Organizing Knowledge as an Ontology of the Domain of Resilient Computing by Means of Natural Language Processing

— An Experience Report —

Algirdas Avižienis[†], **Gintarė Grigonytė**[†‡], **Johann Haller**[‡], **Friedrich von Henke**[⋆], **Thorsten Liebig**[⋆] and **Olaf Noppens**[⋆]

[†]Faculty of Informatics, Vytautas Magnus University, Kaunas, Lithuania
[‡]Faculty of Humanities, Saarland University, Germany
[⋆]Faculty of Engineering and Computer Science, Ulm University, Germany

Abstract

Scientists typically need to take a large volume of information into account in order to deal with re-occurring tasks such as inspecting proceedings, finding related work, or reviewing papers. Our work aims at filling the gap between text documents and a structured representations of their content in the domain of resilient computing by combining computer linguistics and ontological methods. The results of our research include: a thesaurus of the domain, automatic clustering of the domain documents, a domain ontology, and a tool for constructing ontologies with the aid of domain thesauri.

Introduction

Documents containing scientific and technical knowledge are being generated and stored at a rapidly increasing rate. That is especially true in the field of informatics (computer science and engineering) that has been undergoing explosive growth in the past half century. The finding of specific knowledge about some aspect of informatics is made difficult by two factors:

(1) the absence of a structured representation (an ontology) of the fundamental concepts of the field;
(2) the existence of significantly different terminologies that describe synonymous or near-synonymous concepts of informatics.

Regarding (1), the only existing and widely used taxonomy that could be used to build an ontology is the *ACM Computing Classification System* (CCS)[1]. The CCS was created in 1988 and was last revised in 1998. It has fallen far behind the evolution of informatics and information technology.

As an illustration of (2), we have the concepts *resilience*, *dependability*, *trustworthiness*, *survivability*, *high confidence*, *high assurance*, *robustness*, *self-healing*, whose definitions appear to be identical or to overlap extensively. In many cases the definitions themselves have multiple versions that depend on a given author's preference. In this paper we describe our research effort that addresses the solution of the above problems for the domain of resilient computing that is the topic of the European Network of Excellence ReSIST[2] (Resilience for Survivability in Information Society Technologies). The research employs natural language processing techniques and tools as well as knowledge representation methods to accomplish two goals:

(1) to create a thesaurus and an ontology of resilient computing;
(2) to conduct automatic clustering experiments to organize the documents from the resilient computing domain.

Approach and Initial Resources

The basic architecture of our approach consists of a corpus, a thesaurus, an ontology, and a meaningful linking between both. The thesaurus is compiled from a representative corpus of the resilient computing literature. This thesaurus serves as the list of relevant terms in the chosen domain. The resilient computing ontology on the other hand contains a structured, expert-generated representation of pertinent concepts. These two representations are coupled by a bidirectional mapping between concepts and thesaurus terms. In order to establish this mapping we have developed a graphical mapping tool as plug-in to the ontology authoring environment OntoTrack (Liebig and Noppens 2005). An overview of our architecture is given in Figure 1.

Furthermore, we have implemented a clustering algorithm which groups similar documents as regards their thesaurus terms (depicted as groups of documents on the left hand side of Figure 1). Such a cluster refers to a cloud of thesaurus terms which, in turn, links to several concepts in the ontology. Since the ontology covers many different aspects of resilient computing (from different kinds of failures to attributes as well as methods to prevent or remove faults) this linkage can be understood as a "footprint" providing a condensed description of the document(s) content. For instance, a cluster of documents which try to increase a certain attribute of secure systems (say availability) by ruling out some type of fault (e.g. software-faults) with the aid of applying a certain fault removal technique (such as testing or model-checking) will link into appropriate concepts of the ontology at a certain level of detail. Our hypothesis is that typical topics within sets of documents will create distinct clusters whose linkage to the ontology will tell something

