The Design of the Cornetto Database

Version: 3

Date: 19 August 2008

Project: STE05039

http://www.let.vu.nl/onderzoek/projectsites/cornetto/start.htm

Funded by the Stevin framework (www.taalunieversum.org/stevin)

Authors:

- Katja Hofman, ISLA, University of Amsterdam
- Isa Maks, Faculteit der Letteren, Vrije Universiteit Amsterdam
- Piek Vossen, Irion Technologies/ Faculteit der Letteren, Vrije Universiteit Amsterdam

Table of Contents

1 Introduction

The Cornetto project (STE05039) is funded by the Nederlandse Taalunie in the STEVIN framework (www.taalunieversum.org/stevin). The goal is to build a lexical semantic database for Dutch, covering 40K entries, including the most generic and central part of the language. Cornetto will combine the structures of both the Princeton Wordnet and FrameNet for English, by combining and aligning two existing semantic resources for Dutch: the Dutch wordnet (Vossen 1998) and the Referentie Bestand Nederlands (Martin et al 1999). The Dutch wordnet (DWN) is originally based on the conceptual structure of the Van Dale VLIS database, and has been revised and refined in the EuroWordNet project (1996-1999, www.hum.uva.nl/~ewn). DWN has a similar structure as the English Wordnet although the top-level hierarchy was developed from an ontological framework and more horizontal relations are defined, such as roles and subevents. It has 55K entries, 70K word meanings and 110K semantic relations. The *Referentie Bestand Nederlands* (RBN) was developed with financial support of the governments of The Netherlands and Flanders (1996-1999, revised in 2004). It contains 45K entries with corpus-based descriptions, covering morphology, syntax, combinatorics, semantics and pragmatics. RBN is concept-based but also encodes the common word combinations that go with concepts, ranging from free combinations, to collocations, idioms, pragmatic formulae, proverbs and all sorts of typical and common usage and clichés, lexical functions and compositional and selectional preferences. Combinations can be encoded as lexical functions a la Me'lcuk or FrameNet-like structures.

The data for Cornetto will primarily be derived from conceptually aligning the two Dutch databases. Additionally, the database will be aligned to the Princeton Wordnet for English, from which further semantic resources can be accessed, such as ontologies and Wordnet Domains (Magnini et al. 2002). The alignment will be done automatically but will be refined in a manual editing process.

This document describes the design of the data structure and the content of the Cornetto database (CDB). In the next section, we will first globally describe the modules and data elements from a linguistic point of view. Section 3 will then give an overview of the structure from a database point of view. This specification also defines the databases and the data for the client programs to edit and view the data. The structure is formally defined in an XML

schema and all data are also available outside the database as XML files. The functionality of the client programs is defined in section 4. It explains how to install the client and make calls to the server. It also lists the main edit and view functions for the client program. In sections 5 and 6, we give a more detailed description of the content of the lexical units and the synsets. We try to give explicit criteria for distinguishing the different data units and their relations.

2 Global design of the database

Both DWN and RBN are semantically based lexical resources. RBN uses a traditional structure of form-meaning pairs, so-called Lexical Units (Cruse 1986). The Lexical Units are word senses in the lexical semantic tradition. They contain all the necessary linguistic knowledge that is needed to properly use the word in a language. DWN is built around the notion of synsets. Synsets are concepts as defined by Miller et al. (1991) in a relational model of meaning. Synsets are mainly conceptual units strictly related to the lexicalization pattern of a language. Concepts are defined by lexical semantic relations. For Cornetto, the semantic relations from EuroWordNet are taken as a starting point (Vossen (ed.) 1998).

Outside the lexicon, an ontology will provide a third layer of meaning. The Terms in the ontology represent the disjoint Types, organized in a Type hierarchy. A Type represents a class of entities that share the same essential properties. Types are disjoint in the sense that their members (instances) belong to only a single type. Terms can be combined in a knowledge representation language to form expressions of axioms, for example the Knowledge Interchange Format or KIF (http://logic.stanford.edu/kif/dpans.html). In principle, the meaning of concepts is defined in the ontology independently of language but according to the principles of logic (Welty and Guarino 2002). Likewise in Cornetto, the ontology represents an independent anchoring of the relational meaning in the wordnet. The ontology is a formal framework that can be used to constraint and validate the implicit semantic statements of the lexical semantic structures, both the lexical units and the synsets. In addition, the ontology provides a mapping of a vocabulary to a formal representation that can be used to develop semantic web applications.

The Cornetto database (CDB) thus consists of 3 main data collections:

1. Collection of Lexical Units (LUs), mainly derived from the RBN
2. Collection of Synsets (SYs), mainly derived from DWN

3. Collection of Terms (TEs) and axioms, mainly derived from SUMO and MILO

A fourth data type is represented by Wordnet Domains (Magnini et. al. 2002). Domains represent clusters of concepts that are related by a shared area of interest, such as Sport, Education or Politics. Whereas different instruments may belong to the same ontological Type Instrument, they may belong to different Domains. Wordnet Domains (DMs) are not represented as a separate collection. Given these data structures, we can thus have the following relations in the database:

- LU<-> SY
- SY <-> SY (within Dutch and from Dutch to English)
- SY <-> TE
- SY <-> DM

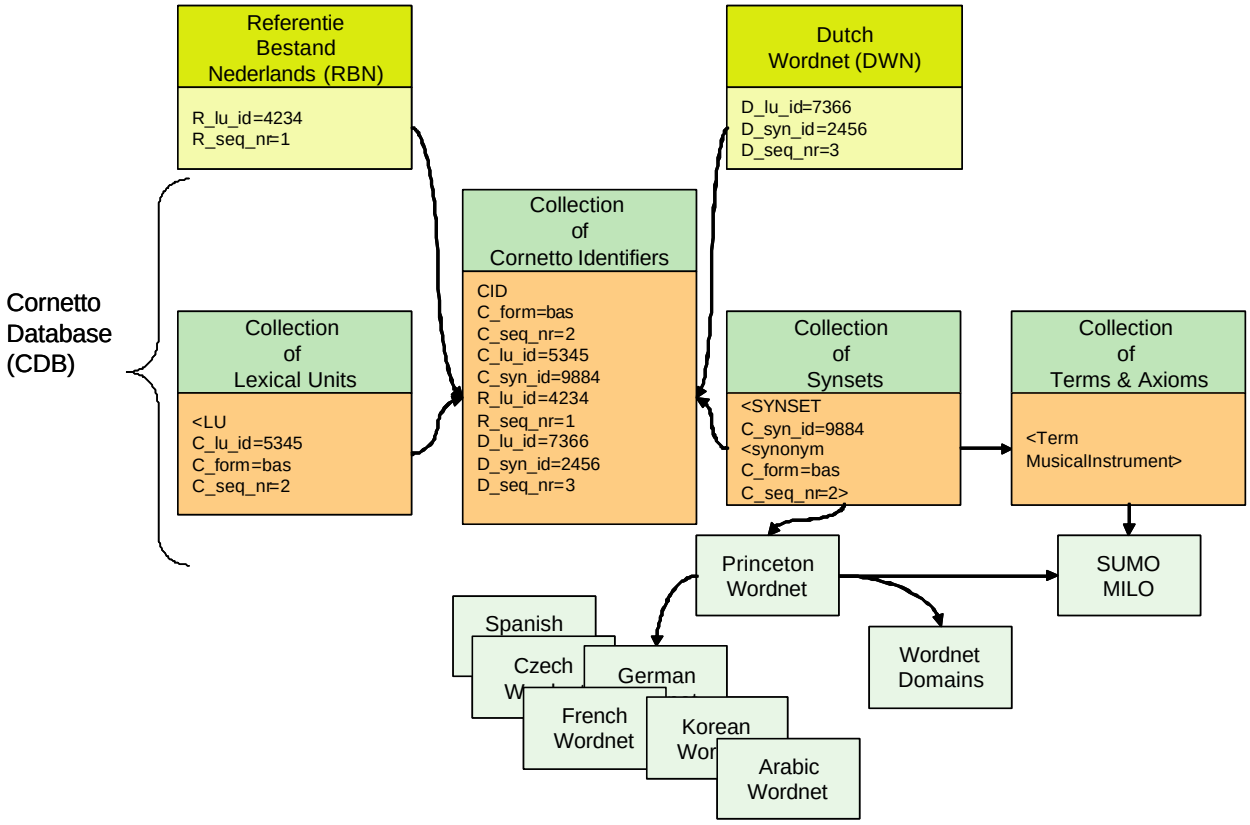

Figure 1: Overview of the Cornetto data collections

Figure-1 shows an overview of the different data structures and their relations. The different data can be divided into 3 layers of resources, from top to bottom:

- The RBN and DWN (at the top): the original database from which the data are derived;
- The Cornetto database (CDB): the ultimate database that will be built;
- External resources: any other resource to which the CDB will be linked, such as the Princeton Wordnet, wordnets through the Global Wordnet Association, Wordnet domains, ontologies, corpora, etc.

The center of the CDB is formed by the table of Cornetto Indentiefiers or CIDs. The CIDs tie together the separate collections of LUs and SYNs but also represent the pointers to the word meanings and synsets in the original databases: RBN and DWN and their mapping relation.

The direct mapping of LUs to Synsets is called macro-alignment. It involves matching the word-meaning pairs with synonyms in the synsets. In addition to macro-alignment, there is also a mapping though the implicit information expressed for each unit of information, so-called micro-alignment. Moreover, the LUs will contain frame representations, where the frame elements may have co-indexes with Synsets from the wordnet and/or with Terms from the ontology. For example, the verb *behandelen* (to treat) will have a frame element for the object which is restricted to the person that is ill or wounded. Micro-alignment then involves specifying the synsets that acts as the object of the verb behandelen: {zieke, patient}.

Micro-alignment means that any semantic constraint in the frame representation is directly related to the semantics in the other collections. Any explicit semantic relation that is expressed through a frame structure in the LU will also be represented as a conceptual semantic relation between Synsets in the Wordnet database.

The Synsets in the wordnet are represented as a collection of synonyms, where each synonym is directly related to a specific LU. The conceptual relations between Synsets are backed-up by a mapping to the ontology. This can be in the form of an equivalence relation or a subsumption relation to a Term or an expression in a knowledge representation language. Finally, a separate equivalence relation is provided to one ore more synsets in the Princeton Wordnet.

