IJCLA VOL. 1, NO. 1-2, JAN-DEC 2010, PP. 249-267
RECEIVED 23/11/09 ACCEPTED 16/01/10 FINAL 08/03/10

Using Linguistic Knowledge for Fine-tuning Ontologies in the Context of Requirements Engineering

JÜRGEN VÖHRINGER[1], DORIS GÄLLE[1], GÜNTHER FLIEDL[1], CHRISTIAN KOP[1], MYKOLA BAZHENOV[2]

[1] *Alpen-Adria-Universität Klagenfurt*

[2] *National Technical University "Kharkov Polytechnic Institute"*

ABSTRACT

Nowadays ontology creation is on the one hand very often hand-knitted and thus arbitrary. On the other hand it is supported by statistically enhanced information extraction and concept filtering methods. Automatized generation in this sense very often evokes "shallow ontologies" including gaps and missing links. In the requirements engineering domain fine-granulated domain ontologies are needed; therefore the suitability of both hand-knitted and automatically generated gap-afflicted ontologies for developing applications can not always be taken for granted. In this paper we focus on fine-tuning ontologies through linguistically guided key concept optimization. In our approach we suggest an incremental process including rudimentary linguistic analysis as well as various mapping and disambiguation steps including concept optimization through word sense identification. We argue that the final step of word sense identification is essential, since a main feature of ontologies is that their contents must be shareable and therefore also understandable and traceable for non-experts.

Keywords: requirements engineering, ontology engineering, rule mapping, incremental linguistic analysis, WordNet querying

1 INTRODUCTION/MOTIVATION

In the last ten years the job of creating ontologies moved from an Artificial-Intelligence question to a central topic of the exploding semantic web community [17]. Usually ontology creation is very often hand-knitted and thus arbitrary or supported by statistically enhanced information extraction and concept filtering methods. This shift resulted in an uncontrolled growth of ontologies on the one hand and a heightened degree of ontology generality on the other hand. Both developments entail that many existing ontologies are not usable in real-world-applications like requirements engineering.

For supporting systematic and application oriented ontology engineering we previously researched and proposed linguistic guidelines for structuring concept and property notions in OWL represented ontologies [1]. But these guidelines only support ontology generation if domain expertise is sufficiently available in a clearly decoded manner. Obviously specific domain information quite often exists in a not explicit and ambiguous textual format. In this case the elicitation of domain specific concepts still poses many difficulties to the ontology designer.

Hence we developed a linguistic system for supporting the ontology designer to make implicit information easier to trace. Our methodology includes algorithms for tagset mapping, multi-level chunking and wordesense identification. The tagging task is carried forward to QTAG, a probabilistic tagger written in Java [2]. The mapping engine we developed for splitting up standard tags into ontologically relevant tags and specific attributes generates unique input for our rule based chunker.

Some chunking heuristics needed for grouping words to morphological units and syntactical chunks are then used for decoding linguistic candidates for conceptualization nodes in the ontology layer. The main contribution of our research work presented in this paper is an efficient method for incrementally including contextual information in the ontology representation. By combining standard natural language processing methods with certain expansion strategies we definitely improve the usability of standard ontologies. Our approach preserves the basic and partially generic knowledge format for storing domain knowledge and its guided updating.

The approach consists of the following three main steps:

1) linguistic preprocessing: extracting words and phrases from natual language text

2) linguistically guided incremental ontology engineering
3) filling up ontology concept description slots through WordNet based word sense identification

The paper is structured as follows. In section 2 we give an overview of related work. In section 3 we roughly present our theoretical approach including the description of the linguistic pre-processing layers, in particular the mapping, and multi-level chunking steps. In section 4 the output of our ontology refinement tool is shown and an ontology example is described. We also propose a list of rules for ontology element creation. In section 5 we describe our Wordnet based tool for incremental optimization of standard ontologies through wordsense disambiguation. Section 6 gives a summary of the proposal presented in this paper.

