ReSIST: Resilience for Survivability in IST

A European Network of Excellence

Contract Number: 026764

Deliverable D34: Resilience ontology: final

Report Preparation Date: December 2008
Classification: Public

Contract Start Date: 1st January 2006
Contract Duration: 39 months
Project Co-ordinator: LAAS-CNRS
Partners: Budapest University of Technology and Economics
City University, London
Technische Universität Darmstadt
Deep Blue Srl
Institut Eurécom
France Telecom Recherche et Développement
IBM Research GmbH
Université de Rennes 1 – IRISA
Université de Toulouse III – IRIT
Vytautas Magnus University, Kaunas
Fundação da Faculdade de Ciencias da Universidade de Lisboa
University of Newcastle upon Tyne
Università di Pisa
QinetiQ Limited
Università degli studi di Roma "La Sapienza"
Universität Ulm
University of Southampton

Table of contents

1. Introduction: computer-aided construction of a thesaurus and an ontology of resilience

The objective of this project is to create a structured representation of the concepts underlying the contents of the large and very rapidly increasing set of documents that represent knowledge in the technical domain of *resilience*.

The purpose of the representation, in the form of a thesaurus and an ontology, is to be able to use natural language processing tools to perform computer-aided identification and classification of existing documents concerned with resilience that have been generated from the time when the correctness of the results of computations became a concern of the first computer users in the 1940's until the present, and to classify new documents as they are generated.

Resilience in this discussion is defined as "the persistence of dependability in the presence of changes" (Laprie, 2008). In the remaining part of this section the term "dependability" is used as an abbreviation for the definition given above, that is, for "resilience".

1.1 The need: sometimes we don't know what we are talking about

Dependability has naturally concerned most disciplines of informatics (computer science and engineering) since the early days. As a consequence, significantly different terminologies were developed by different communities to describe the same aspects of dependability. The terminologies became entrenched through usage at annual conferences, in books, journals, research reports, standards, industrial handbooks and manuals, patents, etc.

As an illustration, we have the concepts of *resilience, dependability, trustworthiness, survivability, high confidence, high assurance, robustness, self-healing*, etc., 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.

An example of a long-term effort to create a framework of dependability concepts is the effort within the Technical Committee on Dependable Computing and Fault Tolerance of the IEEE Computer Society and the IFIP Working Group 10.4 that began with a special session at FTCS-12 in 1982. Since then it has resulted in a series of papers, a six-language book in 1992 (Laprie, 1992), and in 2004 the paper "Basic Concepts and Taxonomy of Dependable and Secure Computing" (Avizienis et al., 2004). To the extent of our knowledge no other technical domain of informatics has produced such a taxonomy.

The use of several synonyms or near-synonyms that lack well-defined distinctions is a source of continuing confusion that leads to re-inventions and plagiarism, impairs the transfer of research results to practical use and blocks the recognition of related documents.

The orderly progress of dependability research and its practical applications requires that past work as well as new results should be classified on the basis of a single ontology and thus made accessible to the entire profession. However, it is unreasonable to expect that a committee formed by the different communities could by volunteer effort create a taxonomy document from which a single consensus ontology could be generated.

It must be concluded that today the purely "intellectual" (i.e., human) process of ontology building for dependability concepts is reaching its limits. The complementary solution is to augment the human effort by the use of automatic natural language processing tools that have been developed by computer linguists. The next step must be computer-aided building of a consensus ontology.

1.2 The means and progress: natural language processing by computers can help

During the past decade much progress has been made in the development of computer tools for human language processing. Such tools have been developed for the extraction of term candidates from a corpus (set of texts). A *thesaurus* (list of important terms with related terms for each entry) is constructed from the candidates. The *ontology* for a given domain is a data model that represents those terms and their relationships. Automatic indexation of the texts is carried out using the thesaurus, followed by clustering analysis using statistical and linguistic techniques. A measure of similarity between texts is computed that serves as a basis for automatic classification.

The applicability of the above listed techniques to texts in the dependability domain has been investigated at the Center for Computer Linguistics of Vytautas Magnus University (VMU) in Kaunas, Lithuania, and at the Artificial Intelligence Institute of Ulm University in Germany. The study has used the tools developed at the Institute for Applied Information Research (IAI) of Saarland University in Germany, whose researchers are Affiliate members of ReSIST and have made significant contributions to this research. Valuable advice and support have been received from ReSIST partners LAAS, Newcastle, and Southampton.

