Revised Methodology for Knowledge Engineering

Andrei Mikheev and Costanza Navarretta

November 1993

ET-12 Project
Methodologies for Constructing Knowledge Bases
for Natural Language Processing Systems
Report 4
Interim Version

Contents

Preface

In the first two phases of the project *Methodologies for Constructing Knowledge Bases for Natural Language Processing Systems* we analysed the state-of-the-art of knowledge engineering for natural language processing systems (Report 1) and we determined which *kinds* of knowledge should be encoded in natural language understanding systems and which characteristics possible support material should have to shore up the elicitation process (Report 2 and Report 3). In Report 2 we distinguished between the nature of conceptual analysis and knowledge elicitation in the process of building knowledge bases. We proposed a theoretical framework for conceptual analysis and we described the *types* of knowledge which should be included in a conceptual model. Then we defined which information presupposed in the texts is relevant for Natural Language Understanding systems and we adopted a working methodology for extracting this information from a text corpus. In Report 3 we analysed the reusability potential of existing "supporting" linguistic resources and we provided guidelines for collecting prototype text corpora to be used under the knowledge elicitation process.

The objectives of the present phase of the project (Work Package 3:4) are to refine the theoretical framework for conceptual analysis and on knowledge elicitation initiated under Work Package 2, to evaluate existing linguistic resources in the light of their reusability potential for supporting the implementation of the methodology and to test and refine the methodology for some selected domain.

This report (Report 4, interim version) contains a working description of the knowledge elicitation and acquisition methodology which we have defined and on the basis of which we will evaluate the (re)-usability potential of existing linguistic resources (Report 5). The methodology will then be tested in practice (Report 6) and appropriaterly refined (Report 4, final version).

1 Introduction

In the previous phases of the project (Bech *et al.* 1993a) we decided to follow a corpus-supported approach which ensures linguistic anchoredness, extendability and some (re)usability potential of existing resources. Such an approach, for example, was taken in the design of a KB for the TACITUS system (Hobbs 84). However, this approach needs a well specified methodology and a clear description of the different subtasks, the means for their fulfillment, criteria for validation of output results, etc.

In Work Package 2 (Bech *et al.* 1993b) we outlined a framework for knowledge acquisition for natural language understanding systems. This framework is based on the description of three main tasks in text interpretation and types of knowledge structures and interpretation strategies required for each task. We recognized two main activities in knowledge acquisition: knowledge elicitation and conceptual analysis. Elicitation techniques provide bottom-up acquisition of the required knowledge from a text corpus and top-down conceptual analysis is applied for the organization of this knowledge in a consistent way.

The corpus-based strategy for knowledge elicitation which we adopted in the previous phase of the project[1] was the following:

- Make an extensive list of the content words in the text corpus to be processed and an extensive list of general relevant facts about the text corpus and about the content words in it.
- Collect morphologically related words.
- Divide the resulting groups of morphologically related words into subdomains ("clusters").
- Give a first organization of the knowledge in each subdomain.

Next for each content word (or for each group of morphologically related words) do:

- Look for all occurences of the word in the text corpus to see the contexts in which the word is used. When necessary, look at previous or following sentences to resolve anaphora.

[1] It is an extension of Jerry Hobbs' three–step strategy (Hobbs 1984).

- Reduce the citations to their predicate-argument relations. Examine the contexts and determine what facts about the word are required to justify each of the occurrences of the word.

- Make a preliminary division of these predicate-argument relations into heaps, according to a first analysis of which predicates should go together. This first analysis is based i.a. on the knowledge enterer's linguistic knowledge and on his knowledge about the text corpus. If more facts are presupposed in a citation, patterns must be split up.

- Give an abstract characterization of the facts about the word that justify each of the heaps (making explicit the first analysis that underlay the classification into categories in the previous step). Recognizing a more abstract characterization may lead one to join two heaps, and failure to find a single abstract characterization may lead one to split a heap.

Although some of the above steps can be supported by existing tools (e.g. concordance and corpus tools, tools for clustering morphologically related words), the methodology still presents some problems. It is too general, it leaves many decisions to the intuition and judgement of the single knowledge engineer and, if applied to large text corpora, it requires huge resources. To reduce the cost of constructing knowledge bases the elicitation method must be specialized with strategies, techniques and tools which facilitate the extraction of different types of knowledge. To improve the reliability of the resulting knowledge bases the method must be also supplied with methods and techniques for analysing large text corpora. Central to both issues is the reusability of existing linguistic resources such as machine-readable dictionaries and lexical knowledge bases[2].

Our ultimate goal is, of course, to determine which of these specialized strategies/techniques can be automated so that the cost connected with the construction of knowledge bases can be reduced and the process of selecting the relevant facts from the texts can be made less introspective and thus more reliable.

In Navarretta (1993) we determined that large general-language and technical text corpora can support the knowledge elicitation process in different ways, i.a. they can be used to assist the process of characterizing words

[2] We will evaluate some of these reusable resources in Report 5.

which only occur few times in the source text corpus, to guide the step of grouping occurrences of a word in heaps by identifying the distributional behaviour of the word in a larger corpus, to provide a test-bed for the processing system and to test the lexical coverage achieved by the system under the development phases, to evaluate the suitability of the higher levels of the ontological model (in TACITUS' terms the "core theories"), to analyse linguistic phenomena that deviate from the "normal" use of the language (syntactically ill-formed utterances).

Working with large general-language and/or technical corpora requires statistical strategies and methods[3] which permit to recognize recurring relations and patterns in the huge material at hand. We have investigated whether and how some of this strategies can be (re)used in our methodology.

The main goals of this report are to combine knowledge elicitation and conceptual analysis into a single methodology, to elaborate the description of cognitive principles for knowledge organization and to refine the adopted elicitation strategies.

The main feature of the proposed methodology is that an analysis of lexical semantics is performed not on a word by word basis but rather on the basis of cognitive schemata. These cognitive schemata are abstract conceptual structures which represent templates in which information from texts should be arranged. The semantics of words can then be described in terms of these conceptual schemata.

The methodology starts with the acquisition of conceptual vocabulary by preprocessing large amounts of texts. The next step is vocabulary structuring which can be seen as a preliminary activity for vocabulary conceptualization. During conceptualization lexical items are organized into conceptual types. After having a well established system of conceptual types the next activity is conceptual characterization of them. This step can be seen as a multilevel conceptualization of the domain: the corpus is seen as a collection of abstract conceptual schemata each of which covers a set of fragments of the corpus. Each of these fragments are analyzed for subfragments up to the level of lexical entries. The methodology ends up with characterization of not only lexical entries but also cognitive schemata of the domain (corpus).

[3] The availability of large text corpora has been seen as one of the main reasons for what is called a revival of the 1950s empiricism in computational linguistics (Church and Mercer 1993).

2 Domain Vocabulary Acquisition

2.1 The Task

The first task in knowledge engineering for a new corpus is to perform basic vocabulary acquisition. This basic vocabulary is an extensive list of lexical entries which can be found in the corpus. The main characteristic of this vocabulary is its orientation on the sublanguage of the domain (or corpus). Unlike general morphological or other word lists the domain vocabulary consists of lexical entries rather than words. A lexical entry can be a word ("admit"), a phrase ({cardiac catheterisation}) or a pattern({date˜{{day˜}{month˜}{year˜}}}).

Though the preprocessing of the corpus causes extra work, this will considerably contribute to the interpretation process. Many domain specific terms are multi-word phrases which can be more easily handled as a single entry rather than composed each time from individual words.

Each lexical entry should be supplied with its part of speech and frequency of appearance in the corpus. Morphologically related entries should be grouped together since they will share the same conceptualization. For example, lexical entries "to admit - TV" and "admission - N" correspond to the same conceptualization, the entry "admitted - Adj" is derived from this conceptualization.

For a corpus of about 500 PDSs (described in Bech *et al.* 1993b), which contains roughly 90,000 running words, a vocabulary of about 2,900 lexical entries was created. The corpus was preprocessed and tagged using a number of tools. Amongst the 2,900 lexical entries there are roughly 700 verbs, 950 adjectives, 150 function words and approximately 1,100 nouns. Here is an excerpt from the lexical entry list:

admit: transitive verb, 94; *admitted*: adj, 56; *admission*: noun, 16.

{*cardiac catheterisation*}: noun, 64.

{*left ventricular function*}: noun, 27.

{*date˜*{{*day˜*}{*month˜*}{*year˜*}}}: noun, 500.

{*amount˜*{{*quantity˜*}{*unit˜*}{*measure˜*}}}: noun, 420.

The next step in building the domain vocabulary is to separate contents and functional entries. Contents lexical items correspond to nouns, verbs, adjectives and adverbs. Functional words are pronouns, determiners, prepositions etc. Contents entries will be mapped into concepts while functional ones are mapped into relations. In fact functional entries constitute a rather small dictionary which can be completely reusable moving from one corpus to another.

2.2 Elicitation Strategies

2.2.1 Identification of Technical Words

Huizhong (1986) describes a method for automatically identifying scientific/technical terms in text corpora. The method is based on the fact that one of the specific features of English for Science and Technology is the high concentration of terms and it has been applied to nine scientific texts of about 30,000 running words each and to a non-scientific text of the same size (Graham Greene's *The Human Factor*).

Single-word terms are recognized according to their frequencies of occurrence and distribution while multi-word terms are identified by their collocational behaviour. The applied strategy is the following: extract the frequency of each word in each of the ten texts. For each word record the density (in how many texts the word appeared), **D**, and the average frequency, **F**. To provide a measure of frequency dispersion use the relative standard deviation (the standard deviation divided by the average frequency), **SD**. Finally calculate the peakratio (the maximum frequency of occurrence divided by the average frequency), **P**, and the rangeratio, (the maximum frequency divided by the minimum frequency), **R**.