2 Related Work

[29] argue that the accuracy and robustness of automatically or semi-automatically engineered ontologies needs to be improved for real-world applications and they propose fuzzy algorithms for real-world-ontology engineering. [28] proposes the use of glosses in ontology engineering for improving the accuracy. We agree that for real-world applications like ontology engineering in requirements engineering projects, automatically generated ontologies might not be suitable and we therefore propose linguistic heuristics for supporting ontology creation and fine-tune ontologies through step-by-step integration of domain knowledge.

Concerning linguistic preprocessing the most relevant linguistic methods used in our approach are tagging and chunking. For tagging English free texts many open source systems like the decision based “Treetagger” [3], the rule- and transformation-based “Brill tagger” [4], the maximum-entropy “Stanford POS Tagger” [5], the trigram based probabilistic “QTAG” [2] etc. are available. For chunking some NLP toolkits exist, e.g. “MontyLingua” [8], “MontyKlu” (an online-version of “MontyLingua” developed by members of our research group in Klagenfurt [9]), the OpenNLP chunker [10] and the “NLTK Toolkit” [11]. These systems mainly provide standardized and acceptable output, but as we know according to practical requirements engineering needs they have not been tested yet.

3 LINGUISTIC PREPROCESSING

3.1 Extended Tagging format

We have chosen “QTAG” as the basis for our extended tagging format which we have adopted for these special purpose. Since QTAG is a java-based, extendable, trainable, language independent tagger, it was easy to integrate in our engineering toolset [6,7]. We extract relevant information from the QTAG output and transform it into the extended tagset format described below. Therefore, we have to use some additional methods and heuristics to elicit semantic information needed during the further processing steps of the engineering workflow. Our enriched tagset consists of standard POS-categories with lists of additional specialized attributes (e.g. v0 with subclass attribute “tvag2”[3]). These attributes are necessary for identifying ontological key relations. Table 1 shows how typical standard part-of-speech tags are extracted from the QTAG output and reassigned using the NIBA tagset notation[4]. Additional information about concrete part-of-speech instances is presented by using fine-granulated attributes[5]. As an example, the verb “is” in QTAG gets the tag <BEZ>. This tag decodes, that “is” is an auxiliary verb with the inherent morphosyntactic values present tense, singular, third person and having “be” as the base form.

Table 1. Mapping Rules for mapping standard tags to attributed tags

BEZ	<=>	v0	verbclass="aux" temp="pres" form="ind" num="sg" ps="3" baseform="be"
NPS	<=>	n0	type="proper" num="pl"

[3] We use “tvag2” for annotating a mono-transitive verb with agentive subject

[4] Central NIBA tags are e.g. v0 (= main verbal element), n0 (= noun), a0 (= adjective) etc.

[5] Typical tag internal attributes are “base form = go” or “type = common” etc.

3.2 Chunking rules

Based on some variants of the X-bar Theory [24] and on some core definitions in the existing NIBA Tag system [25] we composed a set of chunking rules for English for the production of syntactically and morphosynctactically motivated chunks (Table 2).

Table 2. Extended Chunking Rules

Rule (Summands → Result)	**Rule level**	**Rule descriptions**	**Examples**
n0+n0 → n0	1	Compound Noun	blood pressure
[pt0]+a0 → a2	1	Adjective Phrase	very nice
[a0]+a0 → a2	1	Adjective Phrase	bright green
[pt0]+q0 → q2	1	Quantor Phrase	very many
[q0]+q0 → q2	1	Quantor Phrase	one million
[pt0]+adv0 → adv2	1	Adverb Phrase	very often
[adv0]+adv0 → adv2	1	Adverb Phrase	yesterday noon
pron0(type=pers) → n3	1	Noun Phrase	she
v0(verbclass=aux)+[adv0]+v0 → v0(type=complex)	1	Complex Verb	will certainly go
v0(verbclass=aux)+pt0(type=neg)+v0 →v0(type=complex)	1	Complex Verb	would not write
v0+pt0(type=verbal) → v0(type=complex)	1	Complex Verb	wake up
q2+q2 → q2	2	Quantor Phrase	two hundred million
pron0(type=poss)+n0 → n3	2	Noun Phrase	his mother
[det0]+[a2]+[q2]+n0 → n3	3	Noun Phrase	the nice two girls
[det0]+[q2]+[a2]+n0 → n3	3	Noun Phrase	the three busy scientists
p0+n3 → p2	4	Prepositional Phrase	of blood pressure measurement

There are several types of chunking rules, which are arranged in a certain order that should be followed during the chunking process. Summands are the array of input nodes which are needed for building the next resulting upper node of the chunking tree. Some of summands are strictly required for rule producing, they are written without square brackets, but some are not obligatory, they are placed inside brackets.