The *corpus* of texts used in this investigation is composed of the texts of over 2800 papers presented at 29 FTCS and 9 DSN conferences (1971-2008) that are sponsored by the IEEE Computer Society and IFIP. The results of the research are presented in this report.

The encouraging results of the processing of texts from the FTCS/DSN community leads to the conjecture that similar processing of texts from other conferences, journals, books, industrial documents, etc., will produce other ontologies that can be merged into a consensus ontology that covers the entire discipline of resilience.

1.3 The obstacle: classification is a problem for the entire field of informatics (computer science and engineering)

A dependability ontology is an integral part of an ontology for all of informatics, or (in North American terminology) of computer science and engineering. Such an ontology does not exist at present. The only existing and widely used taxonomy that could be used to build it is the ACM Computing Classification System (CCS). The CCS was created in 1988 and was last revised in 1998. It has fallen far behind the evolution of informatics and information technology. The concepts of dependability are treated very inadequately, and many significant dependability terms are altogether missing in the 1998 ACM CCS taxonomy.

Most documents that deal with dependability refer to "dependability of X", where X is: hardware, software, system architecture, database, etc. These upper-level terms of the informatics ontology must be available when classifying dependability documents. The existing CCS is a severe handicap, but it must be used until a better one is available. At this time the ACM is initiating the next update of the CCS, with one goal being the development of a flexible incremental process of updating.

1.4 A grand challenge

The coming update of the CCS is a grand challenge to the dependability community: it must take part in the process of creating an up-to-date and evolvable version of the CCS that adequately incorporates dependability concepts. The new CCS would allow the computer-aided construction of a thesaurus and an ontology for the entire informatics profession. However, we must put our own house in order first: a consensus dependability ontology with explicit synonymy relations must be available to the CCS builders. The prize to be gained is also grand: a "researcher's assistant" (or "referee's helper") that uses the ontology to search

the immense collection of past publications for relevant references in the dependability domain.

1.5 An unsatisfactory alternative: the “info-skeptic’s” view

A different view of the informatics ontology problem is also possible: all information concepts, systems and theories are human-made, in contrast to natural phenomena that exist as given facts to be investigated by the physical sciences. Therefore the disappearance of a concept and its replacement by a synonym is simply a case of survival of the fittest: if the concept’s originators were not able to assure its survival, then someone else will rediscover and rename the concept in due time. Thus there is no need to keep track of the past, because the good stuff will reappear under a different name.

An illustration of the introduction of a synonym with the pretense of originality is the recent appearance of the term “self-healing” in informatics literature. The concept of “healing” implies either a physiological process of an organism getting well or a process of recovery through prayer or faith. However, the technical explanations of “self-healing” do not cover these aspects. They are similar to long-established techniques of self-repair and fault tolerance of computer software and hardware, and the claim of the term’s originality is not supported by the evidence in the publications.

1.6 Continuation of the research

The original Description of Work of ReSIST did not contain the separate “Resilience thesaurus and ontology” (IT-T3) task. The topic of “ontology engineering” was included in the RKB task. The need for a separate task became evident during the first year of the ReSIST NoE project, and the Task IT-T3 of Work Package WP1 was introduced in Workplan update D8.

The results of the effort during the next two years are presented in this report. It is evident that we have made a good start, but also that many interesting and significant questions remain to be answered. For this reason the participants from VMU Kaunas, Ulm, LAAS, Newcastle, Southampton, and IAI Saarbruecken have expressed a strong interest to continue co-operative resilience ontology research past the completion of the ReSIST contract.

The most likely framework for the continuation is the formation of a Special Interest Group on Ontologies as part of the IFIP WG 10.4 “Dependable Computing and Fault Tolerance” activity. Such a SIG Ontologies would provide world-wide IFIP sponsorship for continuation of the research. Support for the research could be sought from national research sponsors as well as from the EC and other international entities.

2. A research plan: organization and retrieval of knowledge in the resilience domain by means of computer-based natural language processing tools

This section presents a long-range research plan for the organization and retrieval of knowledge in the resilience domain. It is evident that our results will be applicable in all other domains of engineering and science.

