be better understood by users, and the linguistic ontology can also serve as a guide to the terminology.

2.2.1 Types of Linguistic Concepts

We distinguish between the following major types of linguistic Concepts:

Linguistic objects can be thought of as existing in themselves. They include Concepts such as `Sentence`, `Morpheme` and `Phonological Segment`, as well as classes representing `Language` and various groups of languages.4

Linguistic properties are (linguistically salient) properties predicated of a linguistic object. For example, `Basic Word Order` is a property of `Languages`, while `Referential` is a property of certain words or syntactic constituents. Properties, in this terminology, do not relate one linguistic object to another but can be thought of as one-place predicates. Properties are generally associated with a set of possible values; for some the values are True/False or Present/Absent, while for others it may be one of several possibilities with linguistic meaning such as a paradigm, as with the property `Case` which can have the values `Nominative/Accusative/Dative`, etc.

Linguistic relations model a phenomenon involving two or more linguistic objects or properties. For example, following Corbett (1998:191), `Agreement` is modelled as a relationship involving a `controller` ('the element which determines agreement'), a `target` ('[t]he element whose form is determined by agreement'), a `domain` ('[t]he syntactic environment in which agreement occurs'), and `agreement features` ('in what respect there is agreement'). The participants in a relation play distinguished `roles`, whose names may be particular to each relation: for `Agreement`, the roles are `controller`, `target`, etc. Complex phenomena that are not explicitly relational are also treated in terms of roles: for example, `Stress Assignment` can be described as involving a `Method` (algorithm) that makes reference to types of feet, edge-sensitivity, extrametrical material, etc.

We use OWL Classes to model all three types of linguistic Concepts. For linguistic objects, which are not inherently relational, this is an obvious choice. Linguistic properties are modeled using Classes, not OWL Properties, because within linguistic theory they are concepts in their own right, and have class-like properties that should be encoded in the ontology.5 This modeling choice allows the characteristics of properties and relations to be modeled more accurately and results in a more uniform ontology schema.

Linguistic relations are also modeled as OWL Classes. In this case there is no alternative, since an OWL Property can relate at most two classes,6 while linguistic relations can have higher arity. We follow a customary method for modeling higher-arity relations (see the W3C recommendation for the application of this method to relationships between individuals (Noy & Rector, 2004)). Roles are modelled as OWL Properties, with names that start with the

[4] A linguistic object corresponds to the GOLD class *Linguistic Unit;* however, the two ontologies (presently) have different subclass hierarchies (EMELD, 2003).

[5] In particular, linguistic properties participate in relationships with each other: for example, the *Determinant* relation (see §2.2.2 below) is a relationship between linguistic properties, one of which is (partially) determined by the value of the other.

[6] Object Properties in OWL relate two classes. Data Properties associate a single class with a value.

special prefix hasRole. For example, we define an OWL Property hasRoleTarget that associates the linguistic Agreement relation with its target.

The ontology implementation also includes auxiliary concepts and properties which we use to model the linguistic concepts. The hasRoleTarget OWL Property is an example of such an auxiliary property. This is not considered a linguistic relation: it is simply part of the implementation.

2.2.2 Relationships between Linguistic Concepts

Entities in the ontology are organized according to the following major relationship types:

Subsumption (super- and subordinate Concepts). Some linguistic concepts are specializations of others. For instance, 'grammatical case' is subsumed by the more general Concept of 'case'. A subsumption relationship between two Concepts is expressed by putting the relevant OWL Classes in a class/subclass relationship.

Loose synonymy (variant linguistic terminology used to refer to the same phenomenon). When two phrases denote the same conceptualization of a phenomenon, it is useful to link them in order to provide a means of searching using different vocabulary than used for naming the ontology Concepts. (Loose) synonymy between two phrases is currently implemented as an *annotation* on the class (essentially, a data property that gets a value for the entire class). The Concept Agreement Marker is for example annotated with the alias Person Inflection.

Related phenomena (variant linguistic terminology used to refer to the similar or related phenomena). Basic Word Order has this relationship to Predicate-Based Word Order. Although the two phrases denote slightly different conceptualizations of phenomena, it is useful to link them in order to provide a means of unified searching across both component databases in which the terms occur. As neither of the standard annotations OWL:sameAs and OWL:equivalentClass (Bechhofer et al., 2004) captures our required semantic correspondence, we use our own TDS:equatesWith annotation to equate two related concepts.