The semantic units of the Cornetto database, whether LUs or synsets, are based on the word meaning distinctions that are currently made in the RBN and DWN. An automatic alignment module, will map word meanings from RBN and DWN. The output of this program will be a mapping table representing the macro-alignment, where each word meaning in RBN is mapped to all word meanings of the same word (with the same part-of-speech) in DWN. The mappings are scored using overlapping features across the two resources.

Consider the following example. If we look at for example the noun "bas" (*bass*), we see that it has 4 meanings in both the RBN and the Dutch wordnet DWN. Together, these meanings represent at least 6 different lexical units:

1. lowest singing voice, RBN-1 & DWN-2
2. man with the voice of a bass, RBN-1 & DWN-3
3. biggest and lowest string instrument, RBN-3 & DWN-4
4. bass guitar, RBN-4
5. part of the music for the bass, DWN-1
6. bass singer or player, DWN-3

The alignment program will initially create a mapping relation between all 8 meaning units. However, only a few of these mapping will have a sufficient score. It is then possible, that some word meanings have multiple mappings and some do not have any mapping. The program will then create a minimal set of LUs and synsets that represent the word-meaning units:

1. If there is a best scoring mapping between an LU in RBN and a synonym in DWN, we create a single unique LU which becomes a synonym of a synset that is created in Cornetto. The LU receives the ID from RBN and the synset receives the ID from DWN;
2. For all remaining mappings do not create LUs and/or synsets in Cornetto but they will be stored as additional mappings that can be accessed as weighted alternatives;
3. If there is no mapping for a LU in RBN to a synonym in DWN, we create a unique LU in Cornetto that is only linked to the RBN LU and we do **not** create a synset for the LU in Cornetto;

4. If there is no mapping for a synonym in DWN to an LU in RBN, we create a synset in Cornetto with the DWN synset ID and we create a Cornetto LU with the DWN LU ID and the sparse information available in DWN;

As a consequence, we will have LUs from RBN that are not linked to synsets. These need to be manually assigned as a synonym to a synset. If there is no proper synset available, a new synset needs to be created.[1]

[1] Note that the number of different senses and consequently the sense sequence numbers in the Cornetto database may be different from both RBN and DWN.

3 The structure of the database

The Cornetto database is implemented in the Dictionary Editor and Browser (DEB II) platform (Horák et al. 2006). The database backend itself is realized in the native XML database Berkeley DB XML. The use of this native XML database allows easy import and exchange of data in a standardized data format. Furthermore, it provides powerful data access through the built-in XQuery engine.

The Cornetto database consists of 5 tables, reflecting the logical structure of the database as described in the previous chapter. The structure of each table is defined in an XML Schema file (http://www.w3.org/TR/xmlschema-0/). The definition of the Cornetto database tables thus consists of the following set of files:

- *cdb_lu.xsd* – defines the structure for storing lexical units
- *cdb_synset.xsd* – defines the structure for storing synsets
- *cid.xsd* – definines the structure for storing mappings between lexical units and synsets
- *rbn_redundancy.xsd* – defines the structure for storing redundant information from RBN
- *dwn_redundancy.xsd* – defines the structure for storing redundant information from DWN

End users of the Cornetto database will primarily access the first two tables, *cdb_lu* and *cdb_synset*. The *cid table* is primarily used for the alignment process but also links the cornetto database to RBN and DWN. The last two tables, *rbn_redundancy* and *dwn_redundancy* are exclusively used throughout for the alignment process and will not be part of the final Cornetto database. Below, we described each table in more detail.

3.1 Lexical Units

The *cdb_lu* table (Cornetto DataBase Lexical Units) is implemented as a list of lexical units. Lexical units are identified by a unique identifier *c_lu_id*. These identifiers are used to link lexical units via cid records to synsets and rbn_redundancy entries.

Cdb_lu contains structures to represent the form, syntactic and morphological information, semantics, pragmatics, and usage examples of lexical units. While the structure of form, pragmatics and usage examples is invariant, morphology, syntax, and semantic information depend on the part-of-speech of the lexical unit. This is realized as a *choice* construct. Figure 2[2] gives an overview of the table's structure.

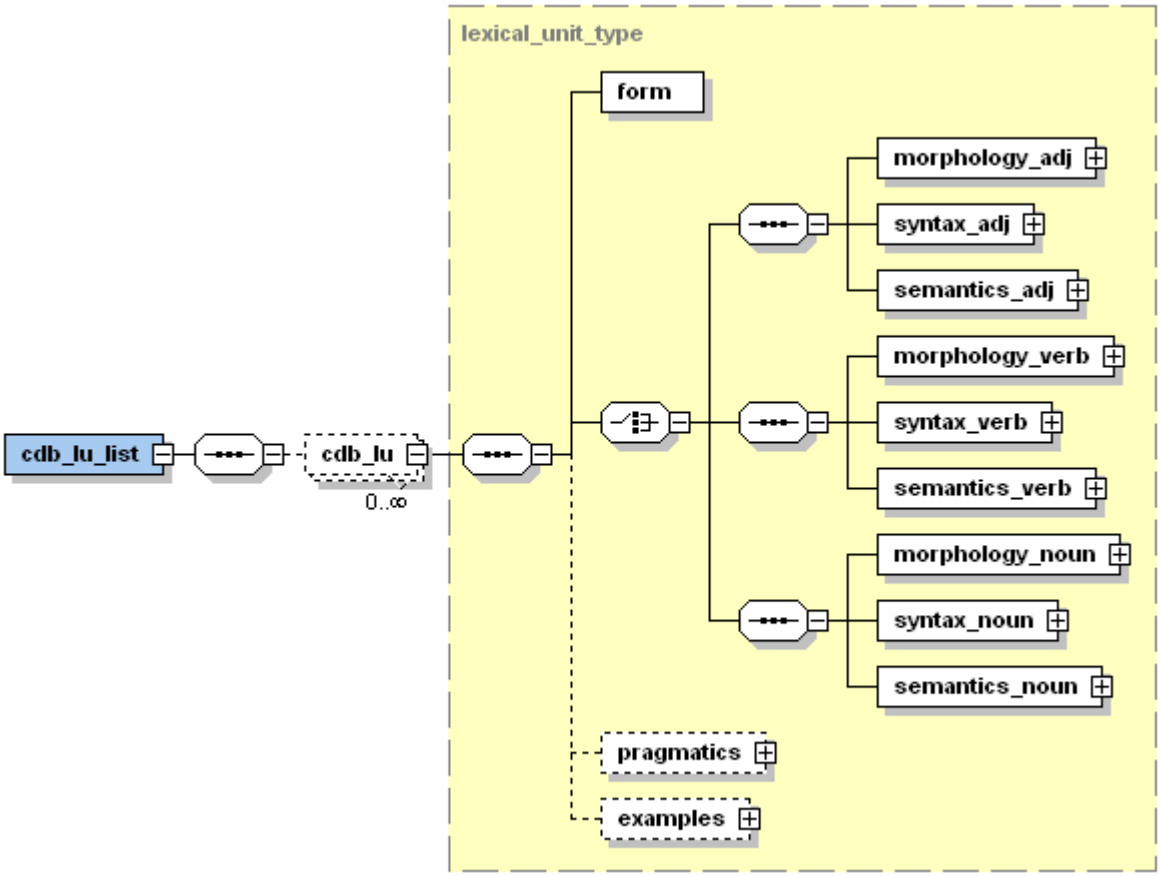

Figure 2: overview of the structure of the cdb_lu table for storing lexical units

3.2 Synsets

The table *cdb_synset* contains a list of synsets (Figure 3). Every synset is identified by unique identifier *c_synset_id*, which is used to reference synsets for example in the *cid* table. An additional attribute, *d_synset_id*, links synsets to their source concepts in DWN in order to make the lookup for the alignment process more efficient.

Each synset contains one or more synonyms. Each of these synonym entries consists of a pointer to a lexical unit (*c_lu_id*) and to a cid entry (*c_cid_id*). In addition to synonyms,

[2] Figures 2 – 5 were generated using the software package ALTOVA XML Spy.

synsets define relations between different synsets, and contain links to other semantic resources. Relations to other synsets are defined in the element *wn_internal_relations*. Each of these elements contains a pointer (*c_synset_id*) to a related synset and the name of the relationship, such as *HAS_HYPONYM* to define a hypernym relation. Links to concepts in other wordnets, in the current implementation to Princeton WordNet are included via the element *wn_equivalence_relations*. Synsets can be mapped to different versions of PWN. Finally, the elements *ontology* and *wn_domains* allow mappings of synsets to an ontology and to WordNet domains. Theoretical aspects of these mappings are discussed in more detail in the following chapters.

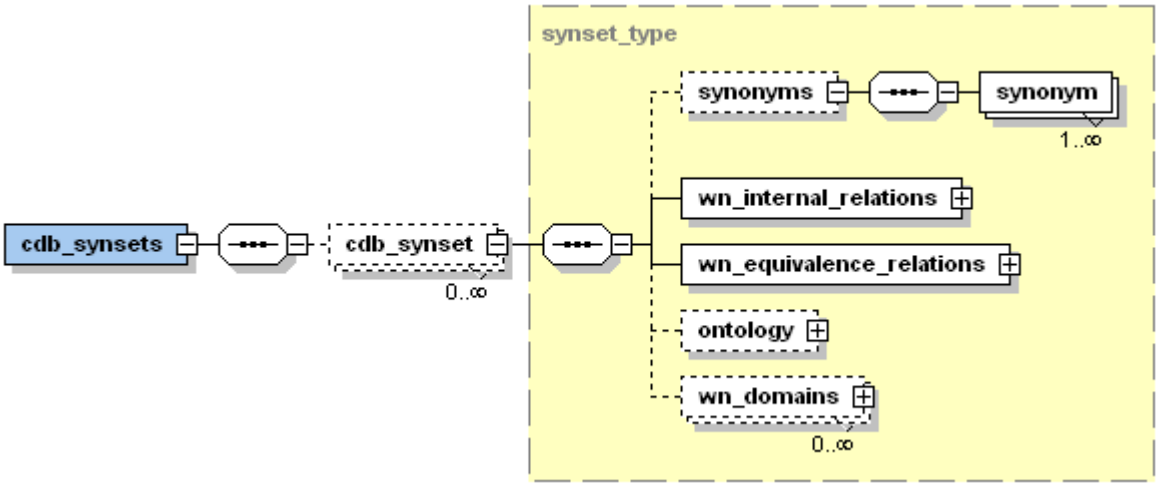

Figure 3: Structure of the table *cdb_synset*. Each synset contains a list of synonyms and relations to other synsets and external semantic resources.