[1] `http://www.acm.org/about/class/ccs98-html`

[2] `http://www.resist-noe.net/`

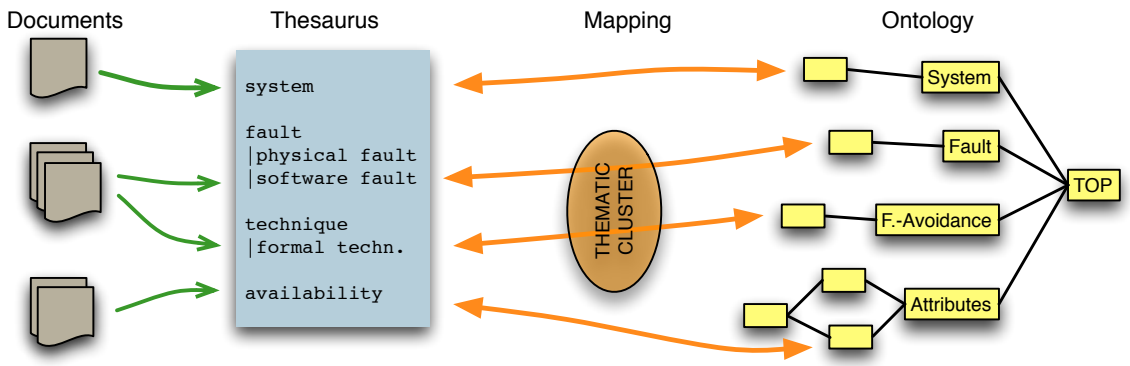

Figure 1: Conceptual architecture

about its content so that experts can name those footprints (called thematic clusters in Figure 1).

While document clustering, ontology creation, and mapping are about to be completed our current work deals with composing a complete chain of tools in order to classify existing resilient computing literature. Subsequent work will then deal with the approach of thematic clusters.

The corpus of text used in this research is composed of about 2500 papers presented at the 29 annual International Symposia on Fault-Tolerant Computing (1971-1999) and at their successors, the 7 International Conferences on Dependable Systems and Networks (2000-2006). Since the abstracts of the articles carry the essence of information, only abstracts of articles were used to compile the corpus. There are 234.585 running words in the corpus.

A starting point for building the Resilience ontology is the taxonomy presented in the so called ALRL paper (Avižienis et al. 2004), from which an ontology has been built that is expressed in the OWL language[3] and contains around 180 concepts. This ontology needs to be augmented with terms that were not included in (Avižienis et al. 2004) because of space limitations and with synonymous terms from other sets of terminologies currently in use.

The next section describes the approaches used for thesaurus and cluster creation. After that we report about the ontology building process and the mapping tool before ending with a conclusion and outlook.

Thesaurus Creation

The thesaurus of the domain serves two purposes in our research:

1. It provides a testimony for the outcome of the research. The list of important single- and multiword terms were expert reviewed and arranged according to their hypernym-hyponym relationships. The terms represent the domain of resilience in respect to the whole lifespan of the domain. The thesaurus is intended to serve as a point of reference - a unified list of terms in the domain of resilience.

[3] http://www.w3.org/2004/OWL/

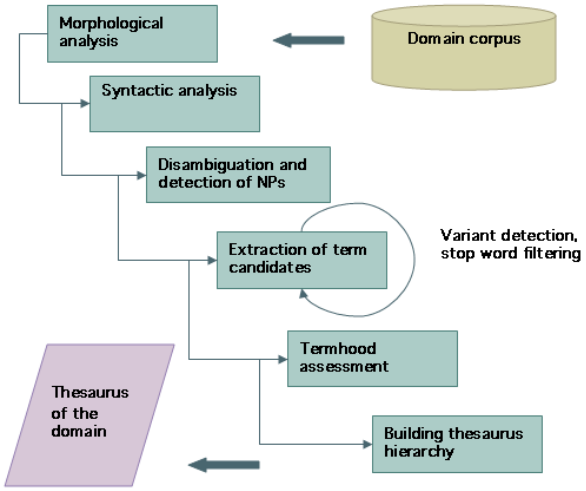

Figure 2: The process of building thesaurus

2. As an integrated component of our document processing architecture it will act as an meaningful index needed to establish intelligent document retrieval tasks.