With the criterion of distribution sat empirically to an appropriate value (in this case ≥ 7) it has been possible to identify function words (these words have very high distribution and fairly high average frequencies of occurrence). Words with very high **D** but relatively low **F** are not function words, sometimes called sub-technical words. Both function words and sub-technical words have fairly low **SD**, **P** and **R**. Scientific/technical terms show very low **D**, but very high **P** and **R**. With an empirically sat **P** and **R** (in the actual case respectively 5 and 10) single-word terms have been recognized.

Multi-word terms have been identified on the basis of their collocational behaviour and of presuppositions such as 'multi-word terms are mainly nominals', 'multi-word terms cannot go across punctuation marks', 'function words should be excluded with the exception of prepositions, because prepositions may be part of multi-word terms', 'adverbs may be part of a multi-word term, but adverbs for text cohesion (e.g. subsequently, naturally, usually) should be excluded', 'no multi-word terms can end up with an adjective or adverb' ([10] p. 100)

Multi-word combinations have been produced in an iterative way: first 2-word combinations, then 3-word combinations and so on. To distinguish between free combinations and terms the frequency counting has been considered. When the frequency of occurrence is high enough all or most of the combinations are scientific/technical terms. The same method applied to the non-scientific text has shown that here almost no multi-word is a term.

The automatic extraction of technical terms and multi-words can be particularly useful when constructing knowledge bases for a specific domain because it permits a primary division between general-knowledge words and domain-specific words (this is also relevant for structuring the vocabulary (see section refrough1). The recognition of terminological multi-words is also important because terminological multi-words have always a fixed meaning and must therefore not be disambiguated by the system.

To be used in the knowledge elicitation process the method should be applied on the source text corpus and on other corpora (here included a general-language subcorpus) of the same dimension and the presuppositions used to identify multi-words should be properly redefined.

3 Rough Structuring of the Vocabulary

3.1 The Task

The created domain vocabulary can be fairly big (several thousands of lexical entries) and to work with such a big list is quite a difficult task. However, one can feel that certain lexical entries can be grouped together constituting semantical clusters or microdomains.

The first rough structuring can split up the vocabulary into three parts: commonsense entries - they are domain independent and are used in many different domains; terms - highly domain specific lexical items; other entries - these can be seen as an intermediate layer between the commonsense and domain specific entries: they are general enough but on the other hand they correspond to a certain professional domain.

Content lexical entries in the PDS corpus have been divided into the following groups (excerpt):

- **Domain independent lexicon**: address, available, basis, continue, current, dear, decide (decision), {date of birth} (DOB), {date˜{{day˜}{month˜}{year˜}}}, enclose, full, good, largely, less, letter, mr, much, {patient name}, plan, report, review, secondary, sex, similar, ...
- **Domain specific lexicon**: angina, {coronary atherosclerosis}, {cardiac catheterisation}, prognosis, {thallium study}, {left ventricular function}, {double vessel disease}...
- **Intermediate layer**: admit, discharge, {main diagnosis}, disease, drug, dr., hospital, patient, {medical treatment}, {surgical treatment}, therapy, surgery (surgical), symptom, consultant...

Each of these three groups is to be split further into subdomains (microdomains) with hierarchical inclusions and interdependencies. In every subdomain some lexical entries can be grouped together if they can be defined in terms of each other (scales, metrics etc).

For example, the domain independent lexical items can be split into the following commonsense domains:

- **Evaluative scales**:
  good – bad; less – much; similar; largely;
- **Time**:
  {date˜{{day˜}{month˜}{year˜}}}, date, current
- **Person**:
  sex, {date of birth} (DOB), mr., {patient name}

- **Activities**:
  plan, review, report, decision (decide)

3.2 Elicitation Strategies

3.2.1 Clustering

A large group of the statistical methods applied to large text corpora in the field of computational linguistics is that of clustering methods. The aim of these methods is that of automatically classifying words according to their contexts of use, usually for supporting the disambiguation of alternative analyses proposed by a parser. The most interesting methods for our purpose are those which can identify classes of semantically related words, giving a useful guideline to the step of clustering relevant facts for each word in our methodology.

Pereira *et al.* (1993) are investigating "how to factor word association tendencies into associations of words to certain hidden 'sense classes' and associations between the classes themselves" (p. 183). In their first experiments they have addressed the specific problem of classifying nouns, using the relation between a transitive main verb and the head noun of its direct object. A newswire text corpus has been used[4].

[4] Pereira and his colleagues have first extracted verbs and nouns using a parser, later they have used a statistical part-of-speech tagger. They have not yet compared which of the two strategies gives the best results.

Their classification method constructs a set C of clusters and cluster membership probabilities $p(c|n)$. Each cluster c is associated to a cluster centroid p_c, which is a distribution over the class of verbs obtained by averaging appropriately the p_n[5].

The application of this clustering method so far indicates that it is possible to group nouns according to their participation in a transitive relation with verbs using a general divisive clustering procedure for probability distribution.

The resulting clusters of word classes can be primarly used to disambiguate alternative analysis proposed by a grammar, but the defined clustering method can also resolve the problem of data sparseness[6] because the likelihood of unseen events is estimated from that of "similar" events which have been registred (in this case the likelihood of a direct object for a verb is estimated from the likelihood of that direct object for "similar" verbs, where "similarity" is understood as the similarity of the contexts in which words occur). Of particular interest for us is the fact that the clusters of word classes extracted with distributional clustering techniques seem to be semantic significant[7]. If this is the case the step of clustering semantically related facts in our knowledge elicitation methodology could, in part, be automatized and, when dealing with words that only appear once or twice in the source text corpus, it could be made more reliable.

Pereira and his colleagues believe that the method is general and can be applied to other grammatical relations and to other word classes, but the generality of the method must be proved in practice.

Hatzivassiloglou and McKeown (1993) are investigating how to cluster adjectives according to their meaning. With the long-term goal of defining a

[5] To determine the centroid distributions $p_c(v)$ they use the maximum likekihood estimation principle. Because the only considered information is the measure of object-to-cluster similarity, the membership of a word to a cluster is determined by maximating the configuration entropy for a fixed average distortion. A complete description of the method can be found in [22].

[6] For large corpora the number of possible joint events is much larger than the number of event occurrences in the corpus, so that many events are seen rarely or never. For this reason simple cooccurrences of groups of words are not always reliable.

[7] In the experiments made by Pereira and his colleagues the following two clusters of direct objects for the verb *fire* have been identified: a) missile, rocket, bullet, gun, b) officer, aide, chief, manager, corresponding to the two senses of *fire* 'shoot' and 'sack' respectively.

method for automatically identifying adjectival scales[8] Hatzivassiloglou and McKeown have found a technique for clustering adjectives in a given text corpus so that the elements in each cluster describe different values of the same property. The basic hypothesis behind their approach is that adjectives describing the same property tend to modify approximately the same set of nouns.

The applied clustering method combines statistical techniques and linguistic information, it is domain-independent and presupposes the use of a parser on the source text corpus. It consists of the following steps:

1. Linguistic data from the parsed corpus are extracted in the form of syntactically related word pairs or sequences of syntactically related words.

2. The various types of co-occurrence relations which have been identified in the text are forwarded to a set of independent *similarity modules*, which operate in parallel. These modules weight, when possible, similarity/dissimilarity between any two adjectives according to some linguistic criterion.

3. The different judgements are combined, resulting in a dissimilarity measure for any pair of adjectives.

4. The adjectives are clustered according to the dissimilarity measure (a cluster contains adjectives that are similar) with a non-hierarchical clustering algorithm.

Until now only two similarity modules have been implemented based on two types of linguistic data: a) data that help to establish that two adjectives are related (adjectives are similar if they have similar distributions[9]); b) data that help to establish that two adjectives are unrelated (a strong indication

[8] Hatzivassiloglou and McKeown follow Levison (1983) in defining linguistic scales as sets of words, of the same grammatical category, which can be ordered by their semantic strength or degree of informativeness. The elements of linguistic scales must be totally ordered, but in the case of adjectives this restriction is usually relaxed so that the elements of an adjectival scale can belong to two totally ordered sub-scales (containing positive and negative degrees of the common property).

[9] For each possible pair in the adjective-noun frequency table the two distributions of nouns are compared using Kendall's τ coefficient so that the two random variables are the two actual adjectives and each paired observation is their frequence of cooccurrence with each given noun.

that two adjectives do not belong to the same group is the fact that they occur as pre-modifiers within the same NP[10]).

The system has been satisfactorily tested on a 8.2 million word corpus of stock market reports from the Associated Press' newswire. A subset of 21 adjectives, all modifying the noun *problem*, has been selected. The initial results have been promising and Hatzivassiloglou and McKeown in the future will improve the system i.a. using the fact that adjectives with a higher degree of semantic contents seem to form more stable associations than relatively semantically empty adjectives.

The above described adjective clustering method can support the step of clustering related facts in our own methodology. Techniques to automatically extract adjectival scales will be even more useful in that scales can be found in nearly every domain.

Other word classification algorithms are based on class-based n-gram models (Brown et.al. 1992, Lafferty and Mercer 1993).

Given a vocabulary of V words which are partitioned into C classes using a function π, mapping a word w_i into its class c_i, a language model is an *n-gram class model* if it is an n-gram language model and if for $1 \leq k \leq n$, $Pr(w_k \mid w_1^{k-1}) = Pr(w_k \mid c_k)Pr(c_k \mid c_1^{k-1})$. An n-gram class model has $C^n - 1 + V - C$ independent parameters.