3.3 CID – Mapping table

The table *cid* is a central mapping table, which links the different parts of the Cornetto database together and facilitates the alignment process between RBN and DWN. To facilitate this mapping process, *cid* entries contain all information necessary to uniquely identify records in the cornetto database, in RBN, and in DWN. For example, to uniquely identify a record in RBN, the identifier *r_lu_id* is necessary, as well as part-of-speech, and *r_seq_nr*.

The structure of *cid* entries is flat. Each entry consists of a number of attributes which constitute pointers to other database elements, such as lexical units and synsets. In addition to these pointers, *cid* entries contain meta-information about the status of the mapping, such as *score* (which is determined during automatic alignment and indicates the confidence that a

particular mapping is correct), the *date* when a mapping was created, and the *name* of the author of the mapping. Table 2 lists the attributes of *cid* entries. During the automatic alignment process, several alternative *cid* mappings may be generated. Each mapping will be associated with a *score* and the mapping with the highest score will be selected as the active mapping for a given lexical unit. During manual alignment, these mappings will be reviewed. As a result, a different candidate mapping may be selected, new mappings may be created, and wrong mappings may be deleted.

Table 1: The *cid* table links the different elements of the Cornetto database. This table is central to the automatic and manual alignment between RBN and DWN.

cid Attribute	**Description**
cid_id	unique identifier, assigned by the alignment program or by manual editing
c_lu_id	pointer to a Cornetto lexical unit (*cdb_lu* table)
pos	part-of-speech
form	word form
seq_nr	sequence number
r_lu_id	pointer to RBN and *rbn_redundancy* table
r_seq_nr	sequence number in RBN
d_lu_id	pointer to DWN and *dwn_redundancy* table
d_seq_nr	sequence number in DWN
d_sy_id	synset identifier in DWN
score	confidence score assigned by the automatic alignment program
date	date when the mapping was last changed
name	name of the person that created or last modified this record
status	Alignment status, takes one of the values: CM (confirmed manually), RM (rejected manually), CA (confirmed automatically), RA (rejected automatically), or neutral (neither confirmed nor rejected)
selected	Indicates whether this is the currently active mapping for the given *c_lu_id*

3.4 Redundancy Information

The redundancy tables *rbn_redundancy* and *dwn_redundancy* contain information that is derived from RBN and DWN respectively and will be used during the automatic and manual alignment process. As this data partially overlaps with the information in the previously

described tables, it allows validation of alignments. The data in the *rbn_redundancy* and *dwn_redundancy* tables will not be part of the final Cornetto database and will not be distributed to end users.

The structure of both tables is determined by the data requirement for the alignment process. For example, the element *synonyms* in *dwn_redundancy* contains information about synonyms defined in DWN (Figure 4). This structure mirrors the structure of the *cdb_synset* table and will allow editors to verify alignment. Additional semantic information, such as *definition* and *differentiae* will not be included in the Cornetto synsets, but again plays an important role during the alignment process. The same holds true for the elements of *rbn_redundancy*.

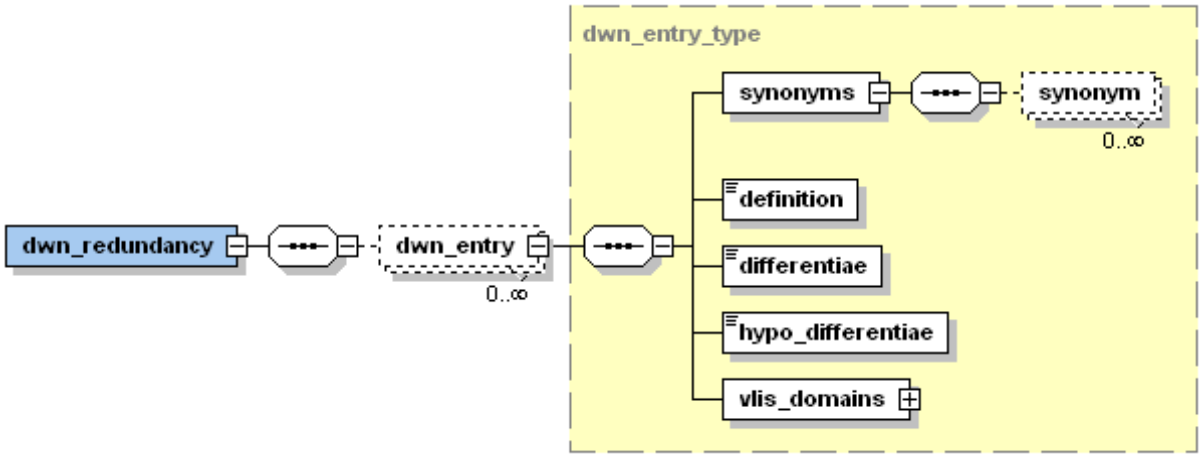

Figure 4: Structure of the table *dwn_redundancy*. This table contains data used during the alignment process that will not be part of the final Cornetto database.

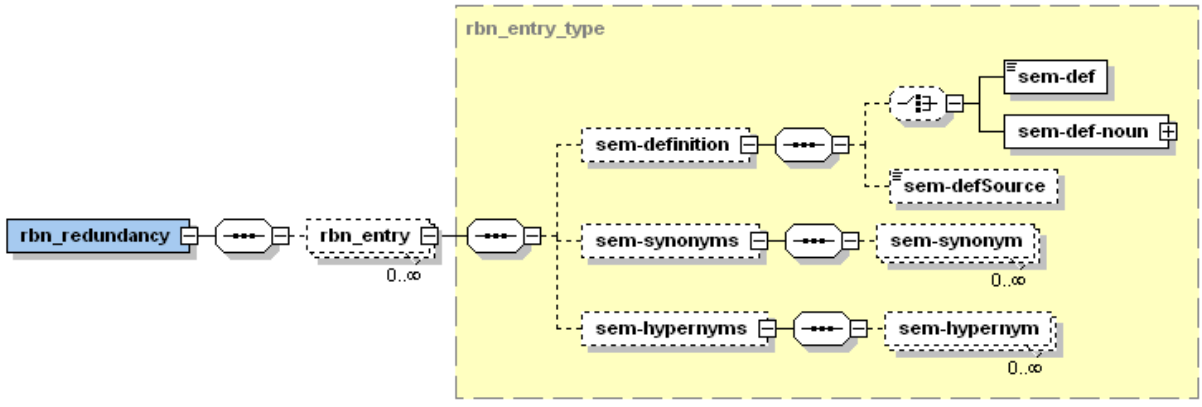

Figure 5: Structure of the table *rbn_redundancy*.

4 The database client functions

On top of this database backend, several access layers provide functionality that supports different access scenarios. Figure 6 gives an overview of the three main access scenarios: the Cornetto client, programmatic access through the Cornetto HTTP API, and direct access through the database API. Discussion in this chapter mainly focuses on the Cornetto client but also includes a short overview of the Cornetto HTTP API. For further information on directly accessing the Cornetto database through the database API, the reader is referred to existing documentation[3].

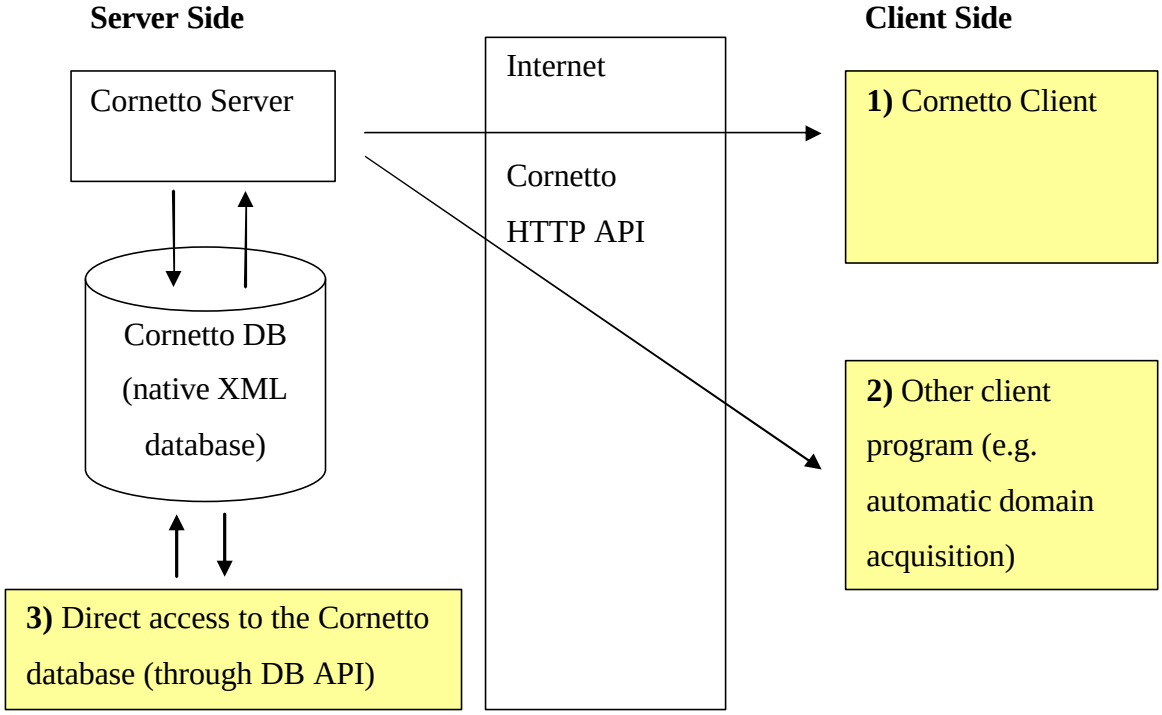

Figure 6: Cornetto supports 3 main access scenarios. The Cornetto client allows manual access for exploring and editing database content (1). Exposing the Cornetto server's HTTP API allows the efficient development of other client applications (2). Finally, the Cornetto database can be accessed directly through the database API (3).