The objective of our research is to use computer-based tools for natural language processing to the fullest extent possible. Human experts will participate where absolutely necessary. This research plan is intended to define research that will be continued after the conclusion of the ReSIST NoE project. It is our goal to do as much as time and resources allow during the remaining part of ReSIST and to continue without interruption after its conclusion.

The long-range goals of this research are:

(1) To fill the gap that exists between knowledge being created (documents being generated) in the domain of resilience and the structured representation of the content of those documents by using natural language processing tools to create a thesaurus and an ontology of *resilience* that is defined as: *the persistence of dependability in the presence of changes* (Laprie 2008).
(2) To integrate the natural language processing tools and to conduct automatic classification experiments with documents in the resilience domain in order to discover and classify the existing resilience literature and to advance the state-of-the-art in the automatic classification of technical literature.

We expect to generate the following results:

- a comprehensive collection of the terminology for the resilience domain in the form of a thesaurus;
- a resilience ontology for effective automatic information organization, classification, and retrieval;
- an integrated set of natural language processing tools for thesaurus and ontology building, automatic clustering analysis, and automatic document classification;
- a specification of an user-friendly interface which interactively supports the visual analysis and integration of terminologies, ontologies, and classification tools;
- a comprehensive roadmap for further research.

The research is divided into three tasks that are described below.

2.1 Task 1: ontology development

The starting point for future research is the experience and results gained in the pilot project that has been conducted within ReSIST and is described in this report. The corpus of the abstracts from the Compendium of 2830 papers presented at 38 FTCS and DSN conferences (1971-2008) was processed to extract about 8000 term candidates for the computer-aided construction of a thesaurus and ontology. The thesaurus was employed to do automatic indexing of the Compendium papers, followed by the identification of about 800 clusters by means of automatic clustering analysis.

The fact that the Compendium represents the terminology used by one of several research communities is a significant limitation of the generality of the thesaurus and ontology. Other potential sources of texts for term extraction and subsequent creation of a thesaurus and ontology of resilience are:

(1) Other long-term conferences and journals, such as SAFECOMP, Software Reliability Engineering, Survivability, HASE, EDCC, PRDC, Oakland and other security conferences, IEEE Transactions on Reliability, on Computers, on Dependable and Secure Computing, IBM R&D and Systems Journals, etc.
(2) Project documents: ReSIST and other EU projects, USA project documentation, etc.
(3) Industrial documents: standards, white papers, manuals and handbooks, product descriptions, etc.
(4) Patents related to resilience, dependability, security, etc.

In further research we will choose several of the most readily accessible document collections that are within our time and budget limits. We then proceed as follows:

1. We will apply existing tools for terminology extraction. The terms will be organized in a hierarchical thesaurus with upper nodes and lower level nodes.

2. The thesaurus will be used to index the documents. The indexing results are the basis for assessing how closely any pair of documents are related; this again is used as basis for clustering methods.

Comparison of the results of two or more efforts will serve to validate the results. As the final step the resulting final ontology will be compared with the ALRL ontology. This also includes the discussion of the formal ontology's expressiveness and suitability of its structure for modeling the resilience domain. Furthermore, we plan to adapt existing user-friendly semi-automatical ontology development tools to support both the overall task by visually analyzing and validating the resulting ontologies, and the process of comparing and merging ontologies.

2.2 Task 2: automatic classification of documents and retrieval of resilience information

At the present time there exists a large number of documents that contain resilience-related information but are not identified by keywords or classified according to the existing ACM Computing Classification System (CCS). The CCS itself was last revised in 1998 and has fallen far behind the evolution of information technology. Concepts of resilience, dependability, and security are treated very inadequately**,** and many significant terms are altogether missing in the CCS taxonomy.
The results of the first task will be used in the following ways:

1. The ontology is to be enriched by experts using the automatically acquired thesaurus, and the thesaurus can be enhanced with metadata which connects terms to nodes in the ontology.
2. The documents indexed with the thesaurus can thus be enriched with metadata referring to the ontology.
3. Results from point 2 again allow us to conduct experiments in automatic classification of documents both without and with training data. The comparison of the results will show the relative effectiveness of the approaches.

Once the three steps have been covered, we will have a document collection which will have been tagged and classified with respect to the existing ALRL ontology and newly acquired lower level ontology nodes. The result of this task will be made available to the research community through the RKB.