We will describe the methodology for automatic domain thesaurus creation in this chapter. Our approach is based on linguistic pattern matching for automatic terminology extraction and IDF (Inverse Document Frequency) measurement for termhood assessment of terms. The automated process of building the thesaurus is depicted in Figure 2.

Rule based morphological analysis. The process of thesaurus learning starts with linguistic analysis of document abstracts set. For each word in the text, the MPRO system (Maas 1996) delivers information such as lemma, part of speech, derivation, semantic class etc. For instance, word *programming* is analyzed:

{string = programming, c = adj, vtyp = ing, more: {nb= sg, case= nom}, s = program#ing, ls = program, sem = derivationalAdj; activity}

Rule based disambiguation and syntactic analysis. Once we have morphologically annotated text, grammar

rules for morphological disambiguation and syntactic parsing can be applied. We use KURD (Carl and Schmidt-Wigger 1998) - a formalism that interprets rules based on finite-state technology. An example rule for identifying NP:

noun_phrase: IF *node{c=adj} AND -node{c=noun} THEN pattern {c=np}.

Detection of NPs. Morphological and statistical analysis is followed by the tagging of acronyms, proper names, possible single word terms and noun phrases:

Based on the <style code=acronym>**VFF** </style> approach, an <style code=simpl> **approach**</style> to find the <style <code=np>**optimal number**</style>[...]

Variant and non-basic term form detection. We have addressed variant issue due to a detailed morphological analysis, i.e., words that have the same morphemes can be easily detected and decision which form to use can be taken.

`fault-tolerant design`,
`fault tolerant design`

Stop words filtering. Applying a stop word (i.e. commonly used word, such as 'a') list filtering is a common practice in the terminology extraction field. In order to assure that only relevant NPs will be extracted, we have used a stop word list (i.e. words like `less, never, next, etc`).

Candidate term extraction. Combining rich morphological and syntactical analysis with pattern matching techniques of AUTOTERM (Haller 2006), (Hong, Fissaha, and Haller 2001) grammar allowed us to extract a wide span of entities:

Possible Terms: software fault; redundant system;
Toponyms: England;
Acronyms: SCHEME;
Names of Persons and Organizations: Jack Goldberg; N. Levitt; John H. Wensley Computer Science Group;

Termhood assessment. We consider two requirements: first, a term should not be too general, i.e., a term occurring in a document has to be a reliable indicator for what topic the article is about; and second a term should not be too specialized, i.e. such terms that only occur once and about whose status we therefore cannot be sure. To check whether these two criteria are met, IDF measure (1) – a measure of the general importance of the term - is used. IDF is obtained by dividing the number of all documents by the number of documents containing the term:

$$idf(t) = log(\frac{|D|}{(d : t \in d)}) \quad (1)$$

The candidate term extraction step has resulted in 6818 terms. Evaluation of the system (a sample of 10% of the abstracts) showed 82% of recall (which would be 18% of silence in the term extraction field) and 67% of precision (noise=33%). After the IDF values were obtained, and a threshold had been chosen by domain experts, the term list was pruned down to 5,710. Precision has increased up to 79% (noise decreased to 21%).

Hierarchical representation building. Extracted terms are represented via a hypernym-hyponym relationship. To create a hierarchy from general to more special terms we used a simple method: non-compound terms are top level hierarchy nodes; for a term t_x with n compound parts, we look up whether there is a term t_y consisting of the n-1 rightmost term parts; if so, the term t_x becomes a subterm of t_y.

```
fault
|bridge fault
|design fault
||latent design fault
||residual design fault
```

Clustering

Document clustering is another key component of the framework described earlier. Clustering is a quick and effective way for organizing documents (Mitchell 1997). It does not require expert knowledge and time. But one disadvantage of clustering has to be addressed if we want to use clustering as reliable method for organizing the documents – how to interpret the cluster?

In general clusters are represented by the features by which the objects of the clustering got assigned to one cluster. In our case having clusters of the documents gives us a list of features by which one cluster of documents can be discriminated from another cluster. The very important thing is what we choose the features of the documents to be. The ideal way would to have an expert describing a document. This, however, is not possible. Instead we have decided to use the AUTINDEX tool (Haller and Schmidt 2006) for automatically assigning features to each document.