3.3 Identification of Semantic Roles

Due to the fixed and transparent subject-verb-object (SVO) structure of English, the identification of semantic roles in chunked sentences is by default a quite simple and straightforward task. According to [31] we propose automatic role labeling using partially mainly propbank and verbnet information. The Verbclass Tag in column a in Table 3 of a concrete verb triggers the assignment of a role in column c to a N3(P2) in d via indexation from left to right.

Table 3. Verb classes and their (morpho)syntactic and semantic features[23]

Nr.	Tag (a)	Verbclass (b)	PAS[6] (c) (Argument Structure)	Syntactic context[7] (d)
1	aux	Auxiliary verb	V-fin	_V0
2	eV	Ergative verb	$[TH_i]$[8]	$N3_i$_
3	iV	Intransitive verb	$AG_i/TH_i[]$	$N3_i$_

[6] PAS = "Predicate Argument Structure"; it includes the verb class specific semantic roles and brackets, which decode the argument status of these roles. They can have an external status (subject function) or an internal status (object function).

[7] P2 stands for prepositional phrases; N3 decodes nominal phrases in our framework; N2 is a reduced nominal phrase in predicative function; N3 A2 decodes an adjective phrase in our framework. For further explanation see [23].

[8] The acronyms for the default semantic roles are TH = Thema (neutral object), AG = Agens(the Actor of an Action), GO = Goal (the final point of a process), SO = Source (the starting point of a process), LOC = Location and EXP = Experiencer (a person, who undergoes the process of experiencing something).

4	lokV	Locations verb	TH_i [LOC_j]	$N3_i$_$P2_j$
5	possV	Possessive verb	GO_i [TH_j]	$N3_i$_$N3_j$
6	psychV	Mental verb	TH_i [GO_j]	$N3_i$ _$N3_j$
7	tvag2	Monotransitive verb with agent subject	AG_i [TH_j]	$N3_i$ _$N3_j$
8	tv3	Ditransitive verb	AG_i[TH_j,GO_k /SO_k]	$N3_i$_$N3_j$ P2 $(N3)_k$
9	sentV	Perception verb	EXP_i [TH_j]	$N3_i$_$N3_j$
10	copV	Copula verb	TH_i [$Pred_j$*/$Pred_k$*]	$N3_j$_$A2_j$/$N2_j$
11	tv2	Monotransitive verb without agent subject	TH_i_TH_j	$N3_i$ _$N3_j$

Nevertheless we have to take into account that the phrasal structure sometimes inhibits simple solutions like for example left to right counting of nouns. Thus we used the following algorithm to cope with the problem of phrasal complexity:

- create a set of rules which can operate on simple singular term subjects and objects (e.g. proper nouns and personal pronouns);
- consult the exception database with already assigned verbal subclass tags using training sentences which include higher level argument patterns referring to more complex phrases;
- reconstruct the structure of the primarily assigned phrases if relevant morphosyntactic features don't fit;
- leave open the possibility to manually change wrong/exceptional assignments or to add new information about verb classes, noun phrases and other patterns;

4 OUR APPROACH: LINGUISTICALLY GUIDED INCREMENTAL ONTOLOGY CREATION

To avoid using non-fitting ontologies for specific domain relevant demands, particularily in requirements engineering, we take textual descriptions as a starting point for our processing. These texts are generated by filtering those text segments from extensive, domain-relevant documents, in which key words or key phrases occur, which can be accepted as candidates for concept- or relation-notions in the ontology. Utilizing various filtering strategies, in a first step kewords in a text are identified, which are deemed important for a specific domain. Afterwards sentences from the original requirements that contain those keywords are filtered. A precondition is that these sentences form a cohesive text block. For further information about this process see [30]. It was produced.