Meronymy (part/whole relations).7 Some linguistic concepts are modelled in a strict hierarchical structure. Certain linguistic hierarchies are organized so that units of one type are a *direct part* of the next higher unit, e.g., in the 'prosodic hierarchy' (Nespor & Vogel, 1986), which is a hierarchy of utterance constituents from a prosodic perspective in which "[e]very prosodic category in the hierarchy has as its head an element of the next-lower level category". (Kager, 1999:146). The direct-part-of relation is a specialization of the general part–whole relation.8 We encode part-whole relationships via a meronymic predicate, isDirectPartOf (and its transitive closure, isPartOf). This relation is asserted between pairs of classes, and serves to organize them into meronymic hierarchies. For example, *mora* > *syllable* > *foot* > ... form a meronymic hierarchy.

Determination is the name we use when a linguistic property is defined in terms of one or more other linguistic properties. For example, if a heavy syllable is defined as a syllable with a long vowel or a coda, then both of these are determinants of the property Syllable Weight (even though the presence of only one is enough to make a syllable heavy).[9] The determinant relation is a relationship (supporting, not linguistic) between linguistic properties

[7] A variety of meronymic relationships may be required. For instance, Story (1993) based on Landis et al. (1987); Winston et al. (1987) and Chaffin et al. (1988) lists seven types of meronymic relations: component–object, member–collection, portion–mass, stuff–object, phase–activity, place–area and feature–event.

[8] Our implementation follows the current W3C recommendation, which calls for expressing meronymic relationships in terms of a direct part relation when 'what is needed is not a list of all parts but rather a list of the next level breakdown of parts, the "direct parts" of a given entity' (Rector & Welty, 2005).

[9] Determination only holds when the definition of a concept involves aspects of another. It should not be confused with implicational relationships which are empirically based. In the latter case we have two concepts which are independently defined, and the implicational relationship is an empirical fact rather than part of their meaning.

or relations. The names of determinant OWL properties begin with the prefix `isDeterminant` or `hasDeterminant`, which is recognized and treated specially by the system.

Form-function relationship. Here "function" is used in its linguistic sense: It refers to the linguistic function served by some linguistic entity. This relationship associates entities of type linguistic object with linguistic properties expressing their possible function. For example, `Agreement Marker` is a possible function of `Affix.` A form-function relationship is a linguistic relation, and it is implemented accordingly (*i.e.*, as an OWL Class with roles expressed as OWL Properties).

3 Ontology-mediated data integration

3.1 The heterogeneity challenge

In order to integrate the data from various component databases, the system must overcome the heterogeneity of the imported data, which can have several sources:

Different types of content. So-called "analytical" typological databases consist of logical variables describing each language as a whole. Other databases contain example sentences with detailed annotations ("sentence databases"). One of the goals of the project is to integrate different types of content so that, for example, a single query could search both examples and logical variables for relevant information.

Different theoretical commitments. The information in the various databases reflects the analytical and theoretical commitments of its creators. The TDS places a high priority on preserving and presenting to the user the framework of database-specific assumptions required to properly interpret the data extracted from a component database. Such information will allow a knowledgeable user to recognize the descriptive content of a statement about language, and to gain useful information from it even if it does not exactly match one's own theoretical orientation. (It should be clear that conversions between theories cannot be automated with any reliability). Therefore it can be useful for users to view information that is not expressed in terms of their theoretical framework. For example, information about the properties of "subjects" can be useful even to linguists who do not believe that this is a well-founded notion.

Constructed for different purposes. The focus and detail of coverage varies depending on the creators' own research interests, even where there are no theoretical disagreements. Such variability is not theoretically contentful, but it leads to differences in the structure and content of conceptually similar data.

Different notational conventions. In many cases the different databases use equivalent, or near-equivalent, ways of describing data. An obvious example of this is the use of different gloss labels for broadly accepted linguistic categories, such as "*p*" or "*pl*" for *Plural.* It is generally easy to reconcile purely notational differences, but the databases can also differ in the details of how such concepts are applied. Such cases are modeled as database-specific Notions, which constitute variations of the Concept in the global ontology.