A bigram class model predicts a word w_2 given a previous word w_1, by first predicting the class of w_2 given the class of w_1, and then prediciting the actual word in the class. The probability of a training corpus $C = w_1,\ldots,w_N$ is then estimated as
$Pr(C) = Pr(c(w_1))Pr(w_1 \mid c(w_1)) \prod_{i=2}^{N} Pr(c(w_i) \mid c(w_{i-1}))Pr(w_i \mid c(w_i))$.

[10] If both adjectives modify the head noun and if they are antithetical, the NP would be self- contradictory (e.g. hot cold, red black); if the two adjectives are non-antithetical scalar adjectives and both modify the head noun, the NP would violate the Gricean maxim of Manner, since the same information is conveyed by the strongest of the two adjectives (hot warm); if one adjective modifies the other, the modifying adjective has to modify the other in a different dimension (light blue shirt - blue is the color, light indicates the shade). The latter point is not always true, but the authors rely on the fact that combinations of adjectives as blue-green are often hyphenated.

If the training set is large the logarithm of the likelihood[11] for this kind of bigram model can be broken into two terms, one involving the sum over all the words in the vocabulary, the other involving the sum over all class bigrams.

$$\begin{aligned} L(\pi) &\approx \textstyle\sum_w Pr(w) \log Pr(w) + \sum_{c_1 c_2} Pr(c_1 c_2) \log \frac{Pr(c_2|c_1)}{Pr(c_2)} \\ &= -H(W) + I(C_1, C_2), \end{aligned}$$

where $-H(W)$ is the entropy[12] of the unigram word distribution and $I(C_1, C_2)$ is the average mutual information of adjacent classes.

The problem of finding optimal classes, i.e. classes where the average mutual information is maximized, is computationally hard. In Brown *et al.* (1992) is given the following two-phases' algorithm to compute these classes: 1) Assign each word to its own class and compute for adjacent classes how much mutual information would be lost if they were merged. Merge the classes for which the loss in average mutual information is least, then iterate (e.g a lot of information would be lost if *is* and *the* were merged, while the average mutual information between *Monday* and *Tuesday* is quite low from the point of view of likelihood). 2) After having derived a set of classes from successive merges visit each class and move each word to another class if so the likelihood of the training corpus can be improved.

This algorithm can be applied for dividing words into classes which capture both syntactic and semantic aspects of English[13]. The order in which clusters are merged is also of interest because it determines a binary tree the root of which corresponds to the cluster containing the complete vocabulary and the leaves of which correspond to the words of the vocabulary. The internal nodes of the tree correspond to groupings of words intermediate between single words and the entire vocabulary. Words that are statistically similar with respect to their immediate neighbors in running text will be close together in the tree.

Brown and his colleagues have also applied the clustering algorithm for identifying pairs of adjacent words that are present in the text next to one another more often than one could expect and pairs of words that occur

[11] The likelihood with which the model generates the training text.

[12] Entropy is a measure of the information content of a probabilistic source.

[13] In [4] are reported classes obtained from a vocabulary of 260,741 words. Examples of these classes are the following: head, body, hands, eyes, voice, arm, seat, eye, hair, mouth; anyone, someone, anybody, somebody; had, hadn't, hath, could've, should've...

near one another more often than one could expect[14]. The latter groups of words are both morfologically and semantically related (e.g. write, writes, writting, written, wrote, pen; sell, buy, selling, buying, sold).

This method has the shortcoming of forcing each word into only one single class disregarding the ambiguity of natural languages. This happens because the method averages the mutual information over all contexts in which a word appears.

Lafferty and Mercer (1993) are working to ameliorate this problem by using morphological information about each word. Their version of the clustering method takes the characters which constitute the words as basic elements.

3.2.2 Disambiguation of Word Senses

Word sense disambiguation is one of the problems one has to cope with when analysing natural language texts. Justeson and Katz (1993) describe a method for disambiguating adjective senses by the nouns or the noun phrases they modify, using co-occurrences in large text corpora. They have used statistical inference methods as tools for organizing and analysing the large amount of material collected from large text corpora and they have addressed the problem of finding clues within the context of a word that indicate its sense fairly reliably. In particular they use the mutual relevance of adjectives and noun senses which is content-specific (semantic) rather than word specific (lexical)[15].

The disambiguation method is based on the observation that some nouns are strongly associated with particular senses of some of the adjectives that modify them (e.g. *old* in relation with the noun *man* has the sense of "aged", while the same adjective in relation with the noun *house* has the sense of "used"). These nouns are called **indicators** for the sense of an adjective.

[14] The former pairs of words are called *sticky* where *stickness* is an asymmetric phenomenon. The latter pairs of words are related in a so-called *semantic stickness*-relation which is symmetric.

[15] Word-specific relations between adjectives and nouns are idiomatic, noncompositional pairs or 'freezes' in which the adjective itself has no independent sense, e.g. *hard cash* and *short cut*.

Adjectives are disambiguated according to their co-occurrence with sense-specific antonyms because an adjective and its antonym refer to opposed values of the same attribute. Antonyms co-occur in direct comparison or in contrastive opposition.

Justeson and Katz have tested the method empirically for disambiguating five common adjectives: *hard, light, old, right, short* using the following ten antonymous pairs: *hard-easy, hard-soft; light-dark, light-heavy; old-new, old-young; right-left, right-wrong; short-long; short-tall.* They have extracted all the sentences containing co-occurrences of one target adjective and each of its antonyms from the APHB corpus[16]. The sentences in which co-occurring antonymous adjectives modify the same noun constitute a subcorpus in which the ambiguous members of the antonym pairs are discriminated relative to their antonym-specific senses. In the collected subcorpora only nouns that are specific to particular senses of the target adjective are used. The sense specificity of a noun has been demonstrated by looking at all the occurrences of the adjective-antonym pairs with the given noun. To prove that the sense specificity was not connected with the contrastive structures of the considered sentences Justeson and Katz have extracted from the entire APHB corpus a random sample of 100 sentences containing adjectival instances of each target adjective (excluding all freezes and quantificational expressions), then they have disambiguated them manually. Being only interested in discriminating between the two antonym-related sets of senses of the targets, they have only considered the instances of a target adjective occurring in a sense for which an antonym exists. They have then projected from the disambiguated subcorpora to the whole corpus the probability of each sense of the target adjective given the noun it modifies.

Because the mutual relevance of nouns and adjectives is concept- and not word-specific, they can generalize from the noun indicators obtaining a small number of conceptual categories (e.g. the sense ”aged” of ”old” can be extracted using the feature ”human”). Sometimes generalization does not lead to semantic but to syntactic cues for sense identification such as *predicative indicator features*, e.g. ’That’s not quite right’, and *infinitival indicator features*, e.g. ’It is hard to be a child’).

The results of Justeson and Katz’ experiments show that some adjectives can be disambiguated by the nouns they modify, largely on the basis of

[16] Sentences in which the adjectives are not used adjectivally have been eliminated manually.

general semantic attributes characterizing those nouns[17] and that a small number of close syntactic relations channel much of the semantic interpretation involved in disambiguation, at least in the case of sense dichotomies in adjectives.

Whether the method can be extended to other adjectives and to more syntactic relations of adjectives (e.g. verb-based disambiguation of adjectives) has to be demonstrated. The method is very interesting because it has been applied to common adjectives which can easily appear with different senses in the same domain, thus not permitting an a-priori disambiguation on the basis of it.

[17] The attributes are semantic and most then relate to noun facts and not to nouns per se (e.g. there is a pertinent ambiguity in nouns as *nun, friend, empress* that should be considerate +animate when designing a person and as -animate when designing relationships or the role itself).

4 Domain Vocabulary Conceptualization

4.1 The Task

If tasks 1 and 2 can be seen as elicitation of lexical material at this stage conceptual analysis and initial organization of this material are to be performed. The ultimate goal at this stage is to transform heaps of lexical entries into ordered structures of conceptual types. The main difference between a lexical entry and a concept is that concepts have a single fixed meaning. A lexical entry can correspond to several concepts (polysemy) and a concept can be mapped onto several lexical entries (synonymy).

For example, there can be two conceptual entries Admit-1 and Admit-2 if the word "admit" is used in its first two senses in the corpus:

> *Longman's Dictionary of Contemporary English:*
> - to permit (a person or thing) to enter; let in
> - to state or agree to the truth of (usually) something bad; to confess
> - to have space or room for
> *Concise Oxford Dictionary:*
> - accept as valid or true
> - allow (a person) entrance or access
> - (of a hospital etc.) bring in (a person) for residential treatment

If in the corpus two or more lexical entries are used as synonyms (i.e. they correspond to the same concept) the conceptualization of one of them should be taken as the main one and all the rest of lexical entries will correspond to this concept. For example, if in our medical domain words "discharge" and "release" both correspond to the meaning :*to let a patient out of a hospital*, they should correspond to the same concept, say, Discharge.

For the domain dependent part of the vocabulary (terminology) correspondence between lexical entries and concepts is usually straightforward, thus every lexical entry from this part is granted the concept status. Lexical entries from the other parts of the vocabulary may not have that straightforward mapping. At this stage they can be mapped into conceptual entries like the domain dependent lexica, but at further stages of KB design this should be refined.

The other task at this stage is to provide structuring among concepts. This structuring need not to be total: instead one can create many Conceptual Type Lattices (CTLs) even in one microdomain. Hierarchical inclusion of subdomains also contributes to the structuring of conceptual types – types from a microdomain can be subsumed by a single type from its parent domain.