2.3 Task 3: roadmap for the future

Throughout the execution of the next stage of this research we will collect ideas about additional research that could follow the present work and compile them into a roadmap that will serve as the basis for further research proposals. The roadmap will also document the difficulties that were encountered during this effort and propose how they could be overcome.

3. Domain independent automatic term extraction

3.1 Introduction

Terminology extraction plays an important role in building lexical resources and is currently applied widely in IE, IR, ontologies and Knowledge Base building fields. From a NLP perspective, there are several approaches for terminology extraction: linguistic, statistic and hybrid. Terminology extraction systems based on linguistic approaches have a higher than 70% coverage in term extraction (Bennet, 1999, Bourigault, 2001). Statistical term extraction approaches, when given a big annotated training corpus, can perform almost as well, but these

methods do not always guarantee the integrity of the term (Frantzi, 1999, Jisong Chen, 2006). The practice however shows that linguistic approaches (i.e., rule based) outperform statistical ones in regards to precision, although one can get better results by combining both linguistic and statistical approaches in various stages of term extraction (Schiller, 1996, Bourigault, 2001).

In the process of term extraction (Nagakawa, 2001) a step entailing "recognition of all NPs" (or so called extraction of term candidates) is generally considered as default. Since domain language is a more specific subset of a general language, rule based language processing tools can be applied for terminology extraction in any domain. The important question is how to distinguish between domain specific terms and general NPs. Nagakawa (2001) notes, that in order to extract domain specific terms from term candidates, a ranking of term candidates according to their termhood is necessary. A term's informational value (*termhood*) can be captured by statistical methods (IDF, MI, log likelihood, entropy, etc.).

Our approach to term extraction and building structured lexical resources is based on linguistic pattern matching and using IDF measurement for term quality assurance. It is in a way similar to approaches presented by Bourigault (1992), Daille (1994), and Paulo (2002).

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.
2. It is a component of the framework as described in chapter 1. The thesaurus of the domain, was used for indexing domain corpus for IR tasks. We will describe the methodology for automatic domain thesaurus creation in this chapter. Our approach to building the thesaurus is based on linguistic pattern matching for automatic terminology extraction and IDF measurement for termhood assessment of terms. All the steps are automated. The process of building a thesaurus is depicted in Figure 1.

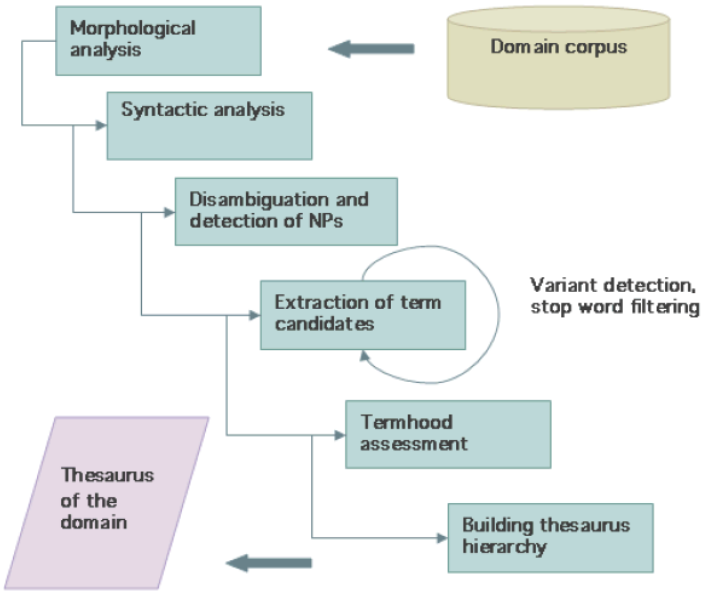

Figure 1: The process of building a thesaurus

3.2 The corpus

The corpus of text used in this research is composed of the 2830 abstracts of research papers presented at:

1) the 29 annual International Symposia on Fault-Tolerant Computing (FTCS 1971-1999),
2) and at their successors, the 9 International Conferences on Dependable Systems and Networks (DSN 2000-2008).

The text files were created from PDFs using the free *pdftotext* tool delivered with the Unix program XPDF. The corpus contains 234,585 tokens.