For a given document *A*, the tool assigns a list of features *l(l1, l2, ..., lk)* taken from thesaurus *T* that is the thesaurus of the domain which we described in the previous section. *A* is a single document from the domain corpus described above. The list of the features is a list of terms taken from the domain thesaurus. Based on these features, similarities between the documents can be calculated. One of the standard measures is correlation; the correlation measure is based on covariance. While covariance gives us direction of relations between two vectors, it tells us nothing about its strength. Correlation normalizes results to a scale from +1 (perfect match) to -1 (perfect contradiction).

Our approach to document clustering is combining the hierarchical clustering algorithm described in (Manning and Schütze 1999) and the method of correlation clustering described in (Gael and Zhu 2007). In addition to this we use linguistic intelligence to build the feature representation of the data we will be clustering. AUTINDEX delivers a list of descriptors and their weight (importance) for each document. This information is used for building feature vectors.

The process of clustering

1. Every document is represented by a vector which contains descriptors and their weights. For example:
   (a) = (computer system[100], microprocessor[0], network[20], operating system[35], system message[0])
   (b) = (computer system[0], microprocessor[45], network[100], operating system[0], system message[56])
2. Similarities between all documents are calculated

3. The correlations between one document and all other documents are calculated
4. The hierarchical clustering algorithm is applied

Using the described methods the documents of the corpus were clustered into 345 clusters. Each cluster is represented by a cloud of terms taken from the resilient computing thesaurus.

Resilience Ontology

An ontology is a formal representation of a set of relevant concepts as well as relationships of a domain. The focused domain of this work refers to the main concepts with respect to resilient computing. The basic terms of this research field are characterized in (Avižienis et al. 2004) that provides in depth descriptions and classifications of threats, means, and attributes mostly on a textual level. This widely accepted scheme is an excellent blueprint for building an ontological representation of this domain. A version of such an ontological representation was provided by Brian Randell of Newcastle university in the early stages of this research.

Our own analysis of that part of the ontology dealing with the various types of faults revealed that this hierarchy contained almost no multiple inheritance, i. e. that the sub-fault relationship spanned a tree rather than a graph. In contrast, the categorization of faults in (Avižienis et al. 2004) accounts for eight basic viewpoints which lead to various overlapping groupings. Figure 3 shows the eight viewpoints on the left whose possible combinations lead to 31 fault classes (bottom row).

A more detailed investigation of the distribution of potential faults with respect to their viewpoints showed that this table implicitly encodes several subset, that is sub-fault, relationships. For instance, all *development faults* (upmost row) are also *internal faults* (third row) since the former is a subset of the latter. Furthermore, all *external faults* are *operational faults* (see Figure 3 for an illustration of these two examples). Altogether we were able to identify 11 sub-fault relationships and two fault equivalence relationships from this figure of fault categories. The resulting fraction of the fault hierarchy is shown in Figure 4. The arrows represent the sub-concept relationships and semantically equivalent faults are drawn within a surrounding box.

While the resulting hierarchy of faults may look obvious to domain experts, it is important to remember that the first sketch missed some of the sub-fault relationships of the describing source paper. Since every knowledge-aware processing method can only take explicitly modeled (or implicit but entailed) facts into account it is important to represent even the supposedly obvious.

We are currently in the process of restructuring the other parts of the resilience ontology with analogous methods.

Mapping Thesaurus and Ontology

To sum up, we are facing two different approaches for organizing knowledge: on one hand, the resilience ontology has been carefully constructed and re-structured after revision. It is mainly handcrafted by, or with help of, domain