The type hierarchy is the basis for the organization of the knowledge base. Instead of operating with individual entities of a different nature the hierarchy provides a structuring of these entities into conceptual types according to two principles: abstraction and generalization.

Each conceptual type is characterized by a set of properties, it obeys a particular set of constraints and has certain links with other types. Types are organized into the type system by means of hierarchical inclusion with property inheritance and aggregation of types into conceptual clusters.

The type hierarchy or the *is-a* taxonomy can be found in different knowledge sources: encyclopedias, thesauri, domain specific term and knowledge banks etc. The main property which makes this knowledge structuring so popular is transitivity of the *is-a* relation. Conceptual types can be organized into a conceptual type lattice (CTL) with multiple inheritance and type subsumption.

There are no universally accepted methods for building a CTL. Fairly widely accepted assumptions are:

- that a CTL captures essentially context independent knowledge;
- that a CTL is created for concepts rather than words: each item of the CTL has a fixed single meaning;
- that a CTL includes only the *is-a* taxonomy; aggregative relations are not included in the CTL since in general they are non-transitive. The aggregative taxonomy is built on top of the CTL.

With respect to the first point, it is worth underlining that, although the CTL is context independent, it is partly domain or task dependent. Different domains and tasks can have different views on the same types and their hierarchical inclusion. So the CTL is context independent in the framework of a single domain.

This excerpt from the CTL of the medical domain lists medical procedures. The event type is a general ontological type.

```
event →  procedure.
           →            exercise-test.
           →  administrative-procedure.
                                      →         admission.
                                      →         discharge.
                                      →   hospitalisation.
                                      →      consultation.
                                      →     medical-order.
           →           general-action.
                                      →          puncture.
                                      →          punction.
                                      →         injection.
                                      →     opacification.
                                      →       exploration.
                                      →              swab.
           →              examination.
           →                treatment.
                                      →  non-invasive-treatment.
                                                              →  cardiac-catheterisation.
                                      →      invasive-treatment.
                                                              →                  surgery.
```

There are two main streams in building domain ontologies. The first approach is aimed at the creation of a single CTL which includes all concepts of the domain. When building such a CTL one can distinguish the following parts of it:

- domain specific types - these types are represented in the NL by terminology. Terms usually have a single meaning and it is fairly easy to acquire and structure them.

- general domain types - these types represent general concepts of the domain. Words which represent the domain in this part can have several meanings and be used in other domains differently. However, in the framework of a particular domain these words refer to concepts one to one.

- general purpose types - these types correspond to concepts which are generally used as commonsense knowledge. These types are quite task dependent because of their very broad meaning.

- top level categories - these types determine an ontological foundation of the world. This foundation is domain independent and determines main types with which the knowledge engineer can build domain dependent ontology incrementally.

The other approach to domain conceptualization is aimed at the creation of many self-consistent ontologies. These subdomains' ontologies can be organized into conceptual hierarchy and corresponding concepts can be defined inside a subdomain. Division into subdomains as in case with the single CTL includes commonsense subdomains which correspond to general purpose types and domain specific subdomains with different detailization - these correspond to general domain types and domain specific types. Implicitly all these subdomains are based on the same ontological foundation which corresponds to top level categories.

4.2 Methodology for the Type System Organization

Organization of conceptual types into a CTL is by no means an easy process. A well defined structuring can be provided usually only for domain specific types.

Main principles for domain structuring were defined in Bech *et al.* 1993b. However, we don't think that all conceptual types should obligatorily be provided with their supertypes. If for some concepts there is no obvious and useful structuring they can be left by their own. Moreover, we think that there is no need to force all concepts to be represented in a single CTL, it is more useful to create several CTLs for microdomains and combine some of them if it seems natural.

One can feel however, that top level categories can be reused in many domains. These categories have different ontological existence and require different conceptualization. Importance of the right choice of the main categories is self evident.

In appendix A the Edinburgh team suggests a possible methodology for analysis and organization of conceptual types.

5 Conceptual Characterization

5.1 The general Task

The main aim at this stage is to characterize concepts in terms of their coexistence with other concepts. We need to analyze occurences of a concept in the corpus and generate a canonical structure for it: a structure in which the concept of interest is supplied with valent slots which can be filled in the sentence with certain conceptual types and certain morpho-syntactic features.

The following canonical structure for the concept CARD-CATH which corresponds to the lexical entry {cardiac catheterisation} expects certain conceptual types with certain syntactic attributes to be linked with it:

cardiac catheterisation

[CARD-CAT]-
→(agnt)→[DOCTOR: synt "subject"]
→(ptnt)→[PATIENT: synt "direct object"]
→(loc)→[HOSPITAL: synt "in"]
→(cul)→[TIME-POINT: synt "on"]

Conceptual characterization also can refine the domain vocabulary and the CTL. At the first stage it is possible to apply special elicitation techniques for automatic channeling required information. However, one can not avoid quite labour-intensive manual characterization.

5.2 Automatic Elicitation Strategies

5.2.1 Canonical Structures

Related to the concept of canonical structures are those of *case grammar*, *valency theory*, *thematic roles theory*. Some recently defined statistical strategies can in part facilitate the extraction of canonical structures. In literature on these statistical strategies, names as *valency*, *case semantics*, *thematic roles* etc. are used without a particular commitment to one or the other theory, but referring to what is "usually" understood by these names.

In the following we will just use the names used by the authors that describe the different strategies. The goal of these strategies vary, some are defined to extract the syntactic arguments for verbs (or adjectives), others can, partially, elicitate the deep case semantics of verbs (and/or adjectives).

Manning (1993) reports a method for producing a dictionary of subcategorization frames from unlabelled text corpora. A subcategorization frame is a statement of what types of syntactic arguments a verb or adjective takes. Manning wants to produce all possible subcategorization frames, thus his strategy is to extract as much (noicy) information as possible and then use statistical techniques to filter the results. In this way he can also cope with the sparseness of some cues. He distinguishes 19 classes of subcategorization frames (e.g intransitive, transitive and ditransitive verbs, verbs which take a finite *that* complement, verbs with direct object and *that* complement).

The method consists of two steps: parsing and filtering. In the first step a finite state parser runs through the texts and parses auxiliary sequences. Complements after verbs are recorded and histogram-type statistics for the appearence of verbs in various contexts are collected. The parser does not distinguish between arguments and adjuncts[18], except for the fact that only the first of multiple PPs is counted as an argument.

In the second step of the method a filtering mechanism checks the frames found by the parser. A cue may be a correct subcategorization for a verb, it may contain spurious adjuncts, or it may be wrong due to a mistake made by the parser. Manning uses a method for filtering suggested by Brent (1992). Let $\mathcal{B}_s$ be an estimated upper bound on the probability that a token of a verb which does not take the subcategorization frame s will nevertherless appear with a cue for s. If a verb appears m times in the corpus, and n of those times it occurs with a cue for s, then the probability that all cues are false clues is bounded by the binomial distribution:

$$\textstyle\sum_{i=n}^{m} \frac{m!}{n!(m-n)!}\mathcal{B}_s^n(1-\mathcal{B}_s)^{m-n}.$$

The null hypothesis that the verb does not have the subcategorization frame s can be rejected if the above sum is less than some confidence level C.

[18] Arguments fill semantic slots licenced by a particular verb/adjective, while adjuncts provide information about sentential slots (e.g. time, place) that can be filled for any verb of the appropriate aspectual type.

The results of the method have been compared to the subcategorizations in a dictionary, and a first evaluation has showed that the method works at least as well as previously tried techniques, with the advantage that it can learn all the possible subcategorization frames of verbs and not only a restricted number of them as it is the case in similar strategies.

Liu and Soo (1993) have implemented a system for acquiring domain-independent thematic knowledge using available syntactic resources. They have considered the following thematic roles (argument structure): agent, goal, source, instrument, theme, beneficiary, location, time, quantity, proposition, manner, cause, result. They have applied four clues: 1. the possible syntactic constituents of the arguments, 2. whether animate or inanimate arguments, 3. grammatical functions (subject or object) of the arguments when they are Noun Phrases (NPs), 4. prepositions of the prepositional phrase in which the arguments may occur. The syntactic constituents they have considered are the following: NP, proposition, adverbial phrase, adjective phrase and prepositional phrase.

When a training sentence is entered, arguments of lexical verbs in the sentence are extracted invoking a syntactic processor (if the sentence is selected from a syntactically processed corpus the arguments may be directly extracted from the corpus). The ambiguities that remain after this syntactic processing are resolved by evidences from trainers and large text corpora. To discriminate thematic roles the authors use the following kinds of heuristics:

- **Volition Heuristics**: Purposive constructions (e.g. in order to) and purposive adverbials may occur in sentences with agent arguments (Gruber 76).

- **Imperative Heuristics**: Imperatives are permissible only for agent subjects (Gruber 76).

- **Thematic Hierarchy Heuristics**: Given a thematic hierarchy from higher to lower: Agent>Location, Source, Goal>Theme, the passive by-phrases must reside at a higher level than the derived subjects in the hierarchy, i.e. the Thematic Hierarchy Condition in Jackendoff (1972). Liu and Soo set up the following hierarchy: Agent>Location, Source, Goal; Instrument, Cause>Theme, Beneficiary, Time, Quantity, Proposition, Manner, Result. Subjects and objects cannot reside at the same level.