In the following an examplary text segment from the medical domain is given, that was automatically selected from the original domain-related requirements text using the previously mentioned keyword filtering strategy:

```
With regard to the monitoring of blood pressure measurements, it is important to clearly define time and date at which the blood pressure of a hemodialysis patient is measured in each hemodialysis session.
```

We perform the steps of linguistic preprocessing as proposed in section 3 as a first step of transforming the textual input into a domain ontology:

- QTAG output (standard tags)
- Standard Tags transformed to enriched tags
- Chunking output

The XML output of the linguistic preprocessing can be seen in Fig. 1.

This output contains linguistic tags for words, e.g. n0: “regard”, some attributes (e.g. base-form=”regard”, type=”common”, corelex=”coa” etc.) and chunk-tags (e.g. n3: “the monitoring of blood pressure measurements”). This extended linguistic representation of the input text allows mapping and interpretation in the sense of ontology conceptualization. The Table 4 lists rules used for identifying and creating ontology elements from the preprocessed texts.

In the Table 4 some rules for the most relevant linguistic categories like N3, N0, P0 are listed. The interpretation example in the right column shows that an explicit mapping from text to class names is possible. To sum up: all relevant ontology element types are identified in an unambiguous way. The above listed rules transform linguistic annotators to ontological tags. The tags specify words in a unique manner. The output text below shows strings class candidates, relation designators, attribute identifiers and stop word material, which is filtered out during transformation:

With regard to the **monitoring** of **blood pressure measurements**, it is important to clearly define time and date at which the **blood pressure** of a **dialysis patient** is measured in each **hemodialysis session.**[9]


```
<p2>
<p0 idiom="pof1" idiomphrase="with regard to ">With</p0>
<n3>
<n2>
<n0 num="sg" idiom="pof2" derivedPOS="v0" idiomphrase="with
regard                          to                          " base-
form="regard" type="common" corelex="coa">regard</n0>
<p2>
<p0 idiom="pof3" derivedPOS="ip0" idiomphrase="with  regard  to
">to</p0>
<n3>
<det0 form="general" type="def">the</det0>
<n2>
<n0 num="sg" derivedPOS="v0" base-
form="monitor" type="common">monitoring</n0>
<p2>
<p0>of</p0>
<n3>
<n0 desc="compound" type="common">
<n0 desc="compound" type="common">
<n0 num="sg" derivedPOS="n0" base-
form="blood" type="common">blood</n0>
<n0 num="sg" base-
form="pressure" type="common">pressure</n0>
</n0>
<n0 num="pl" base-
form="measurement" type="common" corelex="ate">measuremen
ts</n0>
</n0></n3></p2></n2></n3></p2></n2></n3></p2>
```


Fig. 1. XML output of linguistic preprocessing for a fragment of the example text

[9] Underlined words decode relations , dotted underlines words function as attributes, **Bold words** are interpreted as classes; All other elements are categorized as stop words and filtered out.

Table 4. Rules for mapping of linguistic categories to ontology elements

Rule Nr	*Rule*	*Description*	*OWL Type*	*Example*
1	N0 → Class	Default-Rule for N0 if no Exception applies (see Rule 2)	Class	*"monitoring" → monitoring (class)*
2	N0 (Exception) → Functional Property	Exception for N0 applies, ifN0 is found (according to rules described in [26])	Functional Property	*"time" → time (slot in class blood-pressure)*
3	N3 → Class	Rules 3 to 5 always apply for N3	Class	*"hemodialysis session" → hemodialysis session (class)*
4	N3 → is_a + Class[10]	Rules 3 to 5 always apply for N3	subClassOf Class	*"hemodialysis session" → isa (subClassOf) Seassion (class)*
5	N3 → belongs_to + Class[11]	Rules 3 to 5 always apply for N3	Functional property Class	*"hemodialysis session" → belongs_to (Functional Property) hemodialysis (class)*
6	(P0 |		Functional	*"blood*

[10] The head (right-most part of the compound) becomes a new class

[11] After Rule 4 the head is removed and the remaining of the original N3 becomes a new class