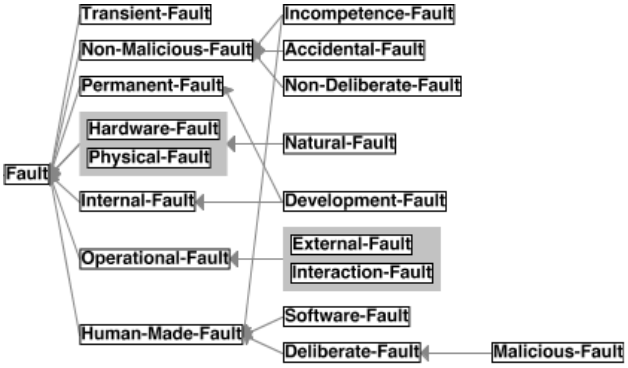

Figure 4: Revised fault hierarchy

experts and is consistent wrt. the underlying logical formalism. On the other hand, the thesaurus has been primarily automatically compiled from various documents in an unsupervised manner. More precisely, the thesaurus not only contains redundant but also irrelevant (wrt. resilience domain) and rarely used – or even mis-used – terms. In our approach we need to bridge the result of both approaches.

In terms of numbers, about 6000 thesaurus terms are facing about 180 well-defined ontology concepts. Here, tool support for the mapping of thesaurus and ontology is recommended. Our tool supports the user semi-automatically and follows the observation that only a small set of relevant terms need to be mapped: the primary structure of the thesaurus as well as the ontology is the hypernym-hyponym (or sub-set) relationship. Depending on the relevance of terms wrt. the resilience domain one can choose different levels of granularity for the mapping: branch versus leaf mapping. Leaf mapping means to map single terms to concepts in the ontology. Utilizing branch mapping one maps a more general term to a concept. Due to the sub-set relationship all sub-terms are also mapped to the given concept. For instance, the terms related to `faults` are key concepts in the resilient computing domain and therefore are typically mapped one-by-one. However, other terms such as *algorithmic circuit verification, transactional rollback, online fault diagnosis* can only be mapped via their hypernym *i.e. verification, rollback, diagnosis*.

We identified the following 4 mapping tasks that are supported by our mapping tool:

(1) Creating term – concept links. By mapping a term to a concept (or, respectively, a concept to a term) we establish a link from a specific thesaurus term to an ontology concept (or vice versa). This means that the given term and concept are semantically equivalent wrt. the resilience domain. The link does not necessarily form a one-to-one relationship: the same term can be linked to several concepts.

Our tool utilizes simple NLP techniques (e. g. entity matching, hyphen recognition) as well as more advanced ones (e. g. variant detection) in order to support the user in semi-automatically establishing an initial mapping between terms and concepts. By dragging concepts from the ontology view of our application (see right-hand side of Figure 5)

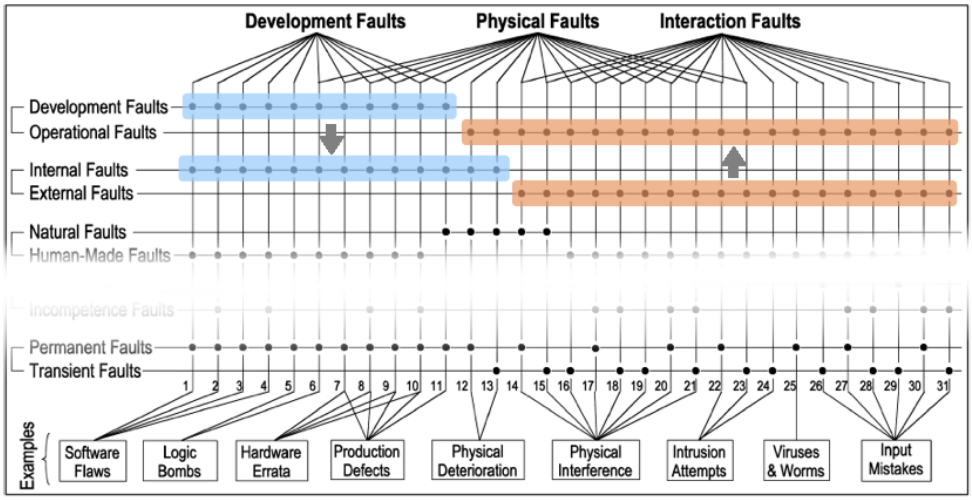

Figure 3: Fault categories as of Fig. 5a (Avižienis et al. 2004)

to the hierarchy of terms (left-hand side) user-defined links can easily be established. Note that using Tablet PCs with pen-like or similar devices simplifies this work.