- **Preposition Heuristics**: The prepositions of the PPs in which the arguments occur often convey good discrimination information for resolving thematic roles ambiguities.
- **One-Theme Heuristics**: An argument is preferred to be Theme if it is the only possible Theme in the argument structure.
- **Uniqueness Heuristics**: No two arguments may receive the same thematic role (exclusive of conjunctions and anaphora with co-relate two constituents assigned with the same thematic role).

Volitions and Imperative Heuristics are for confirming the Agent role, One-theme Heuristics is for Theme, while the remaining heuristics may be used in general. The number of queries may be minimized by applying the heuristics in the order: Volition Heuristics and Imperative Heuristics -> Thematic Hierarchy Heuristics -> Preposition Heuristics. One-Theme Heuristics and Uniqueness Heuristics are invoked each time current hypotheses of thematic roles are changed by the application of the clues.

The average accuracy rate of the acquired argument structures in the first experiments has been 0.86[19]. Failed cases have been mainly due to the clues and heuristics that were too strong or overly committed.

In Velardi et al. (1991) is described an experimental NL processor, DANTE, for learning syncategorematic concepts from texts. The applied knowledge acquisition method is based on learning by observations (based on words' patterns of use) in a large Italian text corpus. The aim of the system is to extract collocative meaning.

Given a sequential presentation of word associations, a many-to-many mapping from words to concept types and a hierarchical ordering of concept types, the method must find for each pair of associated words the concept types and the conceptual relation that interpret that pair (CRC) and a definition for each concept that summarizes its instances (derived CRCs). Thus the approach described in Velardi *et al.* (1991) is different from that of concept formation (Gennari *et al.* 1989) in that instances (words in contexts) are not associated with descriptions (CRC triples) and clustering of instances in categories and the hierarchical organization for these categories are given.

[19] An argument structure was considered correct if, it was unambiguous and confirmed by a trainer who checked the thematic validities of the sentences generated by the learner.

DANTE takes as input the following:

- A list of syntactic collocates (e.g. subject-verb, verb-object, noun-preposition-noun, noun-adjective etc.) extracted through a morphologic and syntactic analysis of the selected corpus. Only sentence parts are parsed and some context-dependent heuristics are used to cut sentences into clauses.

- A semantic bias consisting of a) a domain-dependent concept hierarchy (a many-to-many mapping from words to word sense names and an ordered list of conceptual categories)[20]; b) a set of domain-dependent conceptual relations, and a many-to-many mapping (synt-sem) between syntactic relations and the corresponding conceptual relations; c) a set of coarse-grained selectional restrictions on the use of conceptual relations, represented by concept-relation-concept (CRC) triples (expressed in Conceptual Graph notation, (Sowa 1984)).

The output to the system is a set of fine-grained CRCs, that are clustered around concepts and around conceptual relations and an average-grained semantic knowledge base, organized in CRC triples.

To acquire syncategorematic knowledge on concepts, the applied algorithm proceeds as follows:

For any syntactic collocate sc(w1,w2)

1. Restrict the set of conceptual relations that could correspond to the syntactic collocate using the synt-sem table.

2. Use coarse-grained knowledge and taxonomic knowledge to further restrict the hypotheses.

3. If no interpretation is found, reject the collocate. If one or more interpretations are found, put the resulting CRC(s) on a temporary knowledge base of fine-grained knowledge.

4. Generalize the result by replacing the concepts in the CRC with their closest supertypes, using the structural overcommitment principle (Webster and Marcus, 1989). Add the result to a temporary knowledge base of average-grained knowledge.

[20] Acquiring type hierarchies is after the authors' meaning an open issue because mere property inheritance seems to be inadequate at fully modelling categorization in humans.

5. Repeat steps 1-4 for all the collocates of the same syntactic type, or (user choice) those including the same word W. Further generalize one step up in the hierarchy, based on at least three examples.

6. Present the results to a linguist for a final approval, then add to the permanent knowledge base.

At present the system has only been run on 3000 collocates with positive resultes.

5.2.2 Semantic Constraints

Semantic constraints based on semantic primitives have been extracted from machine-readable dictionaries. Different techniques have been adopted according to the structure of meaning definitions in the used machine-readable dictionaries. One technique used in the ACQUILEX project[21] (Vossen 1991b) is to look at the distributional behaviour of words in meaning definitions (e.g. a verb only co-occurs with humans in a particular argument slot). In one of the machine-readable dictionaries used in the ACQUILEX project, *Longman Dictionary of Contemporary English* (LDOCE), a semantic code, i.e. an indication of the class to which the specific sense of an entry belongs, is associated to many word entries. Examples of classes used in LDOCE are: abstract, concrete with the two sub-classes: animate (human, animal, plant) and inanimate (solid, liquid, gas). When word entries are not associated with a semantic code some heuristics, based on the word's collocational behaviour, have been set up for determining to which class that word sense belongs (e.g. food that is "made with something" is usually an artifact).

In *Collins Cobuild English Language Dictionary*, COBUILD, another machine-readable dictionary often used as source for semi-automatically extracting semantic knowledge, some semantic constraints are expressed in the meaning definitions in a way that is quite easy to make explicit (e.g. two of the entries for the word *buy* contain the following definitions: "if you buy something, you obtain it by paying for it"; "if someone buys someone else, they get their help or services by bribing or corrupting them". The person/object constraints can be easily derived from the fact that *you* and *someone* refer to persons while *something* and *it* refer to objects).

[21] An overview of goals of the ACQUILEX project can be found in Calzolari (1991).

It is often possible to semi-automatically extract semantic constraints from machine-readable dictionaries. If the dictionaries do not contain the word senses relevant to the actual domain, some of the techniques used with machine-readable dictionaries can be applied on a text corpus. E.g. if in the actual text corpus the verb *admit* is only cooccurring with direct objects indicating humans (*him, her, he, she* - the latter ones in passive constructions) the constraint that the subject for the admission must be a human can be automatically added to the system.

5.3 Manual Methodology of Refinement

Refinement of the characterization is a labour-intensive manual work with text corpus which can be supported with a number of tools but cannot be automatized completely.

The approach we take here is characterization of concepts in a framework of a corpus fragment schema. A schema for the corpus should be described as a collection of several schemata of its fragments which can be found on regular basis. For example, a structure of the PDS corpus is actually a structure of a PDS letter since the corpus is a collection of them. The PDS structure can be described as follows:

1. Header -
   a) Patient attributes: name, address, sex, DOB, etc.
   b) Context: hearer - Doctor, speaker - Consultant, theme - Patient, instrument - letter PDS.

2. Social greeting of the speaker to the hearer: Dear Dr XXX

3. Details of the Patient admission: when, where to, diagnosis, history of previous treatments...

4. Medical procedures performed in the hospital.

5. Details of the Patient discharge: when, medications on discharge, suggestions on further treatment.

Each lexical entry now can be marked according to the structural parts of text it can be found in and its frequency of occurence.

admit, admission, admitted: 1 -500, 2 -500.

{*cardiac catheterisation*}: 2 -386

{*left ventricular function*}: 2 - 12,

{*date*˜{{*day*˜}{*month*˜}{*year*˜}}}: 1 - 500, 2 -500, 5 -500.

{*amount*˜{{*quantity*˜}{*unit*˜}{*measure*˜}}}: 5 - 420.

For fragments with a structural representation (for example, tables) the knowledge engineer should create a schema and rules for translating the structure into it.

For textual information the following algorithm can be applied:

1. Choose a prototypical text fragment.
2. Determine the central concepts of the fragment
3. Take the characterization generated at the automatic phase as a template for refinement.
4. Look for all occurences of the corresponding lexical entry in the corpus to see the contexts in which it is used. First look for occurences of the entry in the analyzed fragment and then in other fragments. When necessary, look at previous or following sentences to resolve anaphoric and other references.
5. Apply a type oriented analysis with predefined type schematization.
6. Impose semantic constraints on the schema.
   a) in each citation determine fillers for the schema.
   b) find general characterizations for the fillers.
   c) if there are several cases for one filler - try to reduce them to one i.e. resolve metonymy.
7. Characterize morpho-grammatical features for the fillers.
8. Determine classes of modifiers which can be applied.

One of the first issues of this algorithm is to determine central concepts of a fragment. The following recommendations can help in this:

1. since verbs have the richest valency structures start from eventualities (both verbs and nominalized ones) with the highest frequency.
2. then analyze domain-specific terms.
3. analyze commonsense words.

The most difficult part of the analysis is the type-oriented schematization. Actually, for every type this schematizations should end up with a description of the allowed type coexistence, lexico-grammatical means for their representation and strategies for resolving irregularities. Eventualities (verbs and nominalized constructions) have the most complex schematization and in the following section we outline the methodology for their analysis.

5.3.1 Methodology for Analysis of Eventualities.

The analysis of eventualities is schematized on the basis of the framework suggested in [21]:

1. Choose the most domain specific eventuality in the text fragment.
2. Check if there is already any characterization of it.
3. Look for all occurences of the corresponding lexical entry in the corpus to see the contexts in which it is used. When necessary, look at previous or following sentences to resolve anaphoric and other references. For example, for the word “damage” in the auto-mechanical corpus (described in (Bech *et al.* 1993b) this results in the following:

Never jack under the {rear axle} as	damage	to these components may be incurred.
{Alloy wheels} use special nuts to prevent	damage	to a roadwheel.
If the transmission is	damaged	(...)
Anti-freeze will	damage	paintwork.
Incorrect {towing equipment} could	damage	your vehicle.
To assure proper towing, and to prevent accidental	damage	to your vehicle, (...)
If any unit is	damaged	(...)
Never place the {ignition key} in ... This will result in	damage	to the {steering lock mechanism}.
Never tow an {automatic transmission} model .. as this may cause serious and expensive	damage	to the transmission.
Operating with {insufficient amount} of oil can	damage	the engine
and such	damage	is not {covered by warranty}.
The use of other types of coolant solutions may	damage	your {cooling system}.
It is vital that the correct procedure is always followed (...) or the results could (...)	damage	the vehicle.
Driving even a short distance [with a deflated tire] can	damage	a tire.