	AUX0_V0) + Singular → Functional Property		Property	*pressure of a hemodialysis patient" →"bp_of"*
7	(P0 | AUX0_V0) + Plural → Object Property	Cardinality of connection is n	Object Property	*"monitoring of blood pressure measurements" →"m_of*"*
8	(N0 | N3) +"corelex=hum" → is_a + Class "human"		Functional property Class	*"hemodialysis patient" → isa (subClass Of) human (class)*
9	tvag2 → Functional property + class "agens"		Functional property Class	*"blood pressure is measured" → is_measured (Functional Property) agens (class)*

Fig. 2 shows an ontology fragment which is generated by applying the transformation rules in Table 4. For representation of ontology relevant knowledge OWL [15] and RDF [16] are commonly used [12]. We chose Protégé for representing our ontology example.

[12] Exemplary modern toolkits for ontology engineering are Protégé [12], NeOn [13] and Chimaera [14].

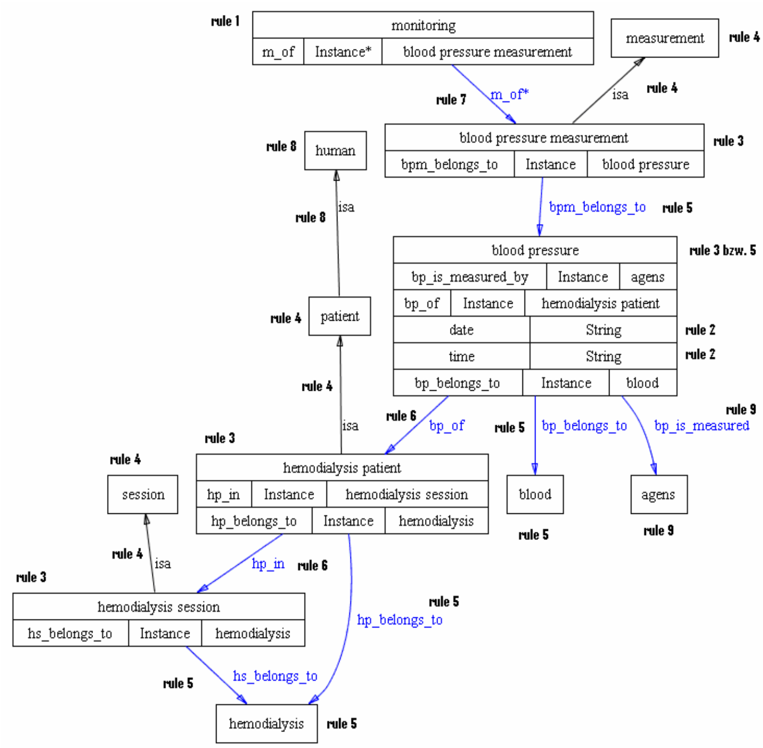

Fig. 2. OWL-Visualization of the above text via the Protégé [12]-Plugin Ontoviz

5 ADDING CONCEPT DESCRIPTIONS THROUGH LEXICALLY DRIVEN WORD SENSE IDENTIFICATION

According to Gruber [27] an ontology is "[…] a formal explicit specification of a shared conceptualization". From this follows that ontology contents has to be sharable and reusable among and across projects within the domain. The ontology fragments that were created stepwise from natural language text based on the heuristics described in section 3 still have empty description slots and are therefore not easily shareable and understandable for non-domain experts. Furthermore missing empty description slots make similarity calculation based on WordNet difficult, since they presuppose a known word sense. Such similarity calculations are important when new concepts are matched with existing ones in further ontology integration steps. For this reason we propose additional measures for fine-tuning the ontology by refining concept notions that are crucial for the specific domain. The description slots of the ontology (e.g. the example ontology in figure 2) are filled by providing a WordNet related engineering mechanism, which will be described in the following. For ontology concepts that have an empty description slot, WordNet is queried regarding the available word senses and their definitions. The following cases are distinguished:

Case 1: The concept is identified in WordNet and has exactly one meaning. This meaning is automatically assigned to the concept. Example: the concept *blood pressure* was identified from the natural language text, is new to the domain ontology and therefore has an empty description slot. Querying WordNet returns one possible meaning:
***blood pressure** -- the pressure of the circulating blood against the walls of the blood vessels; results from the systole of the left ventricle of the heart; sometimes measured for a quick evaluation of a person's health; "adult blood pressure is considered normal at 120/80 where the first number is the systolic pressure and the second is the diastolic pressure"*
This definition is chosen and assigned to the concept, but can still be manually adapted.