(2) Introducing equivalence between terms. The current thesaurus creation process does not consider synonymy issues. Synonyms are not detected and marked in the introduced hierarchy of terms. Note that not all synonyms can be automatically found during NLP: here we have to distinguish well-known synonyms in the field of resilient computing and rare – or even incorrectly used – synonyms only introduced in some of the analyzed documents. Knowing which thesaurus terms are synonymous would improve the structure of the thesaurus and improve the indexing of the domain documents, and therefore – the clustering of the domain documents.

Following the simple "drag-'n-drop" approach, equivalence can be easily introduced by dragging terms to terms.

(3) Adding terms to the ontology. To improve the quality of the initial ontology it should be possible to enrich the ontology with relevant terms automatically extracted from the documents. However, it is very important not to blindly add any term but to pick the most relevant ones as well as only commonly used ones.

Again, mapping is done via "drag-'n-drop" operations. Adding terms as concepts to the ontology means either adding the term as a sub-concept or an equivalent concept to an existing one. Moreover, whole sub-hierarchies of terms can be marked and mapped via one single operation that introduces the corresponding hierarchy into the ontology.

(4) Discarding non-relevant terms. Revising the thesaurus terms by discarding non-relevant terms wrt. the resilience domain is one of the first steps to improve the quality of the thesaurus. However, the whole process cannot be automatically performed because it is not obvious which terms are relevant. For instance, some terms such as DBMS, C-library etc. are frequently used in the set of documents but do not specify any resilience-specific topic. Moreover, the semantics of some terms are not clear (e. g. combinations of faults) or do not refer to any term in the resilience domain. This can only be solved by an expert.

The mapping tool is implemented as a plugin for the ontology authoring and visualization framework OntoTrack (Liebig and Noppens 2005). Therefore we benefit from the various reasoning and checking capabilities to assist the user by automatically detecting possible problems. For instance, a mapping of a term to several concepts within the same hierarchy wrt. hypernym-hyponym relationship is redundant and can be simplified to a mapping to the most specific concept(s). Several working lists of not yet mapped as well as already mapped or removed terms can be presented to the user, and our implemented XML-based exchange format of the mappings enables us to assign mapping tasks to different experts and to integrate the results.

Discussion

Our approach combines methods from two fields, namely Computational Linguistics and Knowledge Representation, such that there is a benefit to both sides. On one hand, expert created knowledge within an ontology is used to categorize documents by a linkage from automatically extracted descriptors to ontology concepts. On the other hand, thesaurus terms gathered with the help of a chain of language processing tools can be used to enrich or refine an ontology of a particular domain.

In particular, the results from our work so far include:

- An *thesaurus of the domain*, which was constructed automatically from the corpus of the resilience domain. Terms are structured via the hyperonym-hyponym relationship.
- A *clustering of the domain documents*. The documents of the domain were automatically indexed with the terms of the domain thesaurus. Based on these features, 345 clusters have been identified. Each cluster is represented by its cloud of thesaurus words. So far we have used clusters only as a means of organizing domain texts. The initial idea about thematic clusters is work in progress.