4. Create a schema for the eventuality. We have to analyze the causative-inchoative structure of the event and characterize the connection between morphologically related forms.

   In our example with "damage" we have it as a transitive verb and as an adjective. Since the intransitive form for "damage" doesn't exist and "damage" specifies a final state rather than the action itself – it is causative-inchoative verb. For such a verb we can create the following paradigmatic schema:

   Ee [Damage(e) & Theme(e,th) & Instr(e,i) & Agnt(e,a) & Cul(e,t) & Time-point(t)]

   $\longrightarrow$

   Ee1 [Cul(e1,t) & Agnt(e1,a) & Instr(e1,i) & Theme(e1,f)
   & Es[Being-Damaged(s) & Theme(s,th) & Hold(s,ti) & Time-int(ti)
   & BECOME(e1,s)]
   & ti.beg = t
   ]

   This says that an unspecified event (e1) culminated and caused the theme of "damage" to become damaged i.e. to be in the state (s).

This schema corresponds to the event damage which is represented in language as the transitive verb “damage”. The adjective “damaged” corresponds to the state “s”. Also we represented temporal order: the time-point of culmination of “damage” is actually the time point of culmination of the unspecified event (e1) and it is the beginning point of the state “s”.

It is important to have such elaborated representation for eventualities since different modifiers should be attached to different parts of the conceptualization. For example, the modifier “slowly” is to be attached to the “e1” event while the modifier “seriously” to the resulting state “s”.

Another benefit from conceptualization of this kind is that when there is an explicit event which causes damage we already have the exact place for it - the event “e1”.

5. Impose semantic constraints on types of thematic roles in the event schema. Specify classes of modifiers which can be attached to different parts of the eventuality. At this stage we are going to create a highly domain dependent variant of “damage”. This variant is a specialization of the general schema for “damage”.

   a) reduce citations to the event schema:

   - Jack(e1), Ve-Component(th).
   - prevent an eventuality e1, Roadwheel(th)
   - Transmission(th)
   - Anti-freeze(f) - theme of e1, Paintwork(th)
   - Towing-equipment(f), Vehicle(th)
   - Accidental(e1), Vehicle(th)
   - Place(e1), Ignition-key(f), Steering-lock-mechanism(th)
   - Tow(e1), Automatic-transmission(f), Transmission(th)
     Serious(s), Expensive(s)
   - Operating(e1), Oil(i), Engine(th)
   - Covered-by-warranty(s)
   - Use(e1), Coolant-solution(f), Cooling-system(th)
   - Procedure(e1), Vehicle(th)

- Driving(e1), Tire(th)

b) find general characterizations for different cases. Types of thematic roles can be subsumed by their common supertype.

- Roadwheel(th), Transmission(th), Steering-lock-mechanism(th), Engine(th), Cooling-system(th), Tire(th) < Ve-Component; Anti-freeze(f), Towing-equipment(f), Ignition-key(f), Automatic-transmission(f) < Ve-Component(th);

  Improper operation of a car-owner with a car component can cause damage of a car component.

  Ee [Damage(e) & Theme(e,th) & Ve-component(th) & Agnt(e,a) & Ve-owner(a)]

  $\longrightarrow$

  Ee1 [Impr-Oper(e1) & Cul(e1,t) & Agnt(e1,a) & Theme(e1,f) & Ve-component(f)
  & Es[Being-Damaged(s) & Theme(s,th) & Hold(s,ti) & Time-int(ti) & BECOME(e1,s)]
  ]

- *Improper operation of a car-owner with a car component can cause damage of the car.*

- *Improper operation of a car-owner with a car component can cause damage to an attribute (paintwork) of a car component (body).*

- The resulting state of damage can be:
  Covered-by-warranty(s), Serious(s), Expensive(s).

  These modifiers for the state of being-damaged can be divided into two types: the first type determines a degree of damage - serious, expensive etc; the other one is just a characteristic. To represent the first case we should create a scale Dam-degree and several its subscales:

  Dam-degree-s: Seriousness
  Dam-degree-r: Recoverableness.
  Dam-degree-x: Expensiveness.

  Also these scales should be provided with mapping into one another. The canonical structure for this looks as follows:

  Es[Being-Damaged(s) & degree(s,d) & Dam-degree-mbr(d)]

  "Covered-by-warranty" is an attribute of a different kind and can be connected to the damage-state by the attributive relation:

Es[Being-Damaged(s) & char(s,w) & Warranty-cover(w)]

We can refine the domain vocabulary and the CTL at this stage. We can create a new supertype in the CTL for a thematic role arguments. We can create a new lexical phrase in the vocabulary. For example, {beyond repair} is a synonym for "unrecoverable" and should be mapped straight to a Dam-degree-mbr.

At this stage, however, we do not conceptualize that "damaged components need to be repaired, and repair costs money". Analysis of required implicatures is a separate phase of KB design and it may happen that it is not necessary to specify many obvious facts.

c) Find common ground for different cases. Quite obviously that case 1 and case 2 are very similar and now our task is to reduce one case to the other. This is a typical example of metonymy when the entire object (vehicle) is used instead of its component. As it was suggested in Hobbs (1984) metonymy resolution is more preferable than creation of several polysemus concepts.

Metonymy resolution requires extra-linguistic facts and in particular that a vehicle has components: Vehicle(x) —→Ve-Comp(y) & Have-comp (x,y). There are several ways how to interpret this fact for resolving the metonymy. However, we would suggest to state declaratively that instead of a vehicle component the whole vehicle can be used:

Ee [Damage(e) & Theme(e,th) & Ve-comp(th) & Have-comp(x,th) & Vehicle(x) & Agnt(e,a) & Ve-owner(a) & own(a,x)] —→
Ee1 [Impr-Oper(e1) & Cul(e1,t) & Agnt(e1,a) & Theme(e1,f) & Ve-comp(f) & Have-comp(x,f) & Vehicle(x)
& Es[Being-Damaged(s) & Theme(s,th) & Hold(s,ti) & Time-int(ti) & BECOME(e1,s)]
]

Also we stated the fact that both Ve-components - f and th - belong to the same vehicle. We can also specify a fact which goes in the opposite direction: *If a vehicle component is damaged the whole vehicle is damaged as well.*

Ve-Comp (y) & Have-comp(x,y) & Vehicle(x) & Es[Being-Damaged(s) & Theme(s,y) & Hold(s,ti)] —→
Vehicle(x) & Es1[Being-Damaged(s1) & Theme(s,x) & Hold(s1,ti)].

Note that the state of being damaged for vehicle is different from the state of being damaged for its component since for example a component can be damaged heavily but if its not very important the vehicle itself is damaged slightly.

Case 3 can be covered by the following facts:

Ve-body(b) →Paintwork(p) & attr(p,b)- *Paintwork is an attribute of the car body.*

Now we can state a fact that damage to a attribute means the damage to the thing as we did for the component and a vehicle.

6. characterize morpho-grammatical features of concepts to fill up certain positions in the schema.
   Damage - Noun:
   damage to Theme(th).

   to Damage - TV:
   Theme(th) - object :*He damaged his vehicle.*
   Event(e1) - subject: *Towing damaged...*
   Instr(i) - subject: *Hammer damaged...*
   Instr(i) - direct object + with: *He damaged it with hammer...*
   Agent(a) - subject: *The owner damaged...*

   Damaged - Adj:
   attributive construction: damaged Theme(th). Eth & Es[Being-Damaged(s) & Theme(s,th)].
   predicative construction: Theme(th) is damaged. Es[Being-Damaged(s) & Theme(s,th)]

7. characterize other eventualities in citations which are used with the target eventuality. In our example: *to make damage, to cause damage, damage incurred, to result in damage, to prevent damage*

   Intransitive verbs like: incur, occur, happen, take place - don't have content by their own and just assert culmination to an event or holding to a state. So we can specify:

   to incur - IV:
   Theme(e) - Event-headnoun ==→Event(e) & Cul(e,t).

   "To damage", "to cause damage" and "to result in damage" correspond to the same case. These constructions assert causality between damage and some other event. But since damage is itself causative-inchoative, this causality corresponds to the unspecified event (e1) in the schema of damage. So we can characterize one conceptualization of these verbs:

Lexical entries:

to cause, to {result in}
Conceptual entry:
1) CAUSE(e,e1) & Event(e) & Event(e1) & Cul(e,t)
2) BECOME(e,s) & Event(e) & State(s) & Hold(s,it).

Canonical structures - paradigmatic schema for causative and inchoative verbs.

"To prevent" is straight opposite to the previous case. This causes non-culmination of causative event:
Lexical entry: to prevent
Conceptual entry:
1) CAUSE(e,e1) & Event(e) & Event(e1) & Non-Cul(e1,t)
2) BECOME(e,s) & Event(e) & State(s) & Non-Hold(s,it).

6 Conclusion

There are many techniques and strategies which can facilitate the extraction of knowledge from natural language texts (and partly from other resources), but the knowledge acquisition process cannot be fully automatized. Especially the refinement of the knowledge base is a labour-intensive manual activity.