Case 2: The concept is identified in WordNet but has more than one possible meaning. In this case the correct meaning is chosen from the list of available word senses. Example: the concept *blood* has an empty description slot and querying WordNet returns the following possible Word Senses, ordered by probability of appearance:
*1. **blood** -- the fluid (red in vertebrates) that is pumped by the heart; "blood carries oxygen and nutrients to the tissues and carries waste products away"; "the ancients believed that blood was the seat of the emotions"*
*2. lineage, line, line of descent, descent, bloodline, blood line, **blood**, pedigree, ancestry, origin, parentage, stemma, stock -- the descendants of one individual; "his entire lineage has been warriors"*
*3. **blood** -- temperament or disposition; "a person of hot blood"*
*4. rake, rakehell, profligate, rip, **blood**, roue -- a dissolute man in fashionable society*
*5. **blood** -- people viewed as members of a group; "we need more young blood in this organization"*

In the medical domain the first, literal, sense of the word is chosen (*fluid that is pumped by the heart*) and assigned to the concept.

Case 3: The concept is not found in WordNet. This usually means that the concept is too specialized and the probability is high that we are dealing with a compound. By applying percolative rules on endocentric compounds we determine the head of the compound, for which again the definitions are determined. Example: the concept *hemodialysis session* is too specialized and hence has no match in WordNet. However it is an endocentric compound with the head *session* and the modifier *hemodialysis.* A search for *session* returns a word sense list as described in case 2:
*1. **session** -- a meeting for execution of a group's functions; "it was the opening session of the legislature"*
*2. school term, academic term, academic session, **session** -- the time during which a school holds classes; "they had to shorten the school term"*
*3. **session** -- a meeting devoted to a particular activity; "a filming session"; "a gossip session"*
*4. seance, sitting, **session** -- a meeting of spiritualists; "the seance was held in the medium's parlor"*
The sense 3 (*meeting devoted to a particular activity*) is selected. Regarding the modifier, one word sense is returned as described in case 1:
***hemodialysis**, haemodialysis -- dialysis of the blood to remove toxic substances or metabolic wastes from the bloodstream; used in the case of kidney failure*
The definition of hemodialysis session is thus constructed from the definition of its parts:
***hemodialysis session** -- a meeting devoted to dialysis of the blood to remove toxic substances or metabolic wastes from the bloodstream; used in the case of kidney failure;*

Case 4: Although the concept is found in WordNet, the description is considered too specialized by a domain expert and therefore inadequate. In this case the chosen description is either manually adapted or the hypernym definition is automatically determined through WordNet querying of its hypernym's concept definition. Example: the concept *hemodialysis* returns the definition
***hemodialysis**, haemodialysis -- dialysis of the blood to remove toxic substances or metabolic wastes from the bloodstream; used in the case*

of kidney failure
Since it is considered too specialized the following hypernym definition is established:
dialysis *-- separation of substances in solution by means of their unequal diffusion through semipermeable membranes*