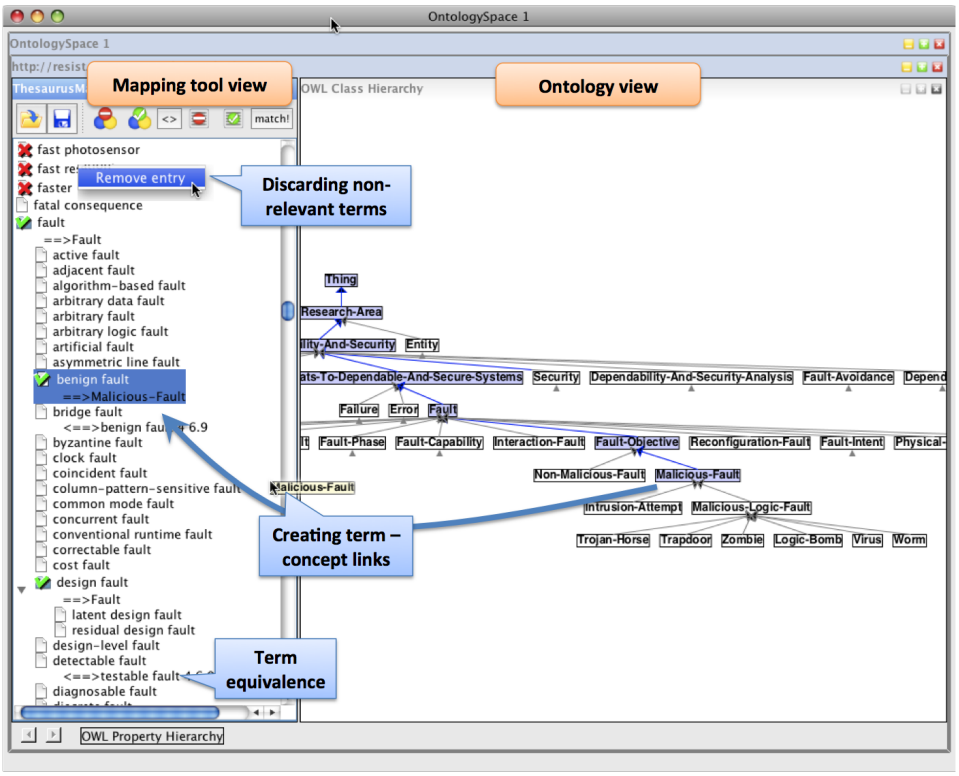

Figure 5: Mapping tool.

Means of clustering is one of possible ways for introducing more structure into a shallow thesaurus representation and therefore could be used for building ontologies.

- Large parts of an *domain ontology* which has been manually constructed by analyzing appropriate literature as well as from our thesaurus.
- A mapping which connects the ontology with the thesaurus as a base for future activities aiming at establishing an automatic document classification process.

The presented approach is domain independent and, since it deals with unstructured texts, is especially beneficial for domains that have no prior knowledge resources, i.e. glossaries, thesauri, organized bases of domain documents.

Acknowledgments

This research has been supported by EC Information Society Technologies contract no. 02764, Network of Excellence ReSIST (Resilience for Survivability in IST). The authors wish to thank their colleagues Mahmoud Gindyeh, Hugh Glaser, Jean-Claude Laprie, Rūta Marcinkevičienė, Ian Millard, and Brian Randell for their advice and assistance during various stages of this research.

References

Avižienis, A.; Laprie, J.-C.; Randell, B.; and Landwehr, C. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions of Dependable and Secure Computing* 1(1):11–33.

Carl, M., and Schmidt-Wigger, A. 1998. Shallow Post Morphological Processing with KURD. In *Proc. of the Conf. on New Methods in Language Processing (NeMLaP)*.

Gael, J. V., and Zhu, X. 2007. Correlation clustering for crosslingual link detection. In *Proc. of the Int. Joint Conf. on Artificial Itelligence (IJCAI'07)*, 1744–1749.

Haller, J., and Schmidt, P. 2006. AUTINDEX – Automatische Indexierung. *Zeitschrift für Bibliothekswesen und Bibliographie – Sonderheft 89* 104–114.

Haller, J. 2006. *Multiperspektivische Fragestellungen der Translation in der Romania*. Frankfurt: Peter Lang Verlag. chapter AUTOTERM - Automatische Terminologieextraktion Spanisch-Deutsch, 229–242.

Hong, M.; Fissaha, S.; and Haller, J. 2001. Hybrid Filtering for Extraction of Term Candidates from German Technical Texts. In *Proc. of the Int. Conf. on Terminology & Artificial Intelligence (TIA 2001)*, 223–232.

Liebig, T., and Noppens, O. 2005. OntoTrack: A semantic approach for ontology authoring. *Journal of Web Semantics* 3(2):116 – 131.

Maas, D. 1996. *Linguistische Verifikation, Sprache und Information*. Max Niemeyer Verlag. chapter MPRO - ein System zur Analyse und Synthese deutscher Wörter.

Manning, C., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press Cambridge.

Mitchell, T. 1997. *Machine Learning*. McGraw Hill Series in Computer Science. McGraw Hill.