Some of the techniques and strategies which we have described in this report are particularly interesting because they provide a way of analysing large text corpora, making the acquisition process less dependent on few texts and on the judgement of a single knowledge engineer. In particular methods that cope with the sparseness of data, considering the "similarity" of occurrences among different words, can improve the quality of the resulting knowledge bases. Many of these techniques use the collocational behaviour of words in large text corpora, and their application on experimental basis has shown a strong connection between the syntactic and the semantic behaviour of words.

Because the material contained in large text corpora is quite huge, one could think that methods involving the use of text corpora would make the knowledge elicitation process much more labour-intensive. This is not always the case because there are available tools for automatically processing corpora of many million words and for computing different statistic calculations on these words in relatively short time.

One problem with many of the statistical techniques and methods for processing natural language is that they are rather new and have been only applied on experimental basis. Some of these techniques and algorithms must be refined and the generality of these techniques together with the observations of the semantic relatedness of word clusters collected with statistical methods must be proved. Thus the existing resources (tools, techniques lexical knowledge bases) which we believe can support the knowledge acquisition process must be evaluated and the methodology which we have defined must be refined i.a. by applying it to some test materials. Our future work in WP 3:4 will cover these two areas.

References

[1] A. Bech, A. Mikheev, M. Moens, and C. Navarretta. Typology for Information Contents. ET-12 Project Report 2, 1993.

[2] A. Bech, M. Moens, and C. Navarretta. Strategies in NLP knowledge engineering. ET-12 Project Report 1, 1993.

[3] M.R. Brent. Robust Acquisition of Subcategorizations from Unrestricted Text: Unsupervised Learning with Syntactic Knowledge. MS, John Hopkins University, Baltimore, 1992.

[4] P.F. Brown, V.J. Della Pietra, P.V. deSouza, Jenifer C. Lai, and R.L. Mercer. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479, 1992.

[5] N. Calzolari. Acquiring and Representing Semantic Information in a Lexical Knowledge Base. ACQUILEX, ESPRIT BRA 3030 16, 1991.

[6] J. Gennari, P. Langley, and D. Fisher. Model of Incremental Concept Formation. *Artificial Intelligence*, 31–40, September 1989.

[7] J.S. Gruber. *Lexical Structures in Syntax and Semantics.* North-Holland Publishing Company, 1976.

[8] V. Hatzivassiloglou and K. R. McKeown. Towards the Automatic Identification of Adjectival Scales: Clustering Adjectives According to Meaning. In *ACL Proceedings, 31st Conference*, pages 172–182, Columbus, Ohio, USA, 1993.

[9] J.R. Hobbs. Sublanguage and Knowledge. Technical Note 329, SRI, California, 1984.

[10] Y. Huizhong. A New Technique for Identifying Scientific/Technical Terms and Describing Science Texts. (An Interim Report). *Literary and Linguistic Computing*, 1(2):93–103, 1986.

[11] R.S. Jackendoff. *Semantic Interpretation in Generative Lexicon.* The MIT Press, Cambridge, Massachusetts, 1972.

[12] R.S. Jackendoff. *Semantics and Cognition.* The MIT Press, Cambridge, Massachusetts, 1983.

[13] J.S. Justeson and S.M. Katz. Principled Disambiguation: Discriminating Adjective Senses with Modified Nouns. In *Making Sense of Words, Proceedings of the 9th Annual Conference of the* UW *Centre for the New* OED *and Text Research*, pages 57–73, Oxford, England, sep 1993.

[14] G. Klose and K. von Luck. The background knowledge of the LILOG system. In O. Herzog and C.-R. Rollinger, editors, *Text Understanding in LILOG. - Integrating Computational Linguistics and Artificial Intelligence. Final Report on the IBM Germany LILOG-Project*, number 546 in Lecture Notes in Artificial Intelligence, pages 455–463. Springer–Verlag, Germany, 1991.

[15] J.D. Lafferty and R.L. Mercer. Automatic Word Classification Using Features of Spellings. In *Making Sense of Words, Proceedings of the 9th Annual Conference of the* UW *Centre for the New* OED *and Text Research*, pages 89–703, Oxford, England, sep 1993.

[16] S.C. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, England, 1983.

[17] R.-L. Liu and V.-W. Soo. An Empirical Study on Thematic Knowledge Acquisition Based on Syntactic Clues and Heuristics. In *ACL Proceedings, 31st Conference*, pages 243–250, Columbus, Ohio, USA, 1993.

[18] C.D. Manning. Automatic Acquisition of a Subcategorization Dictionary from Large Corpora. In *ACL Proceedings, 31st Conference*, pages 235–242, Columbus, Ohio, USA, 1993.

[19] C. Navarretta. Criteria for 'support material'. ET-12 Project Report 3, 1993.

[20] S. Nirenburg, editor. *Machine Translation. Theoretical and Methodological Issues*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, 1987.

[21] Terence Parsons. *Events in the Semantics of English*. MITPRESS, 1990.

[22] F.P. Pereira, N. Tishby, and L. Lee. Distributional Clustering of English Words. In *ACL Proceedings, 31st Conference*, pages 183–190, Columbus, Ohio, USA, 1993.

[23] J.F. Sowa. *Conceptual Structures: Processing in mind and machine.* Addison-Wesley Publishing Company Inc., 1984.

[24] P. Velardi, M.T. Pazienza, and M. Fasolo. How to Encode Semantic Knowledge: A Method for Meaning Representation and Computer-Aided Acquisition. *Computational Linguistics*, 17(2):154–170, 1991.

[25] P. Vossen. Converting Data from a Lexical Database to a Knowledge Base. ACQUILEX, ESPRIT BRA 3030 27, 1991.

[26] M. Webster and M. Marcus. Automatic Acquisition of the Lexical Semantics of Verbs from Sentence Frames. In *Proceedings of the ACL, 27th Conference*, Vancouver, Canada, 1989.

[27] P. Zweigenbaum, B. Bachimont, J. Bonaud, and M. Ben-Said. MENELAS Linguistic and Conceptual Knowledge (version 0.1). Menelas deliverable, 1993.

A Investigations in the Type System Organization

A.1 Analysis of Top-Level Categories

Different systems use different ontologies. They employ ten or twelve generally agreed categories, though they are distributed differently along the tree.

The first known top-level classification is the Aristotelian one (Sowa 1984). It includes 10 categories: SUBSTANCE, QUALITY, QUANTITY, RELATION, PLACE, TIME, POSITION, STATE, ACTIVITY, PASSIVITY.

Jackendoff in his Semantic & Cognition theory (Jackendoff 1983) uses the following top-level (primitive) conceptual categories: THING(or OBJECT), EVENT, STATE, ACTION, PLACE, TIME, PROPERTY, PATH, AMOUNT.

The LILOG (Klose *et al.* 1991) system also distinguishes a top-level hierarchy (upper structure) from middle and low level structures. This upper structure is represented on fig.1. The Aristotelian categories have the following correspondence to the LILOG ones: ACTIVITY - direct event; PASSIVITY - indirect event, SUBSTANCE - object, PLACE and POSITION - spatial concept, TIME - temporal concept.

THING →	Entity.		
	→	Object.	
	→	Eventuality.	
	→	→	State
	→	→	Direct Event
	→	→	Indirect Event
	Abstract Concept		
	→	Spatial.	
	→	Temporal.	
	→	Qualitative.	
	→	→	Measures
	→	→	Units
	→	→	Names
	→	→	....

fig.1

Another well-known top-level ontology was used in the KBMT system (Nirenburg 1987). Like in LILOG it makes an initial distinction between Objects, Events and Properties (fig 2), but the further CTL branching seems to be very implementation oriented and suffers from a lack of generality.

```
CONCEPT →  OBJECT.
                  →           Physical.
                  →  Representational.
                  →             Social.
        →      Event
                  →      Superordinate
                  →           Physical
                  →            Complex
                  →             Social
                  →           Temporal
                  →              Cause
                  →             Effect
        →   Property.
                  →           Relation
                  →          Attribute
```

fig.2

Many other systems are even less consistent with the top-level hierarchy. They include very task specific and implementation dependent concepts and don't follow the structural way.

For example, the domain independent top-level hierarchy of the MENELAS project (Zweigenbaum *et al.* 1993) (fig. 3) was created on the empirical basis i.e. without pre-formalization of the main categories. In result this hierarchy suffers from mixing up categories, properties and domains. The role type of the Menelas' ontology resides as an independent and separate type, though all conceptual types can be used in a particular role. So the role is not a type as it is but a property which can be applied to any type. In our opinion, the conceptual type *social-object* does not really belong to the general hierarchy but rather corresponds to the domain of roles for objects.

Entity →	OBJECT.				
		→	Physical.		
		→		→	Natural
		→		→	Artificial
		→	Conceptual.		
		→	Social.		
→	TIMED.				
		→	Proposition.		
		→	State.		
		→	Event.		
→	TIME				
		→	Time-stamp.		
		→	Duration.		
		→	Frequency.		
→	Space				
→	Attribute				
→	Value				
→	Measure				
→	Unit of measure				
→	Characteristic				
→	Role				

Fig.3

Almost all ontologies mix up conceptual categories with domains these categories belong to. This leads to the fact that when such an ontology is being adopted to a new domain it requires substantial modification. For example, time and space are domains which include many concepts of different nature, rather then conceptual types which subsume these concepts.

A.2 Ontological primitives

The three main categories (using different names) are presented in almost all ontologies. These categories are: Objects, Eventualities and Properties. Simplistically, in the world there are objects which have certain properties and are in certain states and events which change states of this objects.

Usually objects and eventualities have a concept status on their own. So they can be instantiated and one can quantify over them.