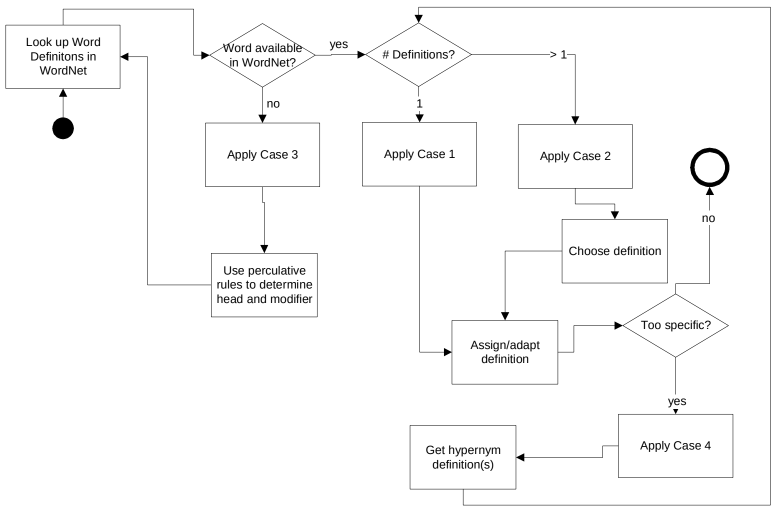

Fig. 3. Word Sense identification based on WordNet

The process (summarized in figure 3) is a guided way of fine-tuning ontologies not according to the quantity but the quality of concepts by adding semantics in order to make them easier understandable for non-experts and facilitate reuse. Using a general lexicon like WordNet allows the standardization of definitions. A prototype implementation is available that utilizes Perl for WordNet querying and allows among other things the listing of available word senses in WordNet, the determination of hypernym definitions and the adaption of definitions where required. Furthermore word sense identification is a bidirectional process as gaps in WordNet (see case 3 above) can be identified and filled. The process above is not limited to WordNet: every lexicon providing definitions can be utilized. More comprehensive lexicons are preferable, for this reason WordNet is a good default choice.

6 SUMMARY

In the requirements engineering domain fine granulated ontologies are necessary for efficient generation of models that can be further used in the application engineering steps. In this paper we proposed a step by step strategy of ontology engineering emanating from manually produced or already statistically filtered text.

Our approach focuses on the diversification of standard tags for optimizing the automatic elicitation of classes, relations and attributes in domain ontologies. Doing this with free text input can only be successful, if certain NLP standard techniques like probabilistic tagging get combined with special procedures like filtering, tag-enriching and chunking. The involved procedures are heuristically founded and follow a multilevel chunking strategy. We described a framework for mapping automatically generated linguistic categories to ontology concepts. Beyond that we showed in detail how these concepts can be refined and therefore optimized based on WordNet, in order to ensure their shareability. Our arguments are supported by a tool set that was developed in our research group for linguistically enhanced requirements engineering The output graph of our example (see chapter 4) proves that creating ontology fragments with linguistic fine-tuning is suitable in the context of requirements engineering.

REFERENCES

1. Fliedl, G., Kop, C., Vöhringer, J.: From OWL Class and Property Labels to Human Understandable Natural Language. In: Lecture Notes in Computer Science, 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, Paris, France (2007)
2. The QTAG POS-Tagger http://morphix-nlp.berlios.de/manual/node17.html
3. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods – In Language Processing (1994)
4. Brill, E.: Unsupervised learning of disambiguation rules for part of speech tagging. In Proceedings of the 3rd Workshop on Very Large Corpora, pages 1–13. (1995)
5. Goldwater, S., Griffiths, T.: A fully Bayesian approach to unsupervised part-of-speech tagging. In Proceedings of ACL (2007)

6. Fliedl G., Kop Ch., Vöhringer J., Winkler Ch.: NIBA Project: Overview. In: ER 2005 24th International Conference on Conceptual Modeling. Alpen-Adria-Universität Klagenfurt (2005)
7. Kop, C., Mayr, H.C. and others.: Tool Supported Extraction of Behavior Models. In: Information Systems Technology and ist Applications, ISTA' 2005 4th. International Conference, 23. - 25. May 2005, Palmerston North, New Zealand. GI 2005, 114-123, (2005)
8. Montylingua :a free, commonsense-enriched natural language understander, http://web.media.mit.edu/~hugo/montylingua
9. Monty Klu Web v0.1, http://montyklu.knospi.com
10. OpenNLP project website: http://opennlp.sourceforge.net/
11. Bird. S., Klein, E., Loper, E.: Natural Language Processing in Python, 2008, http://nltk.org/doc/en/book.pdf
12. Holger Knublauch, Ray W. Fergerson, Natalya F. Noy and Mark A. Musen: The Protege OWL Plugin: An Open Development Environment for Semantic Web Applications
13. Haase P., Lewen H., Studer R., Tran T. Erdmann M. d'Aquin M., Motta E.: The NeOn Ontology Engineering Toolkit. In WWW 2008 Developers Track. April 2008.
14. McGuinness, D. L.. "Conceptual Modeling for Distributed Ontology Environments." In Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000). Darmstadt, Germany. August 14-18, 2000.
15. OWL Web Ontology Language Overview , Deborah L. McGuinness and Frank van Harmelen, Editors. W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-features-20040210/. Latest version available at http://www.w3.org/TR/owl-features/.
16. The Resource Description Framework (RDF): http://www.w3.org/TR/rdf-concepts/
17. Mustafa Jarrar and Robert Meersman (2008). "Ontology Engineering -The DOGMA Approach". Book Chapter (Chapter 3). In Advances in Web Semantics I. Volume LNCS 4891, Springer.
18. SNOMED Clinical Terms: Overview of the Development Process and Project Status: Michael Q. Stearns, MD', Colin Price, MPhil, FRCS, Kent A. Spackman, MD, PhD" , Amy Y. Wang, MD'
19. The Unified Medical Language System: an informatics research collaboration.: Humphreys BL, Lindberg DA, Schoolman HM, Barnett GO.
20. Tom Gruber (1993). "A Translation Approach to Portable Ontology Specifications". In: *Knowledge Acquisitions* 5, (May): 199-220.
21. Natalya F. Noy and Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Stanford University,
22. Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho (2004). *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web*. Springer, 2004.