Different design choices. There are multiple ways to organize a body of information into question/answer pairs, and the databases therefore differ in their structure. This is a prototypical use of ontologies: to guide the mapping between different structures with equivalent semantics. As this source of variation is compensated for, it becomes easier to organize semantically related, but non-identical, sources of information.

Different software. The reliance on different software platforms (applications, operating systems, file formats, etc.) introduces an additional layer of incompatibility, which is straightforwardly addressed without the mediation of ontologies.10

As already described in section §2, the TDSO is involved in addressing several of these forms of heterogeneity. Multiple theoretical commitments can be integrated into the ontology

[10] Our system utilizes a plug-in model to import databases created on different software platforms. Usually they are sufficient to import data into the system. Occasional small glitches will require manual intervention in the form of ad hoc fine-tuning scripts, but these are quickly resolved and only need to be addressed once per database.

thanks to its inclusive design. However, to prevent proliferation in the ontology of too many theory-specific Concepts, the knowledge base designer can decide to capture part of the theory-specific linguistic categories as DTL Notions. By providing links from these Notions to the related Concepts, the theory-specific data can be accessed. As will be discussed in the following section, DTL Notions are used to address integration problems originating from various of the other heterogeneity sources.

3.2 Data Transformation Language

If we want to apply the global ontology to the local, theory-biased data, the semantic gap between the two models has to be bridged. This is especially needed when we want to do cross-database searches. As stated in §1.2 the TDS takes a hybrid approach (Stuckenschmidt & van Harmelen, 2005), which means that a local ontology specifies how each database schema relates to the global ontology. The local ontology is in this case not a full-fledged ontology but the description of a hierarchical overlay on the database schema. This hierarchy is defined in a DTL specification.

As discussed in section §1.3, the nodes of the DTL play a variety of roles, with associated semantic interpretations. At the lowest level of the hierarchical overlay, we find Notions associated with database fields and/or values. These can be viewed as having content specific to the database of origin. It should be noted that such *field Notions* or *value Notions*[11] are not necessarily identical to the original fields and values; they can be transformed in the DTL specification in order to overcome design differences between the component databases. For example, a database field is sometimes split into two or more DTL Notions, or several binary-valued database fields may be combined into a single multi-valued DTL Notion (but only when this can be done with no loss of information). The strings used to encode field values may also be replaced by codes with a standardized use throughout the TDS. For example, the DTL fragment in Figure 2 replaces the database specific notation for Boolean values with the TDS standard, `True/False`:

```
NOTION StressRhythmDiffer IS FIELD Rhythm {
  True  FOR "Y" OR "y";
  False FOR "N" OR "n";
}
```

Figure 2 DTL fragment showing value transformation

While data is restructured in these ways, the contentful meaning of the database fields, *i.e.*, the theoretical orientation and concrete methodological choices behind the collected data, should always remain intact. A contributor's decision to assign a particular Basic Word Order to some language, for example, is informed by a specific perspective on this concept as well as by particular methodological choices, often unstated, that were followed during data collection. These are factors that the TDS neither could nor should try to modify.

On the higher levels in the hierarchy we find Notions that group other Notions. These Notions may be linked to ontology Concepts, in which case they are called *concept Notions.* The hierarchical structure formed by these Notions can be patterned after the organization of related Concepts in the global ontology. For example, the ontology provides a Concept representing the property `Extrametricality`, and defines roles for the participants `Alignment` and `Size.` Accordingly, the DTL specification groups relevant field and value Notions under one Notion, corresponding to the Concept `Extrametricality`*,* and further links them to particular Concepts playing a role in this linguistic relation when appropriate. This hierarchy can be expressed in the DTL by building a nested structure of Notions, as in Figure 3.