4.1 Cornetto Client

The main purpose of the Cornetto client is to support the manual editing of the Cornetto database within the Cornetto project. In addition, the Cornetto client is valuable for users who

[3] An overview of existing online documentation can be accessed at http://www.oracle.com/technology/documentation/berkeley-db/xml/index.html

want to manually explore the database content. To support these different usage scenarios, a fine-grained permission model has been implemented. Members of the Cornetto editing team have access to information from all tables and can edit this information. Users outside of the Cornetto team can view information contained in the *cdb_lu* and *cdb_synset* tables, but cannot view redundancy information or edit any of the displayed information.

The Cornetto client is implemented as a plugin for the open-source web browser Firefox[4], based on the mozilla development platform. Mozilla is an open-source project focusing on high-quality client-server applications.

The user interface of the Cornetto client consists of three main windows:

- the *Cornetto LU* window allows search for lexical units, displays details and allows editing of lexical units
- the *Cornetto Synset* window allows search for synsets, displays details and allows editing of synsets
- the *CID* window facilitates the manual alignment by displaying the selected and alternative mappings

For browsing the content of the Cornetto database, users will mainly interact with the Cornetto LU and synset windows. The top of each window contains a search interface where users can submit queries and see a list of results. The user selects one of the search results to display further details, such as data on form, syntax and semantic information, and usage examples. If the user's access rights include edit rights for the *cdb_lu* table, then the user can also edit the selected

Figure 7: Screenshot of the Cornetto client's LU window.

[4] See http://www.mozilla.com/en-US/firefox/ for additional information regarding the Firefox web browser and the mozilla development platform.

lexical unit. Finally, the raw xml data can be displayed.

The elements of the Cornetto synset window are equivalent to those of the LU window (Figure 8). Pointers to synonyms are implemented as hyperlinks. Clicking one of these links displays the corresponding lexical unit in the Cornetto LU window. Similarly, pointers to related synsets, PWN synsets, or ontology entries are displayed as links if available.

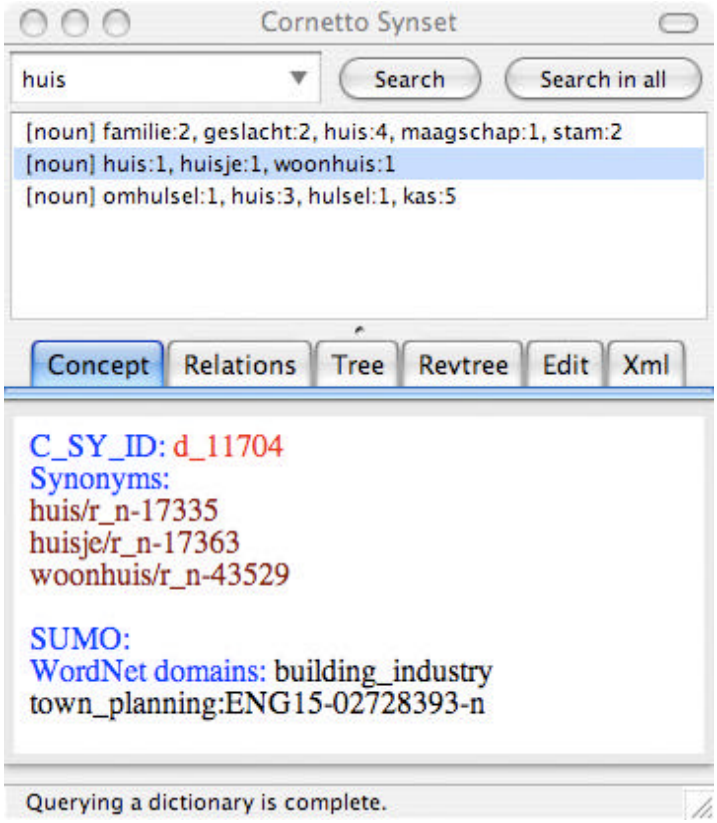

Figure 8: Screenshot of the Cornetto client's synset window.

In addition to browsing the Cornetto database, the Cornetto client will allow manual editing of the Cornetto database. The two main editing tasks will be to review CID mappings and to review sense distinctions. The user actions necessary to complete these tasks were specified in collaboration with our editing team. Table 3 gives an overview of the user actions supported by the Cornetto client.

Table 2: Overview of the two main editing tasks and corresponding user actions

Editing Task	Corresponding User Actions
Review CID mappings	- View a list of all available CID mappings - Change the status of CID mappings - Delete a wrong mapping - Select (activate) a different CID mapping - Manually create a new mapping - Find synsets for LUs without synsets (i.e., add an LU as a synonym to a synset) - Create a synset for LUs without synset - Find equivalent RBN entries for LUs not yet linked to RBN

Review sense distinctions	- Split a Lexical Unit - Merge two Lexical Units - Delete Lexical Units - Create new Lexical Units - Split a synset - Create new Synsets - Delete Synsets

4.2 Cornetto HTTP API

The Cornetto HTTP API is provided to allow integration of the Cornetto database into other applications, such as the automatic domain acquisition toolkit that will be developed in future stages of the Cornetto project.

To interact with the Cornetto server, applications send HTTP requests of the form

servername/*dictionary_code*?action=*action*&*parameters*

where the *servername* consists of the server url and port number, *dictionary_code* specifies the name of the table to be queried, *action* specifies one of the predefined server actions, and *parameters* consists a list of parameters required for each action.

An example for query to a hypothetical server with the address *https://my.example.nl:20000* could be formulated like this:

https://my.example.nl:20000/cdb_syn?action=queryList&word=huis

This request would query the synset table for a list of entries where the word form equals *huis*.

Additional queries will be implemented based on the needs of our editors, and the requirements of future project stages.

5 The content of Lexical units

In this section we will try to formulate explicit criteria for defining lexical units (LUs) in the Cornetto database, from a content perspective. These criteria are important to guide the editing of the data derived from RBN and DWN.

The structure and data of the LU is mainly derived from the RBN. We will therefore follow as much as possible the design principles from RBN for LUs. A lexical unit or LU is a form-meaning relation, such that:

- form is an abstract representation of certain realizations;
- the part-of-speech of these realizations is the same;
- the meaning of these realizations is the same, where meaning is defined by reference to a synset which in turn is defined by the distinct Terms in the ontology or KIF expressions involving Terms from the ontology;

Forms with a different part of speech (e.g. derivations) always grant separate LUs, even if the meaning is very close. Since synonyms in a synset have the same part of speech as well, these forms thus also will belong to different synsets. Since Terms in the ontology represent classes regardless of part of speech, these synsets can still be connected through the ontology.

Syntactic differences do not necessarily grant the distinction of a separate LU per se either, although they very often correlate with (systematic) differences in meaning that do grant a distinction in LUs.

Form variants with the same meaning can also exhibit pragmatic differences, where pragmatics is roughly defined as any functional usage related to register and style resulting in different distributional behaviour (domains, dialects, regions, age-groups).

The basic rule is therefore that only a difference in form and meaning grants the distinction of a separate LU. There are certain tensions with respect to this rule, discussed below.

5.1 Form variants

We abstract from certain differences in form. These variants do have a certain form relation but the meaning is the same (while pragmatics and syntax may be different). Such form variants will be stored in the same LU:

- Inflectional variants, *appel*, *appels*, *appelen*
- Spelling variants
    - Meaning is identical
    - Pronunciation is mostly identical (*droppel*, *druppel*)
    - Spelling is different but the morphology is mostly the same: spelling variation can be systematic or incidental but it is in both cases conventional.
- Shortenings:
    - Meaning is identical
    - Pragmatics is usually different
    - Pronunciation and spelling are different
    - Reduction in length for efficiency:
        - short forms (e.g. bus vs autobus)
        - abbreviation (e.g. enz.)
        - contractions (e.g.)
        - acronyms (e.g. NATO)

In general, regular derivations with compositional meaning (e.g. typical diminutives in Dutch that do not have a specialized meaning) are not included but can be added to a lookup/generation module that incorporates word formation rules.

5.2 Singleword lexical units

Singleword units have different information depending on the part-of-speech: nouns, verbs and adjectives. The main characteristic with respect their form is that they can have a simple or complex morphological structure. In the former case the form is represented by a single root morpheme, in the latter case the structure of the form is decomposed in the elements and the status of the composition (being a compound or derivation) is defined. For a detailed description of the contents of the singleword lexical unit see RBN-Manual and Vliet (2007).

5.3 Multiword lexical units

The RBN, and therefore the Cornetto database, includes ca. 85.000 Dutch word combinations ranging from free, illustrative examples to fixed multiword units like idioms and proverbs. In the RBN these combinations are not included in the macro-structure, but are given within the microstructure of one particular word contained in the expression. One of the objectives of Cornetto is to introduce part of them – the *Cornetto multiword lexical units (MWLU)* - into the macro-structure making it possible to describe them in the same way as the other, singleword, lexical units (SWLU): they will be aligned to a synset and via the synset to the ontology. The combinations which do not meet the criteria set for this new group, remain in the microstructure.

5.3.1 Multiword units in RBN

The 85.000 RBN multiword units are differentiated in 44.000 free and 41.000 fixed combinations. The latter group has been further distinguished into four subtypes based upon various degrees of fixedness and upon special characteristics with regard to their semantic, pragmatic and syntactic behaviour.

The **free** combinations have an illustrative and explanatory function: they illustrate the use of the word in a wide context and explain – in addition to the definition - the keyword in its particular meaning.
(e.g.: *morgen is er een vergadering over het bestemmingsplan van het industrieterrein* (*tomorrow's meeting will be about the zoning plan for the industrial park))*

The **fixed** combinations include fixed and semi-fixed phrases like collocations, idioms, proverbs, phrasal verbs, etc. They are characterised by features like reduced syntactic and semantic transparency, reduced compositionality and a more or less frozen status. The distinguished subtypes are:

1. Grammatical collocations (10.000):
   frequently co-occurring words forming recurrent combinations of a noun, verb or adjective with a function word.(e.g. *met betrekking tot* (*with regard to*), *te voet* (*by foot*), *uit woede (out of anger*)).
2. Lexical collocations with lexical function (15.500):
   combinations of frequently co-occurring lexical words. The collocations are further subcategorised according to their internal semantic relations using a set of lexical functions (loosely based upon Melcuk).(e.g. *zware roker (heavy smoker)*; *een vergadering schorsen (adjourn a meeting))*
3. Lexical collocations – not specified (4.500):[5]
   lexical collocations that cannot be expressed with a Melcuk function. In many cases one of the combination words is not used in its literal meaning or has got a more specific meaning than its basic meaning and therefore the combination has a reduced semantic transparency. (*systematische catalogus (systematic catalogue), open breuk (compound fracture), enkelvoudige breuk (simple fracture)).*
4. Pragmatic formula (1500):
   highly fixed, fully compositional expressions which are associated with a fixed communicative situation. (*goedemorgen (good morning), geen paniek! (no panic), adres onbekend (address unknown))*
5. Idioms (9500):
   expressions with a reduced or lack of semantic transparency ranging from partly to fully uncompositional (e.g. *de kaarten zijn geschud, stoken in een goed huwelijk, een heilige koe, een vette kluif*)
6. Proverbs (350):
   completely frozen sentences (*e.g. Pas als het kalf verdronken is, dempt men de put)*

[5] In RBN a considerable part of these expressions have been encoded as 'qualifiers'.