Properties, though in some approaches they also have a concept status and can be instantiated, do not exist by their own and can be measured.

To find out a formal account for ontological primitives we will try to investigate the difference in their nature. The main principle of conceptual organization is that concepts are linked with relations between each other. Introducing classification into different conceptual types we impose constraints on permissible configurations of concepts and links.

A.2.1 Two Primitive Links

In the very beginning we need to distinguish two main types of arbitrary relation "link". The first type is relation of equivalence "=" and the second one is attributive relation "attr".

The "=" link formalizes different patterns of relations depending on a referential type of a concept. The referential type is represented by quantification over a concept: every, some, a, individual. Every concept can be represented as a box with two fields: type field and referential field. The "=" relation is reflexive, transitive and symmetrical. One can distinguish the following cases:

- [TYPE-1:every]←(=)→[TYPE-2:every] means that both types have the same extension but are different in intention.
- [TYPE-1:every]←(=)→[TYPE-2:some] represents that TYPE-1 is a subtype of TYPE-2 (terminological link).
- [TYPE-1:some]←(=)→[TYPE-2:some] represents that TYPE-1 and TYPE-2 have an non-empty intersection.
- [TYPE-1:individual]←(=)→[TYPE-2:a] represents that individual TYPE-1 is an element of type TYPE-2. (assertional link). This also implies that TYPE-1 = TYPE-2.

All other combinations with the link "=" are considered as non-valid.

The other relation ("attr") is relation of structural inclusion of one concept into another. Since there are many types of inclusion the "attr" relation stands as a generic one. Its main property is asymmetry and this implies that there should be constraints on different conceptual types to be related with this link.

A.2.2 Conceptual meta-types: Entities and Properties

Since we adopted an approach that every distinguishable phenomena in the outside world can be represented as a concept in the KB we should explore different natures of these concepts. First, we distinguish conceptual entities whose intention is a set of necessary and sufficient properties. Both entities and properties are concepts in our approach but they have very different ontological status.

A property concept exists only in association (linked with the "attr" relation) with some other concept it is founded on. Destruction of the main concept implies destruction of all its properties. Destruction of a property concept is possible only with destruction of its main one and implies this. Concepts which are not founded are called essentially independent. We will call essentially independent concepts entities.

To represent this we can create a subtype of the "attr" link - characteristic:
[ENTITY:every]→(char)→[PROPERTY:some]
this says that every entity is characterized by some properties.

Examples:

1. the type COLOR is founded on the type OBJECT and there is no "=" relation between them, so COLOR is a property of OBJECT:
   [OBJECT:every]→(char)→[COLOR:a]; OBJECT<ENTITY; COLOR<PROPERTY
   This says that: a) every physical object (OBJECT) is essentially independent and hence is an ENTITY; b) every color (COLOR) is a PROPERTY; c) every object has (is characterized by) a color.

2. the type WHEEL is an attribute of a type CAR but it is not founded on it, so WHEEL is not a property of CAR.

3. the type CHILD is the "=" relation with the type PERSON so CHILD is not a property of PERSON.

Properties also have different sense of instantiation: an instance of a property is its value from a range of its possible values. If an entity can be interpreted as a set of individuals, properties are interpreted as sets of sets. Quantification over a property is a quantification over a set of its values. Some properties have an unordered set of values, for example SEX:= {MALE, FEMALE}, but many properties can be described as ordered scales and when instantiated can express a degree, amount etc.

A.2.3 A starting ontology

Here we present a starting ontology of top-level conceptual types. However it is not obligatory for a new type to be reduced to one of these. If such characterization can be provided it will be easier to describe a type using already predefined patterns. These patterns can be used for representing internal structure of a concept. In many cases it is not necessarily or even possible to provide an internal structure for a concept. In this case it can be characterized only by its external relations and thus be of none of the top-level types.

Concept								
	→	Entity						
			→	Object.				
					→	Physical.		
					→	Conceptual.		
			→	Eventuality.				
					→	State.		
					→	Event.		
	→	Property						
			→	Quantitative.				
			→	Qualitative.				
	→	Measure						
			→	Unit of measure				
			→		→	Scalable		
			→		→		→	Numerical
			→		→		→	Fuzzy
			→		→	Non-Scalable		
			→	Values				
			→		→	Numbers		
			→		→	Strings		

The following general patterns of relations between top-level categories exist:

[ENTITY]→(char) →[PROPERTY]

[QUANT-PROPERTY]→(meas) →[MEASURE]

[QUAL-PROPERTY]→(val) →[VALUE]

[MEASURE]-
→(unit) →[UNIT-OF-MES]
→(val) →[VALUE]

1) →(val) →[VALUE] : →(degree) →[NUMBER]
2) →(val) →[VALUE] : →(kind) →[VALUE]→(name)→[STRING]

[CONCEPT]→(char) →[NAME] →(meas) →[STRING]

Properties and Measures Two elementary types NUMBER and STRING already have their conceptualization in the way they are used in all computational tasks:

[CONCEPT:every] →(char) →[NAME] →(val)→[STRING:every]

Every property is distinguishable by its own measures. Actually for a property there can be several measures but all this measures should be compatible.

There are two different types of measures - scalable and non-scalable. Scalable measures allow one to compare different values with each other. Numerical scales represent exact values and fuzzy scales are constructed out of grades represented by strings. There can be mapping functions between these two types of scales. Example:
[EVENT]→(cul) →[TIME-POINT] →(meas)→[TIME-SCALE]
TIME-SCALE < SCALABLE
TIME-SCALE-ABS < NUMERICAL
TIME-SCALE-REL < FUZZY (...Yesterday, Today, Tomorrow...)
TIME-SCALE-WK < FUZZY (Sunday, Monday, Tuesday...)

Conceptualization of scales can be done in different ways, though we suggest to do this in a procedural way since many computations for relating one scale to another require quite sophisticated procedures.

Scalable measures can be supplied with comparative and evaluative grades. These grades usually have the same construction but different mapping rules to the scale. The prototyping evaluative grade can have the following structure:

EVAL-GRADE-BASIS: Big – Small
{Smallest, .. Very Small,...Small,..Normal,.... Big,.. Very Big, Biggest}

Non-scalable measures are constructed out of disjoined values represented by strings. These values are not ordered and cannot be compared to each other. An example of a non-scalable measure is the type SEX-MEAS := {Male, Female}. The property type SEX is distinguishable by this measure:
[ANIMATE]→(char) →[SEX] →(meas)→[SEX-MEAS]

SEX-MEAS < NON-SCAL
SEX-MEAS := {Male, Female}.

Eventualities Eventualities can be instantiated only in respect of a time point or time interval. They have participants (objects) and are either culminating in time or holding. They are represented as verbs in language but these verbs often can be nominalized. Thematic roles are used for classifying the arguments of natural language predicates into a closed set of participants types of an eventuality. A list of the most popular roles includes: agent, patient, experiencer, theme, location, source, goal, instrument. Thematic roles form a hierarchy which can capture number of linguistic and extralinguistic generalizations.

Events culminate in a time point: Ee & Cul(e,t). States hold in a time interval: Es & Hold(s,ti). There are two kinds of events: achievements and accomplishments. The nature of achievement events is instanteneous while the accomplishments have a duration. There is also an intermediate eventuality - processes. They have nature of events but are holding in time. In many cases we don't distinguish them from events since almost any event can be seen in a process but instead of culmination they are linked with holding : Ee & Hold(e,ti).

There are several linguistic tests which allows one to distinguish between events, states and process:

- Event (Process) vs. State:"What he did was VERB-EVENT something"
  Example:
  OK. Process: What John did was run.
  OK. Accomp: What John did was make a birdbath.
  OK. Achiev: What John did was win the race.
  Bad. State: What John did was know the answer.

- Achievements vs. Others:"How long did PROCESS/ was STATE".
  Example:
  OK.State: How long was his face red.
  OK.Process: How long did Mary run.
  OK.Accompl: How long did Mary make a sandwich.
  Bad.Achievm: How long did John win the race.

- Event (Process) vs. State:
  "How long did it take to EVENT(PROCESS)"
  "How long did STATE last".

- Event vs. Process:
  "If x V-ing than x not V-ed" - event;
  "If x V-ing than x V-ed" - process. Example:
  Process. If he is running then he has run.
  Event. If he is building a house he has not built a house.

There are two kinds of existence for an eventuality. The conceptual existence (existential quantification) does not imply existence in the world. Real existence is shown by means of culmination and holding.

Examples:

- John's running. Ee[Running(e) & Agnt(e,John) & Cul(e,t)]

- John's cancelled running. Ee[Running(e) & Agnt(e,John) & Non-Cul(e,t)]

The concept of running even if it were cancelled exists and is linked with certain time-point.

Objects Objects can be instantiated by themselves and are distinguishable by a set of properties they have. Actually there are at least two kinds of existence for an object. The first one is existence of an object as a concept. The second one is its materialization in the world. The existential quantification corresponds to the conceptual existence, the state Exist corresponds to materialization.

The conceptual existence allows us to represent facts about objects which do not exist in the world at the moment, for example, Napoleon. The material existence can represent the fact of existence in the world and also the fact of planned but not materialized existence.

Examples:

1. Computer (which never existed).
   Ex & Computer(x)

2. Andrei Mikheev. He is alive.
   Ex & AM(x) & Es[Exist(s) & Theme(s,x) & Hold(s,ti) & ti=now]

3. Napoleon. He lived but now he is not alive.
   Ex & Napoleon(x) & Es[Exist(s) & Theme(s,x) & Hold(s,ti) & ti<now]