[11] In Saulwick et al. (2005) Notions were grouped into the classes derived and underived. This grouping focused on the DTL internal role of a Notion, *i.e.,* would it transform the incoming data (and thus add newly derived semantics) or not. The roles of Notions described in this paper is orthogonal to this grouping, and focuses on the (external) relationships of the Notions. While a Notion is either derived or underived it can play different roles, *i.e.,* have relationships of different types.

```
NOTION EM
  LABEL "Extrametricality"
  LINK AS CONCEPT Extrametricality
  GROUPS {
    NOTION Alignment
      LINK TO CONCEPT Extrametricality.hasRoleAlignment
      MEANS "The word edge at which extrametrical material occurs"
      IS FIELD Stress_EM {
        "Left" FOR "L";
        "Right" FOR "R";
        IGNORE OTHERWISE;
      }
    NOTION Unit
      LINK TO CONCEPT Extrametricality.hasRoleSize
      IS {
        IGNORE FOR FIELD Stress_EM="no";
        COPY FIELD Stress_EM_Unit OTHERWISE;
      }
  }
```

Figure 3 DTL fragment showing Notion grouping and links to Concepts

Because grouping Notions may dominate field and/or value Notions derived from several databases, they cannot be associated with the particular semantics of any one database. They can be viewed as reflecting the "inclusive" semantics of the TDSO Concept to which they correspond. (Of course, the Notions they dominate still carry the original meaning of the component databases).

The relationship between a local ontology concept and its global counterpart may be exact or approximate. The links between Notions and TDSO Concepts indicate the type of relationship and the degree of resemblance between the Concept's and the Notion's particular semantics. A Notion may be more specific in meaning than a related Concept in the global ontology, or it may be strictly more general, or the two may substantially overlap without either including the other. For example, the Notion `BWO Clause` has an overlapping relation with the Concept `Basic Word Order`. A Notion could also be essentially equivalent to the related Concept, in which case we encode it as an "as" relationship; otherwise we encode it as a relation "to" a Concept. The Notion `BWO Clause` is also related to the Concept `predicate-based word order`, and in this case the relationship is of the type "as". The DTL excerpt in Figure 4 uses both types of Concept links:

```
NOTION BWO_Clause
  LABEL "BWO Clause"
  MEANS "…"
  LINK givesValues AS CONCEPT predicateBasedWordOrder
  LINK TO overlapping CONCEPT BasicWordOrder;
```

Figure 4 DTL fragment showing two linkage types

The semantic gap, which prevents the ontology from functioning directly as a schema for the data, is thus bridged by the DTL specification, allowing the general Concepts to be linked to the source data via the DTL Notions. A user searching through the global concept `Basic Word Order` will find the database-specific Notion for `BWO Clause` and will have access to both the documentation of the global concept and the database specific documentation, which would allow him/her to assess the semantics of both and thus bridge the semantic gap.

Thus, in contrast to the global ontology, the local schemata of the component databases are partially theory-specific. The semantics of Notions are informally specified, through free text imported from the component database documentation or written by the TDS domain expert in close consultation with the database creator. Note that there is only one global ontology, while there are multiple local databases and DTL import schemata, although the latter can partially overlap. Integration of the database schemata is thus facilitated by shared Notions and/or shared Concepts.

The DTL specification language is similar in its purpose to the Semantic Interpretation Language (SIL) (Simons et al., 2004); both are structure specifications that induce a data transformation. In the case of the DTL, the data is transformed into a tree which adheres to the schema specified in the DTL specification. Notions in this schema may relate to Concepts in the ontology in more or less precise ways. An SIL specification also defines a transform, but the result of it is a tree which fully adheres to the ontology. Thus in the latter case the resulting document is tightly coupled with the ontology, and the method is only applicable to data that can be fully mapped to the exact ontology semantics. In our task domain, this will only be true for a subset of the data in the source databases; the DTL will import the remainder as source-specific records, whose semantics do not fully match those of the ontology.

4 Ontology-based querying

Query construction in the TDS system is a two-step process. In the first stage, the user searches for database fields relevant to a topic, and aggregates them into a "query basket" (similar to the "shopping cart" of online sales websites). In the second stage, the selection and projection criteria on the collected fields are fine-tuned to arrive at the final query, which is then executed. Here we are mainly concerned with the first stage, which requires the user to select from among a very large number of database fields. The ontology is used to assist in this process.

4.1 The query process

The TDS presents the user with two models: the global ontology and the DTL model. The ontology can be used to provide users with an interface directly based on the relationships within the ontology, or it can be hidden from the users' view and used indirectly to aid the discovery process. Therefore, the user interface may take the following forms:

- A view of the Concepts, with links to related Notions at appropriate places. Concepts may be accessed through an explorer-like tree view of the relationships between the Concepts, via text search, or as an alphabetical index of Concepts.
- A Google-like search for Notions and groups of Notions, which searches links to ontology Concepts as well as the Notion's own documentation and contents.