5.3.2 Multiword lexical units in Cornetto

Within the framework of Cornetto we focus on the multiword units which have a reduced semantic (and often syntactic) transparency and a reduced or lack of compositionality. This criterion is met by about 15.000 expressions:

Idioms (9500)

Proverbs (350)

Lexical collocations- qualifiers (5000)

It is not yet decided whether we take into consideration all these multiword units or not. In particular, those that have a surface form not conform to the syntactic categories included in Cornetto, i.e. nouns, verbs and adjectives, will be difficult to integrate. For instance, complete sentences , like proverbs, express concepts and have semantic relations which will be quite different from the concepts and relations expressed by other syntactic categories. As a consequence, only phrases that function as nouns, verbs, adjectives and adverbs would be selected.

5.3.3 Contents of the Cornetto Multiword lexical units

The description of the multiword LU (table 4) focuses on the description of the whole expression: each entry consists of a canonical form, its syntactic category, a textual form (if this applies), a lexicographic definition, information regarding its use if needed, one or more examples of the construction in context. The link to the synset is realised by a pointer to a cid-entry (c_cid_id) and links to the indivual words of the combination are realised by pointers to single word lexical units (c_lu_id) . Information relative to the individual words is included in the description of those particular words. Each multiword LU can be illustrated by one or more example units.

table 4: the contents of a multiword lexical unit

attribute	**Value**	**Description**
Canonicalform		basic, non-inflected form
Category	NP, VP, AP	Syntactic category of canonical form
Textualform		non-canonical form
Text-category		Syntactic category of textual form
Sy-type	fixed	
Sy-subtype	lexical collocation, idiom	
Sem-		Definition (esp. for

meaningdescription			idioms)
Pragmatics			Domain/Connotation/style/ etc.
M	*C_LU_ID*		Pointer to single word LU
M	*C_CID_ID*		Pointer to cid-table
M	*Examples*		Illustrative 'free' examples of the multiword lexical unit in context.

M = multiple field

The pointers to the individual words are pointers to lexical units, i.e. to form-meaning combinations. This seems contradictory – and sometimes is- with the uncompositionality of the multiword units. However, many multiword units are only semi-transparent and their syntactic and semantic behaviour is often related to their individual parts.

5.3.4 Examples

iemand op weg helpen (help someone get started)

CDB-LexicalUnit	
Canonicalform	iemand op weg helpen
Category	VP
Textualform	zij helpt hem op weg (*she helps him get started)*
Text-category	S
Sy-type	Fixed
Sy-subtype	Idiom
Sem-meaningdescription	Iemand bijstaan bij het beginnen van iets (*help someone to start something)*
C_LU_ID	weg (N)
C_LU_ID	helpen (V)
C_CID_ID	[CDB-synset]
Example	Uw verzekeringsadviseur kan u verder op weg helpen. (*your insurance expert may help you get started*)
CDB-Synset	
Synset	[iemand op weg helpen, iemand op dreef helpen, iemand op gang helpen]
Hypernym	[helpen, bijstaan, ondersteunen, van dienst zijn, begeleiden](*to help)*
Cause	[starten, beginnen aanvangen] *(to start)*

burgerlijk huwelijk (*civil marriage)*

Canonicalform	burgerlijk huwelijk
Category	NP
Sy-type	Fixed
Sy-subtype	lexical collocation
Sem-meaningdescription	

Pragmatics	*Domain*	juridisch
C_LU_ID		huwelijk (N)
C_LU_ID		burgerlijk (A)
C_CID_ID		[CDB-synset]
CDB-Synset		
Synset		[burgerlijk huwelijk]
Hypernym		[huwelijk, huwelijksvoltrekking,]
Near_antonym		[kerkelijk huwelijk]

blinde muur (*blank wall)*

Canonicalform	blinde muur
Category	NP
Sy-type	fixed
Sy-subtype	idiom
Sem-meaningdescription	muur zonder ramen of deuren (*a wall unbroken by windows or other openings*)
C_LU_ID	muur (N) (*wall*)
C_LU_ID	blind (A) (*blind*)
C_CID_ID	[CDB-synset]
Example	ze kijken uit op een blinde muur
CDB-Synset	
Synset	[blinde muur]
Hypernym	[buitenmuur]

roomser dan de paus (more Catholic than the Pope)_

CanonicalForm		roomser dan de paus
Category		AdjP
TextualForm		hij is roomser dan de paus
TextCategory		S
Sy-type		fixed
Sy-subtype		idiom
Sem-meaningdescription		uiterst principieel, overdreven principieel
Pragmatics	*Connotation*	pejorative
C_LU_ID		rooms (A)
C_LU_ID		paus (N)
C_CID_ID		[CDB-synset]
Example		niet dat ik nou roomser dan de paus wil zijn, maar voor een antenne heb je wel een vergunning nodig
CDB-Synset		
Synset		[roomser dan de paus]
Hypernym		[principieel, beginselvast]

6 The content of Synsets

The synsets structure in the Cornetto database is directly copied from the structure of EuroWordnet. Likewise, it consists of a list of synonyms, a list of language-internal relations and a list of equivalence-relations to Princeton WordNet. In addition, we added a field to map the Synset to the ontology and a field to store one or more domain labels (taken from Wordnet Domains).

A Synset is the defined as a set of synonyms (represented by LUs) that refer to the same entities in most contexts:

- further defined by lexical semantic relations;
- and defined by reference to ontology Terms or KIF expressions involving Terms from the ontology;

As such the synsets do both represent a language-internal system of relations and have a mapping to some external formal definition of 'reality' provided by the ontology. Note that semantic relations can thus be found in both the wordnet and the ontology. This is not a problem as long as these relations are not in conflict with each other. It may very well be that hyperonym relations in wordnet are not found in the ontology simply because that level of hyponymy is not present in the more compact Type hierarchy of the ontology.

6.1 Synonyms

Synonymy is defined a very loose way in WordNet (Fellbaum 1998): words that can be used to refer to the same entities in most context. There is no further definition of what these contexts are or how many are needed. In Cornetto, we do not further define this notion either. We will basically take over the synonyms as they are given in the Dutch wordnet. We believe that it is more useful to define all the other relations and information more comprehensively. This then should provide the knowledge to have a more proper definition about synonymy, with evidence on the properties of each word meaning and the synset that are given by the database.

Nevertheless, it is important to realize that the difference in approach between RBN and DWN leads to certain tensions between what establishes a different word meaning and likewise a synset. In general RBN tends to have more abstract and general meanings, assuming that they can be specialized. DWN has more explicit meanings because the specializations may represent concepts for which there is a synonym. Consider for example, the adjective “helder” (clear). In one of its meanings, it can be predicated of liquids, gasses, substances. However, when applied to the sky “helder” becomes synonymous to “onbewolkt” (cloudless), which can only be applied to skies. The synset with “onbewolkt” thus forces one to differentiate a separate meaning for “helder”. In general, we believe that by formalizing relations, ontological structures and criteria, a better view on different word meanings and synonymy can be developed on the long run.

6.2 Language internal relations

The language-internal relations are taken from EuroWordNet.

Relation Type	Parts of Speech	Labels	Data Types
NEAR_SYNONYM	N<>N, V<>V		Syn <>Syn
XPOS_NEAR_SYNONYM	N<>V, N<>AdjAdv, V<>AdjAdv		Syn <>Syn
HAS_HYPERONYM	N>N, V>V	dis, con	Syn <>Syn
HAS_HYPONYM	N>N, V>V	dis	Syn <>Syn
HAS_XPOS_HYPERONYM	N>V, N>AdjAdv, V>AdjAdv, V>N, AdjAdv>N, AdjAdv>V	dis, con	Syn <>Syn
HAS_XPOS_HYPONYM	N>V, N>AdjAdv, V>AdjAdv, V>N, AdjAdv>N, AdjAdv>V	dis	Syn <>Syn
HAS_HOLONYM	N>N	dis, con, rev, neg	Syn <>Syn
HAS_HOLO_PART	N>N	dis, con, rev, neg	Syn <>Syn
HAS_HOLO_MEMBER	N>N	dis, con, rev, neg	Syn <>Syn
HAS_HOLO_PORTION	N>N	dis, con, rev, neg	Syn <>Syn
HAS_HOLO_MADEOF	N>N	dis, con, rev, neg	Syn <>Syn
HAS_HOLO_LOCATION	N>N	dis, con, rev, neg	Syn <>Syn
HAS_MERONYM	N>N	dis, con, rev, neg	Syn <>Syn
HAS_MERO_PART	N>N	dis, con, rev, neg	Syn <>Syn
HAS_MERO_MEMBER	N>N	dis, con, rev, neg	Syn <>Syn
HAS_MERO_MADEOF	N>N	dis, con, rev, neg	Syn <>Syn
HAS_MERO_LOCATION	N>N	dis, con, rev, neg	Syn <>Syn
ANTONYM	N<>N, V<>V		Syn <>Syn
NEAR_ANTONYM	N<>N, V<>V		Syn <>Syn
XPOS_NEAR_ANTONYM	N<>V, N<>AdjAdv, V<>AdjAdv		Syn <>Syn
CAUSES	V>V, N>V, N>N, V>N, V>AdjAdv, N>AdjAdv	dis, con, non-f, rev , neg	Syn <>Syn
IS_CAUSED_BY	V>V, N>V, N>N, V>N, AdjAdv>V, AdjAdv>N	dis, con, non-f, rev, neg	Syn <>Syn
HAS_SUBEVENT	V>V, N>V, N>N, V>N	dis, con, rev, neg	Syn <>Syn