3.3 The methodology

Rule based morphological analysis. The process of thesaurus learning starts with the 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, the word *programming* is analyzed as follows:

{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. The following is an example rule for identifying a noun phrase NP:

noun phrase: IF *node {c=adj} AND –node{c=noun} THEN pattern {c=np}.

Detection of NPs. Morphological and statistical analyses are 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 the variant issue due to a detailed morphological analysis, i.e., words that have the same morphemes can be easily detected and the decision about which form to use can be taken, for instance:

```
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 analyses with the pattern matching techniques of AUTOTERM (Haller 2006), (Hong, Fissaha, and Haller 2001) grammar has 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., 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, the *IDF measure* - 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:

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

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 tx with n compound parts, we look up whether there is a term ty consisting of the n-1 rightmost term parts; if so, the term tx becomes a subterm of ty.

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

3.4 The experiment

Text resources used in the experiment cover 2830 abstracts of papers in the domains of Dependability and Security. Processing through the methodology described in chapter 3.3, we gained 9012 terms. After the informational values were obtained, and we had defined a certain threshold, the term list was pruned down to 7974. All the steps were fully automated.
The manual annotation of the system is described in section 4.

4. The manual refinement of the thesaurus of resilience

For the evaluation of the resilience thesaurus, we have created the annotation system (Figure 2), which is used by ReSIST experts.
The system contains the list of about 8000 *term candidates* that were extracted by fully automatic methods from the abstracts of all papers published in the Proceedings of FTCS and DSN conferences from 1971 to 2008. The list is organized in 80 pages.

Each page has one hundred term candidates (tc). They all are marked **General term** at the beginning and identified by page number and tc number on the page, for example, 1.23 is "tc 23 on page 1".
A page **must be saved** after classifying its tc's, otherwise the work will be lost. Returning to a page and changing it later is possible.
There are seven columns available to classify each term candidate. Going from right to left they are:
D&S (dependability and security): the tc is used in both domains.
Examples: diverse system (1.17), intrusion-tolerant system (1.23)
Security: the tc is usually used in the security domain only.
Examples: anomaly-based detection system (2.91), secure database system (2.94)
Dependability: the tc is usually used in the dependability domain only.
Examples: diagnosable system (1.13), repairable system (1.26)
Computer Science & Engineering: the tc is also used in other domains of CS&E.
Examples: asynchronous system (1.7), operating system (1.11)
Non-term: the tc cannot be recognized for classification and needs to be reviewed for its meaning.
Examples:LEDA system (1.75), faulty QDI system (2.74)
Vague term: the tc is either too general or too vague for use in a taxonomy or ontology.
Examples: tolerant system (1.5), reliable system (1.25), large-scale computer system (1.37), complete system (1.49), complex system (1.50), adversarial system (1.76), component-based system (1.83)
General term: the tc is also used in other domains of science and engineering outside of CS&E.
Examples: system (1.1), electronic system (1.19), autonomous system (1.29), cognitive system (1.48)
Comments:
1. For the construction of the thesaurus, the selection of dependability and security terms would be sufficient (the logic OR of the Dependability, Security, and D&S columns); the more detailed grouping is needed for further study of the term extraction and classification processes.
2. The "non-term" marking identifies tc's that need to be further considered for their meaning or those tc's that are malformed – for example, there are a few cases in which persons' names have been included in a tc.