The TDS currently supports (experimental) versions of all of these forms. In all cases users can get access to a detailed view of a selected Concept, which includes the list of Notions linked to it. This allows the users to see how the source-specific Notions (which are usually at the lowest level of grouping) are related to the inclusive Concept. Based on this detailed knowledge, users can select field and value Notions they find relevant and accumulate them in a "pre-query basket", similar to the "shopping cart" of many e-commerce websites. This "pre-query" can be incrementally built, by adding Notions discovered through any one of the several navigation interfaces. When sufficient desired Notions have been discovered, the user examines the contents of the pre-query basket, choosing values to search for and adjusting search and display options. For example, the method of combining query terms can be adjusted (AND or OR), the display of terms used as selection criteria can be suppressed, or alternative display formats can be requested. Our system can display tables in HTML format, through a Java applet that allows manipulation and reordering of table contents, or in a raw format intended for export.

4.2 "Smart searching"

At the simplest level, ontology Concepts linked to DTL Notions serve as keywords that a user can search for, to discover the linked database fields. But since the linked-to Concept is not an isolated keyword but a node in a structured ontology, relations between Concepts can be exploited to achieve a degree of "smart searching". For example, Figure 5 shows a part of the ontology that is concerned with word order phenomena.

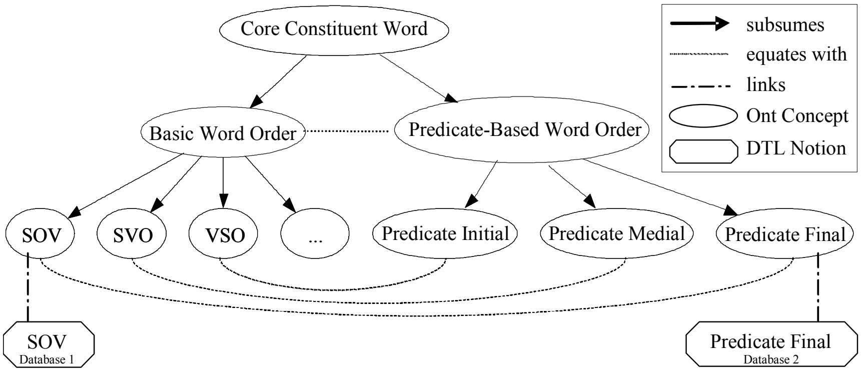

Figure 5 The ontology context of the Concept Core Constituent Word Order

The ontological relationships shown in this figure form the basis for a more advanced form of "smart search". Assume the user has selected the "SOV" value Notion which is linked to the TDSO Concept SOV. The TDS now searches for the existence of special relations between this Concept an another in the ontology. The system will find that the Concept SOV is equated with the Concept Predicate Final. As shown in Figure 5 these Concept are in an "equates with" relationship (a *non-exact* association between related phenomena) with each other. This Concept is linked to the "Predicate Final" value Notion in the DTL. By extending the query with this new Notion the user gets access to related information, even though the database Notions "SOV" and "Predicate Final" are related to different source databases and each reflects theory-specific perspectives on word order.

Other extensions of the query, with semantically related search criteria based on the "equates with" relationship, can be executed by following the subsumption and meronomy relations. These examples of "smart search" facilities show that it is advantageous to exploit the relationships encoded in the global ontology to assist a user in creating a crossdatabase and, possibly, a cross-theory query.

5 Summary

We have presented a modular data integration system which presents the end user with a single interface to diverse linguistic data from multiple component databases, organized in a unified hierarchical structure and interlinked with an integrated global ontology of linguistic concepts. By modeling the ideas found in component databases as a network of Concepts in the global ontology, and using these as a guide to structuring the imported data on the basis of a constrained set of relationship types, we enable the system to perform a variety of automated data-grouping and "smart-search" queries.

As we have seen, the variety of areas in which component databases are heterogeneous is not trivial and requires careful attention to facilitate integration. At the conceptual level, we have shown that it is therefore important to differentiate between the complex, often theory-specific ideas represented in component databases, and the general linguistic Concepts represented in the global ontology.