IS_SUBEVENT_OF	V>V, N>V, N>N, V>N	dis, con, rev, neg	Syn <>Syn
ROLE	N>V, N>N, AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
ROLE_AGENT	N>V, N>N	dis, con, rev, neg	Syn <>Syn
ROLE_INSTRUMENT	N>V, N>N	dis, con, rev, neg	Syn <>Syn
ROLE_PATIENT	N>V, N>N	dis, con, rev, neg	Syn <>Syn
ROLE_LOCATION	N>V, N>N, AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
ROLE_DIRECTION	N>V, N>N, AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
ROLE_SOURCE_DIRECTION	N>V, N>N, AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
ROLE_TARGET_DIRECTION	N>V, N>N, AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
ROLE_RESULT	N>V, N>N	dis, con, rev, neg	Syn <>Syn
ROLE_MANNER	AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
INVOLVED	V>N, N>N, V>AdjAdv, N>AdjAdv	dis, con, rev, neg	Syn <>Syn
INVOLVED_AGENT	V>N, N>N	dis, con, rev, neg	Syn <>Syn
INVOLVED_PATIENT	V>N, N>N	dis, con, rev, neg	Syn <>Syn
INVOLVED_INSTRUMENT	V>N, N>N	dis, con, rev, neg	Syn <>Syn
INVOLVED_LOCATION	V>N, N>N, V>AdjAdv, N>AdjAdv	dis, con, rev, neg	Syn <>Syn
INVOLVED_DIRECTION	V>N, N>N, V>AdjAdv, N>AdjAdv	dis, con, rev, neg	Syn <>Syn
INVOLVED_SOURCE_DIRECTION	V>N, N>N, V>AdjAdv, N>AdjAdv	dis, con, rev, neg	Syn <>Syn
INVOLVED_TARGET_DIRECTION	V>N, N>N, V>AdjAdv, N>AdjAdv	dis, con, rev, neg	Syn <>Syn
INVOLVED_RESULT	V>N, N>N	dis, con, rev, neg	Syn <>Syn
CO_ROLE	N>N	rev	Syn <>Syn
CO_AGENT_PATIENT	N>N	rev	Syn <>Syn
CO_AGENT_INSTRUMENT	N>N	rev	Syn <>Syn
CO_AGENT_RESULT	N>N	rev	Syn <>Syn
CO_PATIENT_AGENT	N>N	rev	Syn <>Syn
CO_PATIENT_INSTRUMENT	N>N	rev	Syn <>Syn
CO_PATIENT_RESULT	N>N	rev	Syn <>Syn
CO_INSTRUMENT_AGENT	N>N	rev	Syn <>Syn
CO_INSTRUMENT_ PATIENT	N>N	rev	Syn <>Syn
CO_INSTRUMENT_RESULT	N>N	rev	Syn <>Syn
CO_RESULT_AGENT	N>N	rev	Syn <>Syn
CO_RESULT_PATIENT	N>N	rev	Syn <>Syn
CO_RESULT_INSTRUMENT	N>N	rev	Syn <>Syn
IN_MANNER	V>AdjAdv, N>AdjAdv	dis, con, rev, neg	Syn <>Syn
MANNER_OF	AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
BE_IN_STATE	N>AdjAdv, V>AdjAdv	dis, con, rev, neg	Syn <>Syn
STATE_OF	AdjAdv>N, AdjAdv>V	dis, con, rev, neg	Syn <>Syn
FUZZYNYM	N<>N, V<>V		Syn <>Syn
XPOS_FUZZYNYM	N<>V, V<>AdjAdv, N<>AdjAdv		Syn <>Syn

For a complete explanation of these relations, we refer to the EuroWordNet documentation: http://www.illc.uva.nl/EuroWordNet/finalresults-ewn.html.

We made two changes with respect to these relations due to requirements of the DebVisdic system:

- Hyponymy relations are encoded in one direction only, from hyponym to their hyperonym
- Features of the hyponymy relations are stored in two directions

In EuroWordNet, every relation has a reverse counterpart. Partly, this was motivated by the requirements of the database system, and partly, to differentiate the exact implication of the relation. In the case of hyponymy, it is thus possible to make a difference between the following relations:

mammal:

has_hyponym+disjunct dog

has_hyponym+disjunct cat

pet:

has_hyponym dog

has_hyponym cat

dog:

has_hyperonym+coordinate mammal

has_hyperonym+coordinate pet

cat

has_hyperonym+coordinate mammal

has_hyperonym+coordinate pet

knife

has_hyperonym+coordinate cutlery

has_hyperonym+coordinate weapon

weapon:

has_hyponym+disjunct knife

has_hyponym+disjunct sword

cutlery:

has_hyponym+disjunct knife

has_hyponym+disjunct fork

Disjointness of hyponymy relations is directional. The hyponyms of mammals are disjoint because it is a rigid class (see below). The reverse hyperonym relation is coordinated, since some of the mammals can also be pets at the same time.

For efficiency reasons, the hyponymy relation is stored in only one direction in DebVisDic. To be able to still store the features of the relations in two directions, we doubled the features for the hyperonyms. In addition to coordinative, disjunctive, reversed, factive and negative, we also have oppositecoordinative, oppositedisjunctive, oppositereversed, oppositefactive and oppositenegative. In the case of the above dog example, the following relations will then be encoded:

dog:

has_hyperonym+coordinate+oppositedisjunctive mammal

has_hyperonym+coordinate+oppositedisjunctive pet

This should be read as: mammal and pet being coordinated for dog, but dog being a disjunctive hyponym for both mammal en pet.

6.3 Equivalence relations to English Wordnet

Equivalence relations are also taken over as they are in EuroWordNet. The original Dutch wordnet was linked to WordNet 1.5. In DebVisDic, WordNet 2.0 is loaded. The mappings have therefore been translated from WordNet1.5 to WordNet2.0, using the mapping tables provided in the MEANING project (ist-2001-34460: http://www.lsi.upc.es/~nlp/meaning/). The equivalence relations in the database therefore have the format:

Equivalence relations:
[EQ_NEAR_SYNONYM] binocular:1, binoculars:1, field glasses:1, opera glasses:1/ENG20-02741178-n
version: pwn_1_5 score: 8417.0 source_id: HEURISTICS_BI

[EQ_NEAR_SYNONYM]field glass:1, glass:4, spyglass:1/ENG20-03211815-n
version: pwn_1_5 score: 4252.0 source_id: HEURISTICS_BI

[EQ_NEAR_SYNONYM]glass:2, drinking glass:1/ENG20-03312858-n
version: pwn_1_5 score: 2616.0 source_id: HEURISTICS_BI

[EQ_NEAR_SYNONYM]looking glass:1, glass:6/ENG20-03551443-n
version: pwn_1_5 score: 2686.0 source_id: HEURISTICS_BI

[EQ_NEAR_SYNONYM]telescope:1, scope:3/ENG20-04233378-n
version: pwn_1_5 score: 2835.0 source_id: HEURISTICS_BI

[EQ_NEAR_SYNONYM]glass:3, glassful:1/ENG20-12990841-n
version: pwn_1_5 score: 958.0 source_id: HEURISTICS_BI

[EQ_NEAR_SYNONYM]glass:1/ENG20-14033732-n
version: pwn_1_5 score: 1983.0 source_id: HEURISTICS_BI

They consist of pointer to WordNet 2.0 with a reference to the method and score from which they originated. The scores shown here are thus scores taken from the mapping provided to WordNet1.5 in the Dutch wordnet.

6.4 Ontology

A new relation is the mapping from the synset to the ontology. The ontology is seen as an independent anchoring of concepts to some formal representation that can be used for reasoning. Within the ontology, Terms are defined as disjoint Types, organized in a Type hierarchy where:

- a Type represents a class of entities that share the same essential properties.
- Instances of a Type belong to only a single Type: => disjoint (you cannot be both a *cat* & a *dog*)

Terms can further be combined in a knowledge representation language to form expressions of axioms *(you can be a watch dog & a bull dog)*, i.e. the Knowledge Interchange Format, KIF, based on first order predicate calculus and primitive elements.

Following the OntoClean method (Guarino and Welty (2002a, 2002b), identity criteria can be used to determine the set of disjunct Types. These identity criteria determine the essential properties of entities that are instances of these concepts:

- **Rigidity**: to what extent are properties of an entity true in all or most worlds? E.g., a *man* is always a *person* but may bear a Role like *student* only temporarily. Thus manhood is a rigid property while studenthood is anti-rigid.
- **Essence**: which properties of entities are essential? For example, "shape" is an essential property of "vase" but not an essential property of the clay it is made of.
- **Unicity**: which entities represent a whole and which entities are parts of these wholes? An "ocean" or "river" represents a whole but the "water" it contains does not.

The identity criteria are based on certain fundamental requirements. These include that the ontology is descriptive and reflects human cognition, perception, cultural imprints and social conventions (Masolo, Borgo, Gangemi, Guarino, and Oltramari 2003).

The work of Guarino and Welty (2002a, 2002b) has demonstrated that the WordNet hierarchy, when viewed as an ontology, can be improved and reduced. For example, roles such as AGENTS of processes are anti-rigid. They do not represent disjunct types in the ontology and complicate the hierarchy.

As an example, consider the hyponyms of dog in WordNet, which include both types (races) like *poodle*, *Newfoundland*, and *German shepherd*, but also roles like *lapdog*, *watchdog* and *herding dog*. "Germanshepherdhood" is a rigid property, and a German shepherd will never be a Newfoundland or a poodle. But German shepherds may be herding dogs. The ontology would only list the **rigid** types of dogs (dog races): Canine => PoodleDog; NewfoundlandDog; GermanShepherdDog, etc.

The lexicon of a language then may contain words that are simply names for these types and other words that do not represent new types but represent roles (and other conceptualizations of types). For example, English *poodle*, Dutch *poedel* and Japanse *pudoru* will become simple names for the ontology type: ⇔ ((instance x PoodleDog). On the other hand, English *watchdog*, the Dutch word *waakhond* and the Japanese word *banken* will be related through a KIF expression that does not involve new ontological types: ⇔ ((instance x Canine) and (role

x GuardingProcess)), where we assume that GuardingProcess is defined as a process in the hierarchy as well. The fact that the same expression can be used for all the three words indicates equivalence across the three languages.