3. The "vague term" category is well populated – here we ask for the classifier's judgment which tc's are not useful for classification because of lack of precision (large, small, simple, complex, etc.) or because of vagueness (tolerant, eternal, reliable, unreliable, etc.). This classification will allow the identification of "stop words" in further term extraction.

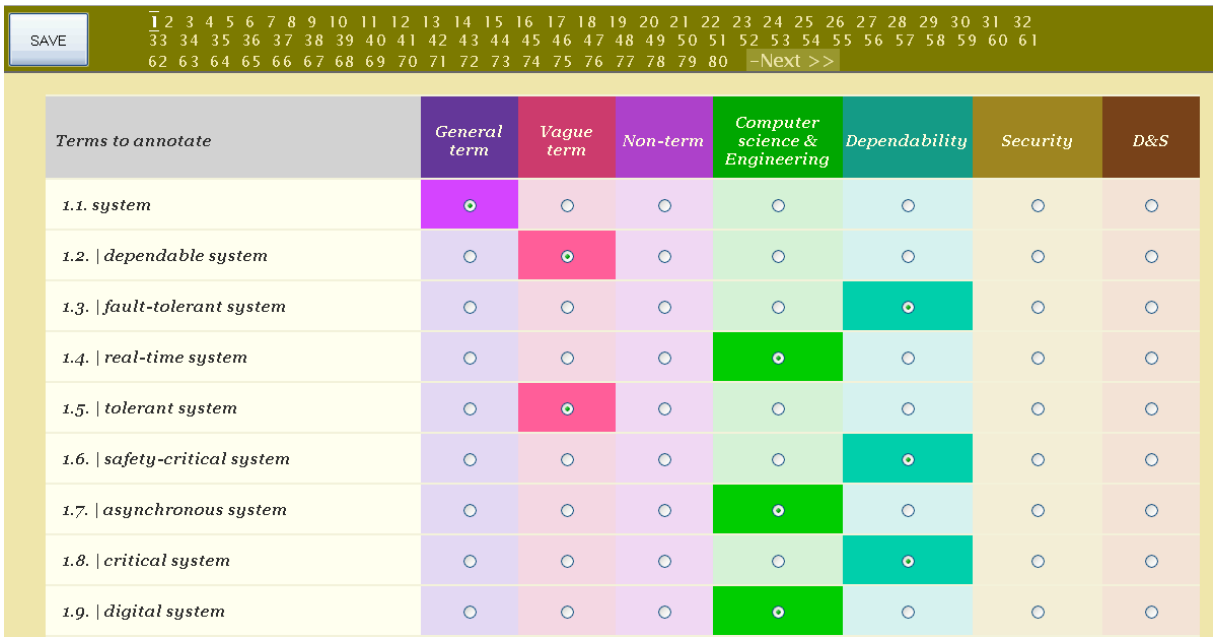

Figure 2: The term annotation system.

5. Organizing technical documents by means of clustering

Organizing documents and performing search is a common but not a trivial task in information systems. With the increasing number of documents, it is becoming crucial to automate these processes. Clustering is a solution for organizing large amount of documents. In this article we propose to improve the RKB[1] Explorer with the help of morphosyntactic analysis and adaptive hierarchical clustering method.

5.1 The motivation

The purpose of this experiment is to use clustering in order to organize documents in the RKB. Scientific publications are accumulated in the RKB where they can later be retrieved using simple keyword/pattern matching techniques. This type of technique to retrieve relevant documents is not very effective. The motivation behind this experiment is to improve the RKB for the following use case scenario:

The RKB interface presents interlinked items from their shared relationship they have through research activity. The information entities are chosen as research projects, research topics, researchers, and publications (Glaser, 2007). Users are able to traverse the datascape by altering selection topics and choosing search results. For instance, when searching for similar publications to ones already known, the user locates a known title and is presented with a list of linked publications. When performing this kind of search, the aim is to return only highly relevant results. User expects to find similar publications within the first 5-10 results. One possible solution for optimizing search results is document clustering (Kouomou, 2005). Clustering is a quick way to acquire relevant document sets. For a particular document, search results would be documents from the same cluster that the given document belongs to.

The quality of clusters is determined by the following criterion: the cluster shouldn't be too small or too large. Clusters are used for representing relevant information in the RKB.

[1] http://www.rkbexplorer.com/explorer/

5.2 The method

Our approach combines correlation values as similarity distance measures and applies a hierarchical clustering algorithm. To acquire distance measures we used numeric values of the importance of NPs in a particular document. For that we performed morphological and syntactical analyses of documents and used the general technical FIZ thesaurus[2] (our resilience thesaurus was not yet available) to calculate the importance of the NPs.

The clustering process can be divided into 4 general steps:

1. Identification of the NPs in the documents.
2. Creation of feature vectors (NPs and their weights) for each document.
3. Calculation of a similarity degree (1-pass correlation or 2-pass correlation), and population of the similarity matrix.
4. Applying clustering algorithm on the basis of the similarity matrix (iterative task).

5.3 The phases of the method

1. First, each document is linguistically analyzed. Tasks include a lemmatisation, a part of speech tagging, and a partial semantic tagging. We have used MPRO software (Carl, 2002) for that purpose. Consequently, the noun phrases NP are marked in each document.

2. As the next step, the importance of each noun phrase – a weight – is calculated. The NPs are weighted by means of the thesaurus (in our case, we have used the English version of the FIZ thesaurus[2]) and the results of linguistic analysis. The weight is calculated according to:
The NP's frequency in the document; the status of the NP in relation to the thesaurus: whether it is a hypernym or hyponym, or has no correspondence in the FIZ thesaurus; the number of semantic classes allocated to the particular NP during the linguistic analysis; the number of semantic classes allocated to the document; and the position of the NP in the document (beginning, end, etc.). A detailed description of the formula we have used is given in Haller (2006).
We use the NPs and their weights for forming feature vectors for each document. Subsequently, each document is represented as a vector in vector space R^N of whose elements are the NPs and their weights. For example, a document vector will appear as follows:

```
D= (ethyl [100]; research and development [87]; compression-
ignition engines [43]; classing [28]; engine performance [27];
project planning [21])
```

We assume the vector space $V = (V_1, V_2, \ldots, V_j, \ldots, V_N)$, where j_i is the j-th document (vector). Besides every vector (document) has different dimensions (depending on the number of NP representing the document).
We have the matrix of documents of Table 1, where columns V_j/D_j are vectors (documents) and rows NP_i - the NPs representing each document. The numeric value W_{ji} refers to the weight of NP_i in the document D_j.

	V_1/D_1	V_2/D_2	…	V_j/D_j	…
NP_1	W_{11}	W_{21}		W_{j1}	
…	…	…	…	…	…
NP_i	W_{1i}	W_{2i}		W_{ji}	
…	…	…	…	…	…

Table 1. Matrix of the documents.

[2] http://www.fiz-technik.de/fiz/thesaurus.htm

3. Finally, the similarities between vectors V= (V_1, V_2, … V_j,…, V_n) are calculated. We have chosen to express the similarity through the statistical correlation. Correlation indicates the strength and direction of a linear relationship between two variables. The coefficient is represented in the interval [-1,1]. Therefore it is simple to decide whether given variables are similar or not, i.e., from non-related (-1) to matching (+1).

The following simplified correlation rule was used:

$$Corr_{XY} = \frac{Cov(X,Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}} = \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i-\bar{y})^2}}$$

Where $\bar{x}$ and $\bar{y}$ are the average values of vectors X and Y.

The correlation matrix (Figure 3) in our experiment serves as the basis for applying the clustering algorithm.

$$\begin{matrix} Corr_{11} & Corr_{11} & \cdots & Corr_{11} & \cdots & Corr_{11} \\ Corr_{21} & Corr_{22} & \cdots & Corr_{2j} & \cdots & Corr_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ Corr_{i1} & Corr_{i2} & \cdots & Corr_{ij} & \cdots & Corr_{iN} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Corr_{N1} & Corr_{N2} & \cdots & Corr_{Nj} & & Corr_{NN} \end{matrix}$$

Figure 3. Correlation matrix, where the main diagonal of the matrix $Corr_{ii} = 1 \;; \forall\, i = 1,\ldots N$

3a. For some of the experiment's settings the correlation matrix was recalculated for a second time in the same way as presented in phase 3.

now $Corr_{neu}(D_i;D_j) = Corr(D'_i;D'_j)$

Where $D'_i = \begin{pmatrix} corr_{1i} \\ corr_{2i} \\ \vdots \\ corr_{ii} = 1 \\ \vdots \\ corr_{ji} \\ \vdots \\ corr_{Ni} \end{pmatrix}$ $D'_j = \begin{pmatrix} corr_{1j} \\ corr_{2j} \\ \vdots \\ corr_{ij} \\ \vdots \\ corr_{jj} = 1 \\ \vdots \\ corr_{Nj} \end{pmatrix}$

And where $Corr_{ii}=0$, $Corr_{ji}=0$, $Corr_{jj}=0$, and $Corr_{ij}=0$.

$$corr_{neu}(D_i; D_j) = corr\left\{ \begin{pmatrix} corr_{1i} \\ corr_{2i} \\ \vdots \\ corr_{ii} = 0 \\ \vdots \\ corr_{ji} = 0 \\ \vdots \\ corr_{Ni} \end{pmatrix}, \begin{pmatrix} corr_{1j} \\ corr_{2j} \\ \vdots \\ corr_{ij} = 0 \\ \vdots \\ corr_{jj} = 0 \\ \vdots \\ corr_{Nj} \end{pmatrix} \right\}$$

After this stage, we get a new matrix based on correlation of correlations between documents (Table 2.).

The 2-pass correlation method enhances the contrast of similarity values. The similarity values obtained in this way are distributed differently than by the standard correlation measurements. The contrast between most similar documents and not-so-similar documents is a lot higher, as shown in Table 3: the list of the 10 most similar documents, calculated by 1-pass correlation and 2-passs correlation method.

	D'_1	D'_2		D'_j		D'_R
	D_1	D_2	.....	D_j	.....	D_N
D_1	$corr_{11}=1$	$corr_{12}$		$corr_{1j}$		$corr_{1N}$