We have highlighted the need to represent and document as faithfully as possible the theoretical perspectives of database creators within the overall system. A function of the TDS, then, is to provide the general semantic context of database-specific terminology. We have shown that in some instances it is possible and, for the purposes of cross-database searching, beneficial to unify different theoretical orientations on particular linguistic phenomena, by establishing superordinate concepts and/or indicating their non-exact equivalence within the ontology.

References

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., et al. (2004). OWL Web Ontology Language Reference. Retrieved 31/05/2005, 2005, from http://www.w3.org/TR/owl-ref/

Chaffin, R., Herrmann, D. J., & Winston, M. (1988). An empirical taxonomy of part-whole relations: Effects of part-whole type on relation identification. *Language and Cognitive Processes, 3*(1), 17-48.

Corbett, G. G. (1998). Morphology and Agreement. In A. Spencer & A. M. Zwicky (Eds.), *The Handbook of Morphology* (pp. 191-205). Oxford, UK; Malden, Mass.: Blackwell.

EMELD. (2003). General Ontology for Linguistic Description (GOLD). 1.2.2. Retrieved 31/05/2005, 2005, from http://emeld.org/gold-ns/index.cfm

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological Engineering* London: Springer.

Gordon, R. G., Jr. (2005). Ethnologue: Languages of the World. 15. Retrieved 31/05/2005, 2005, from http://www.ethnologue.com/web.asp

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In N. Guarino & R. Poli (Eds.), *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation.*Deventer: Kluwer Academic Publishers.

Gruber, T. R., & Olsen, G. (1994). *An ontology for Engineering Mathematics.* Paper presented at the Fourth International Conference on Principles of Knowledge Representation and Reasoning, Bonn, Germany.

Kager, R. (1999). *Optimality theory* Cambridge, U.K.; New York: Cambridge University Press.

Landis, T. Y., Herrmann, D. J., & Chaffin, r. (1987). Development differences in the comprehension of semantic relations. *Zeitschrift für Psychologie, 195*(2), 129-139.

Nespor, M., & Vogel, I. (1986). *Prosodic Phonology* Dordrecht, Holland/Riverton, USA: Foris.

Noy, N., & Rector, A. (2004). Defining N-ary Relations on the Semantic Web: Use With Individuals. Retrieved 31/05/2005, 2005, from http://www.w3.org/TR/swbp-n-aryRelations/

Rector, A., & Welty, C. (2005). Simple part-whole relations in OWL Ontologies. W3C Editor's Draft 24 Mar 2005. Retrieved 17-05-2005, 2005, from http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/index.html

Rosch, E., & Lloyd, B. B. (1978). *Cognition and categorization* Hillsdale, N.J. New York: L. Erlbaum Associates distributed by Halsted Press.

Saulwick, A., Windhouwer, M., Dimitriadis, A., & Goedemans, R. (2005). *Distributed tasking in ontology mediated integration of typological databases for linguistic research.* Paper presented at the 17th Conference on Advanced Information Systems Engineering (CAiSE'05), Porto.

Siewierska, A. (2004). *Person* New York: Cambridge University Press.

Simons, G. F., Lewis, W. D., Farrar, S. O., Langendoen, D. T., Fitzsimons, B., & Gonzalez, H. (2004). *The Semantics of Markup: Mapping Legacy Markup Schemas to a Common Semantics.* Paper presented at the XMLNLP Workshop, Barcelona, Spain.

Storey, V. C. (1993). Understanding Semantic Relationships. *VLDB Journal, 2*, 455-488.

Stuckenschmidt, H., & van Harmelen, F. (2005). *Information Sharing on the Semantic Web* Berlin: Springer.

Taylor, J. R. (1989). *Linguistic categorization: prototypes in linguistic theory* Oxford [England], New York: Clarendon Press; Oxford University Press.

Varela, F. J., Thompson, E., & Rosch, E. (1991). *The embodied mind: cognitive science and human experience* Cambridge, Mass.: MIT Press.

Winston, M. E., Chaffin, R., & Herrmann, D. J. (1987). A taxonomy of part-whole relations. *Cognitive Science, 11*(417-444).