In a similar way, we can use the notions of **Essence** and **Unicity** to determine which concepts are justifiably included in the type hierarchy and which ones are dependent on such types. If a language has a word to denote a lump of clay (e.g. in Dutch *kleibrok* denotes an irregularly shaped chunk of clay), this word will not be represented by a type in the ontology because the concept it expresses does not satisfy the Essence criterion. Similary a word like *river water* (Dutch *rivierwater*) is not represented by a type in the onotology as it does not satisfy Unicity; such words are dependent on valid types. Satisfying the rigidity criteria, for example, is a condition for type status.

From this basic starting point, we can derive two types of mappings from synsets to the ontology:

- Synsets represent disjunct types of concepts, where they are defined as:
  a. names of Terms;
  b. subclasses of Terms, in case the equivalent class is not provided by the ontology
- Synsets represents non-rigid conceptualizations, which are defined through a KIF expression;

The KIF expressions are currently restricted to triplets consisting of the relation name, a first argument and a second argument. The operator of the triplets is AND, and we assume existential quantification of any of the variables, specified as a value of the arguments. Furthermore, we follow the convention to use a zero as the variable the corresponds to the denotation of the synset being defined and any other integer for other denotations.

When we look at the different dogs in the Dutch wordnet then we see 3 types of hyponyms:

- bokser; corgi; loboor; mopshond; pekinees; pointer; spaniël;
- pup; reu; teef
- bastaard; straathond; blindengeleidehond; bullebijter; diensthond; gashond; jachthond (hunting dog); lawinehond; schoothondje (lap dog);waakhond (watch dog)

The first group are names for doc races that are clearly rigid and disjunct. They represent names for Terms. The second group words for male/female and baby dogs. They can be encoded in the same way as man, woman and child for humans. The third group refers to dogs with certain non-rigid attributes. They will thus not represent names for types but are related to the ontology by a mapping to the term Canine and the attribute that applies.
Another case of mixed hyponyms are words for *water*. In the Dutch wordnet there are over 40 words that can be used to refer to water in specific circumstances or with specific attributes.

Water is in SUMO a CompoundSubstance just as other molecules. We can thus expect that the synset of *water* in Dutch matches directly to Water in SUMO, just as *zand* matches to Sand. However, *water* has 3 major meaning in the Dutch wordnet: water as liquid, water as a chemical element and a water area, while there are only two concepts in SUMO: Water as the CompoundSubstance and a WaterArea. In SUMO there is no concept for water in its liquid form, even though this is the most common concept for most people. Most of the hyponyms of *water* in the Dutch wordnet are linked to the liquid. To properly map them to the ontology, we thus first must map water as a liquid. This can be done by assigning the Attribute Liquid to the concept of Water as a CompoundSubstance:

(and
(exists ?L ?W)
(instance, ?W, Water) ,
(instance, ?L LiquidState
(hasAttributeinstance, ?W, ?L)
)

The hyponyms of water in the Dutch wordnet can be divided into 3 groups:

- Water used for a purpose: theewater (*for making tea*), koffiewater (*for making coffee*), bluswater (*for extinguishing fire*), scheerwater (*for shaving*), afwaswater (*for cleaning dishes*), waswater (*for washing*), badwater (*for bading*), koelwater (*for cooling*), spoelwater (*for flushing*), drinkwater (*for drinking*)
- Water occurring somewhere or originating from: putwater (*in a well*), slootwater (*in a ditch*), welwater (*out of a spring*), leidingwater, gemeentepils, kraanwater (*out of the tap*), gootwater (*in the kitchen sink or gutter*), grachtwater (*in a canal*), kwelwater

(*coming from underneath a dike*), grondwater, grondwater (*in the ground*), buiswater (*on a ship*)

- Being the result of a process: pompwater (*being pumped away*), smeltwater, dooiwater (*melting snow and ice*), afvalwater (*waste water*), condens, condensatiewater, condenswater (*from condensation*), lekwater (*leaking water*), regenwater (*rain water*), spuiwater (*being drained for water maintenance*)

Here are some KIF expressions for these concepts:

```
theewater (tea water)
 (exists (?A ?W)
        (and
                (instance ?W Water)
                (hasPurposeForAgent ?W
                (exists (?T)
                        (and
                        (instance ?T Tea)
                        (part ?W ?T)
                 )) ?A)))

bluswater (water for extinguishing fire)
(exists (?A ?W)
    (and
        (instance ?W Water)
        (hasPurposeForAgent ?W
                (exists (?E)
                (and
                (instance ?E Extinguishing)
                (resource ?E ?W))) ?A))

drinkwater (drinking water)
    (exists (?W ?A)
      (and
        (instance ?W Water)
```

(capability Drinking resource ?W)))

putwater (water at the bottom of well)

(exists (?W ?M)
(and
(instance ?W Water)
(instance ?M MineOrWell)
(located ?W ?M)))

slootwater (*in a ditch*)

(exists (?W ?S)
(and
(instance ?W Water)
(instance ?S SmallStaticWaterArea)
(part ?W ?S)))

leidingwater, gemeentepils, kraanwater (*out of the tap*)

(exists (?W ?R ?F)
(and
(instance ?W Water)
(instance ?F Faucet(=Device))
(instance ?R Removing)
(origin ?R ?F)
(patient ?R ?W)))

Through the complex mappings of non-rigid synsets to the ontology, the latter can remain compact and strict.

7 Mapping relations between lexical units, synsets and terms

7.1 Meaning variants

Some derivational variants present difficult cases both as form variants and meaning types. For Dutch, this specifically relates to:

- male/female variants: leraar (male teacher) and lerares (female teacher)
- diminutives: kop*je* (a small cup), bier*tje* (a beer).
- complex verbs: *door*praten, *door*lopen, *plat*drukken, *plat*slaan, *plat*duwen.
- nominalizations: *handeling* (action) and *handelen* (to act)

The phenomena can be observed in many other languages as well. The question here is whether these derivates represent different Types in the ontology. We will discuss each of these below to clarify our point of view.

7.1.1 Male and female variants

Male/female variants are typically found for professions:

- theoloog/theologe
- leraar/lerares
- secretaresse vs. mannelijke secretaresse
- kleuterleidster vs. ?

The main question is: are these different semantic types? We can define the meaning of professions as abstract roles that can be applied to both men and women. The male/female form can then be derived from the neutral form by means of a derivational rule with a corresponding restriction in the denotation. This is not restricted to professions, compare for example: “albino” (animal, human, plant) and “relative” (animal, human).

As explained above, our basic point of view is that the ontology defines the disjunct types by their list of Terms. So if the ontology would make a distinction between the types

MaleTheologist and FemaleTheologist, then we must make a distinction between the corresponding LUs and synsets as well. We call this solution 1:

1. The forms “theoloog” and “theologe” represent distinct LUs that have an equivalence mapping to the distinct synsets. Each of these synsets will be linked to a separate ontology Term:

   LU “theoloog” (GENDER neutral form), synonym of a synset that has an equivalence relation to the Term MaleTheologist

   LU “theologe” is a synonym of a different synset which has an equivalent mapping to the term FemaleTheologist

However, we expect the ontology to be minimalistic in the definition of Terms and therefore define professions as abstract roles that can be combined with other disjunct Terms such as Man and Woman. In that case, we have 2 possibilities:

2. The forms “theoloog” and “theologe” represent a single LU from which we can derive the male and female form variants with derivational rules (zero-derivation for the male variant):

   LU “theoloog” is synonym in a synset, which is equivalent to the Term Theologist

   In this solution, the derivational rule needs to make explicit what form-variant is used for male and female instances of the concept Theologist

3. The forms “theoloog” and “theologe” still represent distinct LUs that also represent different synsets. Each synset is then defined with a complex KIF expression that combines the role with the type Man and Woman:

   LU “theoloog” is a synonym of a synset that is defined by the KIF expression: (and (?0 Man) (?0 Theologist))

> LU: “theologe” is a synonym of another synset that is defined by the KIF expression: (and (?0 Woman) (?0 Theologist))

In Cornetto, we will apply the following principle:

> If the meaning can be defined exhaustively as a role that is neutral with respect to male/female AND if the male/female form can be derived with a regular and compositional derivation then we will distinguish 1 synset for the abstract neutral form (solution 2) above. In all other cases, we will distinguish separate LUs and synsets possibly related to different ontology terms (solution 1) or KIF expressions (solution 3), depending on the ontology.

According to this principle, we will have a single LU in Dutch for *theoloog*. When applied to a man, we will use the neutral form (possibly zero-derivation). When applied to a woman, we can choose to use the neutral form or apply a derivational rule to create *theologe*.

In the case of *leraar* (male teacher) and *lerares* (female teacher), there is no neutral form. The principle then prescribes to distinguish two separate LUs with a corresponding explicit meaning. The same applies to *verpleger* (male nurse) and *verpleegster* (female nurse).

Note that the derivational component still needs to ‘*know*’ when the female form can be generated and when not. There are cases such as *arts* where we do not generate a female form where we do with others (*docente*, *dichteres*, *kapper/kapster*), even when there is a neutral form.

7.1.2 Diminutives

The use of the diminutive is very common in Dutch. In some cases, the form and meaning are regular, in other cases it is not:

a) meis*je* (girl): there is no basic form *meis* and the meaning is neutral
b) kop (cup ,*neutral*) and kop*je* (small cup, and the usage is modest, polite)
c) bier (a beer) and biertje (a beer with normal size, but usage is informal)
d) trein (train) and treintje (small train, meaning is small)

Case a) is a separate LU and synset. There is no other variant. Case d) is completely regular. We do not include the diminutive form. Case b) refers to a smaller size of cup. It looks regular such as d) but the usage, pragmatics is more restricted. This grants the addition to the database as a separate form in which case we must add a separate synset which will be a hyponym of the regular, general concept kop (cup). However, it is not a different type of object and therefore we do no create a new ontology Term but make a KIF expression: (and (instance 0? Cup) (hasAttribute0? Small).

In the case of c), we have a synset that include the basic form "bier" as well. There is no difference in meaning across "bier" and the diminutive form "biertje", at most a pragmatic difference. These thus do represent different LUs but a single synset and a single concept in the ontology, which is defined as a portion of the drink *Beer*, contained in a container (a Cup). The pragmatic difference can be captured in the information for the LU.