23. Fliedl G., Kop Ch., Mayr H. C., Hölbling M., Horn Th., Weber G., Winkler Ch.:
Extended Tagging and Interpretation Tools for Mapping
24. Black, C.A.: A step-by-step introduction to the Government and Binding theory of a syntax. In: Notes on Linguistics 2 (1996), Mai, Nr. 73
Weber, G.: NIBA<tag> Aspekte der Implementierung eines erweiterten Taggers für die automatische Textannotation in NIBA. Master Thesis. Alpen-Adria-Universität Klagenfurt (2007)
25. Mayr H. C., Kop C.: A User Centered Approach to Requirements Modeling. In Proceedings of Modellierung 2002, Köllen Verlag, Bonn 2002, pp. 75 – 86.
26. Gruber T.: A translation approach to portable ontologies. Knowledge Acquisition, 5(2), pp. 199-220, 1993
27. Jarrar M.: Towards the Notion of Gloss, and the Adoption of Linguistic Resources in Formal Ontology Engineering Proceedings of the 15th International World Wide Web Conference, Edinburgh, Scotland, pp. 497-503, ACM Press, May 2006
28. Lau R. Y. K., Li Y., Xu Y.: Mining Fuzzy Domain Ontology from Textual Databases, In: WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Publisher: IEEE Computer Society, November 2007
29. Perkonigg M.: Linguistische Aspekte des Attempto Controlled English (ACE). Masterarbeit, Alpen-Adria-Universität Klagenfurt, 2009.
30. Giuglea A.-M., Moschitti A.: Semantic role labeling via FrameNet, VerbNet and PropBank In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics; Publisher: Association for Computational Linguistics; July 2006

JÜRGEN VÖHRINGER
INSTITUTE OF APPLIED INFORMATICS,
RESEARCH GROUP APPLICATION ENGINEERING,
ALPEN-ADRIA-UNIVERSITÄT KLAGENFURT

DORIS GÄLLE
INSTITUTE OF APPLIED INFORMATICS,
RESEARCH GROUP APPLICATION ENGINEERING,
ALPEN-ADRIA-UNIVERSITÄT KLAGENFURT

GÜNTHER FLIEDL
INSTITUTE OF APPLIED INFORMATICS,
RESEARCH GROUP APPLICATION ENGINEERING,
ALPEN-ADRIA-UNIVERSITÄT KLAGENFURT

CHRISTIAN KOP
INSTITUTE OF APPLIED INFORMATICS,
RESEARCH GROUP APPLICATION ENGINEERING,
ALPEN-ADRIA-UNIVERSITÄT KLAGENFURT

MYKOLA BAZHENOV
COMPUTER AIDED MANAGEMENT SYSTEMS DEPARTMENT,
NATIONAL TECHNICAL UNIVERSITY
"KHARKOV POLYTECHNIC INSTITUTE"