7.1.3 Complex verbs

Dutch is a compounding language and as such has many verb compounds in the lexicon. In many cases, these compounds consist of an adverb or an adjective followed by verb, as in: doorlopen (door-ADVERB + lopen VERB, continue to walk) and opentrekken (open ADJECTIVE + trekken VERB, pull open). There are two important classes of verbs: aspectual verbs and resultative verbs. Aspectual verbs typically are combinations of an adverb with a main verb that refer to the beginning, ending or continuation of the process named by the main verb:

- doorademen: door+ademen (lit. through+breath, continue to breath)
- doorbetalen: door+betalen (lit. through+pay, continue to pay)
- doorlopen: door+lopen (lit. through+walk, continue to walk)
- doorfietsen: door+fietsen (lit. through+walk, continue to walk)
- doorrijden: door+rijden (lit. through+walk, continue to walk)

Within the Dutch wordnet, these verbs are related to two hyperonyms, one that represents the aspectual class and the other being the main verb of the compound verb:

- doorademen[6]:
    - HAS_HYPERONYM ademen (to breath)
    - HAS_HYPERONUM doorgaan (to continue)

Even though these verbs do represent separate synsets, we do not define them as new processes in the ontology. In fact, the will be mapped to the same process as the main verb with that difference that the aspect of the main verb is undefined, while it is restricted for these verbs. The above example, will thus simply be matched with the SUMO process Breathing (subclass of OrganicProcess). From an ontological point of view, we thus do not distinguish between *ademen* en *doorademen*. If necessary it is possible to specify any additional attributes for *doorademen* in a KIF expression without extending the ontology:

doorademen:

(instance ?B Breathing)

(attribute ?B Deep)

(attribute ?B Intens)

Below are examples of resultative verbs that are combined with the adjective “open” (open):

- openknijpen: open+knijpen (lit. open+squeeze, to open by squeezing)
- opendraaien: open+draaien (lit. open+turn, to open by turning)
- openschuiven: open+schuiven (lit. open+slide, top open by sliding)
- dichtknijpen: dicht+knijpen (lit. closed+squeeze, to close by squeezing)
- dichtdraaien: dicht+draaien (lit. closed +turn, to close by turning)
- dichtschuiven: dicht+schuiven (lit. closed +slide, to close by sliding)

Each verb refers to a process that results in the state Open or Close for an object. The verbs all differ in the type of process that causes this, which is expressed by the main verb. These verbs

[6] There is some meaning specialization here: doorademen usually refers to breath deeply and intensely. This aspect is not discussed in this examples.

are now also defined by multiple hyperonyms, one referring to the specific process, and one to a general verb for opening or to the general verb for closing:

- openknijpen: open+knijpen (lit. open+squeeze, to open by squeezing)
  - has_hyperonym knijpen (squeeze)
  - has_hyperonym openmaken (to open)
- opendraaien: open+draaien (lit. open+turn, to open by turning)
  - has_hyperonym draaien (to turn)
  - has_hyperonym openmaken (to open)
- dichtknijpen: dicht+knijpen (lit. closed+squeeze, to close by squeezing)
  - has_hyperonym knijpen (squeeze)
  - has_hyperonym dichtmaken (to close)
- dichtdraaien: dicht+draaien (lit. closed +turn, to close by turning)
  - has_hyperonym draaien (to turn)
  - has_hyperonym dichtmaken (to close)

Also in this case, we are not considering to extend the ontology with new terms. To process itself is best characterized by the main verb, whereas the resultative state is a non-essential property of the process. The conditional end state of the process can be expressed through a complex KIF expression that specifies the change at the end of the process. Being Open and Close can be seen as a change of Shape. Shape alternations are defined in SUMO as follows:

```
(=>
  (and
    (instance ?ALT ShapeChange)
    (patient ?ALT ?OBJ))
  (exists (?PROPERTY)
    (and
      (instance ?PROPERTY ShapeAttribute)
      (or
        (and
          (holdsDuring
            (BeginFn
              (WhenFn ?ALT))
```

```
        (attribute ?OBJ ?PROPERTY))
      (holdsDuring
        (EndFn
          (WhenFn ?ALT))
        (not
          (attribute ?OBJ ?PROPERTY))))
    (and
      (holdsDuring
        (BeginFn
          (WhenFn ?ALT))
        (not
          (attribute ?OBJ ?PROPERTY)))
      (holdsDuring
        (EndFn
          (WhenFn ?ALT))
        (attribute ?OBJ ?PROPERTY)))))))
```

This boils down to the statement that the object is not closed during the squeezing but becomes closed at the end of the process. We now have to express that in the case of *dichtknijpen*, it is the squeezing that causes the ShapeChange. We also need to express that this change closes off an opening.

Such complex statements of changing attributes will apply to many verbs. We therefore propose to use a simpler expression to combine the process of the main verb and the concept ShapeChange in a causal relation: (exists (?SQ ?SC) (and (instance ?SQ Squeezing) (instance ?SC ShapeChange) (cause ?SQ ?SC))).

7.1.4 Parallel Noun and Verb hierarchy

The top of the verb and noun hierarchy is very similar if you look at the major nodes (the ones having many children). Each row in Table 3 contains a noun and verb that are very close in meaning. You can see that the hyponymy relations between these words are also parallel across the parts-of-speech.

Table 3: Top-hierarchy for nouns and verbs

Nominal Hierarchy	Verbal Hierarchy
gebeurtenis (event)	gebeuren (to happen)
handeling (an act)	handelen (to act)
daad (deed)	doen (to do)
verandering (a change)	veranderen (to change, intrans)
beweging (movement)	bewegen (to move, intrans)
verplaatsing (change of location)	verplaatsen (to move position, intrans)

Even though the meanings of these nouns and verbs are very close they still represent different synsets in the wordnet database but not in the ontology. We will thus create equivalent mappings to the same ontology Terms for synsets across parts-of-speech:

gebeurtenis => Process
gebeuren => Process
handeling => IntentionalProcess
handelen => IntentionalProcess
bewegen => BodyMotion
beweging => BodyMotion
etc…

Likewise, the hierarchies of the different part-of-speech are interconnected and the same semantic constraints can be used for each.

7.2 Polysemy

Polysemy refers to the phenomena that a word has different meanings that are somehow related. For example, the word *ambassade* (embassy) can refer both to the institute/office and the building. In some cases, there seems to be a pattern: any word referring to an institute can refer both to the building and the organization. These patterns can either be metaphoric or metonymic in nature. A case of a metaphoric pattern is formed by animal names that can also refer to people with certain properties. A case of a metonymic pattern is the possibility to refer to a whole by means of its parts (pars-pro-toto).

In contrast to other approaches (Briscoe and Copestake 1991, Copestake 1995), these cases of so-called 'regular polysemy' are explicitly listed in Cornetto. The main reason is that these patterns are less regular as they seem and are highly conventionalized and culture-bound. Compare for example *walnut* that can be used to refer to a wide range of objects and products that are related and derived from the walnut tree: fruit, wood, colour, taste, ice-cream, furniture, etc., whereas *chestnut* stands for a similar object but is less productive in our culture and therefore has less conventionalized meanings, even though anybody could productively create them at any time.

Similarly, we do not represent complex types, as defined by Pustejovsky (1995), as a single LU. According to Putsejovsky, a book is a complex type because it entails both the content and the physical object. It is possible to create sentences in which people refer to both aspects of a book at the same time: "He liked reading the book but it was too heavy to carry around". In Cornetto, we see complex types as possible super-entries that group related LUs according to semi-productive patterns. We think it is clearer to list conventionalized form-meaning associations separately but we do require that any possibly semantic relation between LUs is encoded. Likewise, we distinguish two concepts for the content and the object that is the medium for conveying the content (note that a book may contain empty pages!). Nevertheless, the physical object meaning of a book is related to the content meaning in that the written texts represents the content.

In RBN possible relations are indicated by so-called SHIFT labels. There are about 50 different shift patterns between semantic types. These patterns can be used to name the relations between different meanings that are listed explicitly in Cornetto.

8 References

Copestake A. and Briscoe, T (1991). Lexical operations in a unification-based framework. In: Pustejovsky J. and Bergler S. (eds.), Lexical Semantics and Knowledge Representation Association for Computational Linguistics.

Copestake, A. (1995). Representing Lexical Polysemy, in: Proceedings of AAAI Stanford Spring Symposium, Stanford 1995

Fellbaum, C. (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Guarino, N. and C. Welty (2002) Identity and subsumption. In: R. Green, C. Bean and S. Myaeng (eds.), The Semantics of Relationships: an Interdisciplinary Perspective. Kluwer .

Guarino, N. and Welty, C. (2002) Evaluating Ontological Decisions with OntoClean. Communications of the ACM, 45(2), 61-65.

Gruber, T.R. (1993) A translation approach to portable ontologies. Knowledge Acquisition, 199-220.

Horák, A., Pala, P., Rambousek, A., Povolný, M. (2006). DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool. Proceedings of the Third International Wordnet Conference (GWC-06), Jeju Island, Korea.

Maks, I., W. Martin, and H. de Meerseman (1999) RBN Manual, Vrije Universiteit Amsterdam.

Magnini, Bernardo & Gabriela Cavaglià (2000). Integrating subject field codes into WordNet. Proceedings of the Second International Conference Language Resources and Evaluation Conference (LREC), Athens, Greece, 1413–1418.

Niles, I., and Pease, A. (2001) Towards a Standard Upper Ontology. In: Proceedings of FOIS 2001, Ogunquit, Maine, pp. 2-9.

Niles, I. and Pease, A.: Linking Lexicons and Ontologies (2003) Mapping WordNet to the Suggested Upper Merged Ontology. Proceedings of the International Conference on Information and Knowledge Engineering, Las Vegas, Nevada.

Niles, I. and Terry, A. (2004) The MILO: A general-purpose, mid-level ontology. Proceedings of the International Conference on Information and Knowledge Engineering. Las Vegas, Nevada.

Pustejovsky, J. (1995). The Generative Lexicon, MIT Press, Cambridge MA, U.S.A.

Vossen, P (1998, ed.) EuroWordNet: a multilingual database with lexical semantic networks for European Languages. Kluwer, Dordrecht.

Vliet, H.D. van der (2007) The Referentie Bestand Nederlands as a multi-purpose lexical database. International Journal of Lexicography (forthcoming).