D_2	$corr_{21}$	$corr_{22}=1$		$corr_{2j}$		$corr_{2N}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
D_i	$corr_{i1}$	$corr_{i2}$		$corr_{ij}=1$		$corr_{iN}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
D_N	$corr_{N1}$	$corr_{N2}$		$corr_{Nj}$		$corr_{NN}=1$

Table 2. The similarity matrix calculated using the 2-pass correlation method.

1-pass correlation similarity values		**2-passs correlation similarity values**	
FEEDBACK BRIDGING FAULTS	%	FEEDBACK BRIDGING FAULTS	%
1) bridging and stuck-at faults	66	1) design of fault-tolerant clocks with realistic failure assumptions	98
2) on undetectability of bridging faults	61	2) efficient distributed diagnosis in the presence of random faults	98
3) test generation for mos complex gate networks	56	3) software schemes of reconfiguration and recovery	98
4) a nine-valued circuit model to generate tests for sequential circuits	52	4) towards totally self-checking delay-insensitive systems	97
5) sharpe 2002: symbolic hierarchical automated reliability and performance	51	5) on partial protection in groomed optical wdm mesh networks	97
6) concurrent fault diagnosis in multiple processor systems	51	6) test generation for mos complex gate networks	95
7) the algebraic approach to faulty logic	51	7) concurrent fault diagnosis in multiple processor systems	86
8) a two-level approach to modeling system diagnosability	51	8) computer-aided design of dependable mission critical systems	86
9) design of fault-tolerant clocks with realistic failure assumptions	51	9) efficient byzantine-tolerant erasure-coded storage	86
10) a model of stateful firewalls and its properties	51	10) bridging and stuck-at faults	86

Table 3. Similarity values for article “Feedback Bridging Faults”: similarities in % to other 10 most similar articles calculated according to 1-pass correlation (on the left) and 2-passs correlation (on the right) methods.

4. When performing document clustering the aim is to divide a quantity of documents into theme-specific groups. These groups are not known in advance. In other words **the task** is:

For a given directory of elements DIR (NPs, documents) and similarity degree, distance d, between any two elements from DIR, we need to find a quantity C of groups of items from DIR. After calculating the similarities between each document and all other documents, we gain a similarity matrix which serves as a basis for applying the clustering algorithm.

5.4 Clustering algorithm

In this section we describe the clustering algorithm (Figure 4) that is similar to the hierarchical clustering algorithm as it is described in Johnson (1967), or Manning and Schütze (1999), but has the additional constraint that a document can appear only once in single cluster. The clustering algorithm is described as follows:

DIR: the document directory
N: the number of documents
I: the number of clusters
J: documents that have been clustered
$Wert: the chosen similarity threshold

```
1. Load the similarity matrix (Corr) NxN
2. Choose a value of Wert$
3. I = 1 and J = 0
4. Start with basic cluster DIR
5. For all documents in DIR repeat:
5.1. I = I +1;
5.2. Create a new empty cluster I
5.3. Take from DIR and put the first document De in the cluster I
5.4. J = J +1;
5.5. foreach document Df from DIR (N-J document):
5.5.1. If Corr(De,Df)>= $Wert then:
 5.5.1.1. put document Df from DIR to cluster I.
 5.5.1.2. Go to 5.5.
5.6. J = J +1;
5.7. Go to 5.
```

For the experiment, we have also used another variation of the same algorithm, allowing the same document to appear in many clusters.

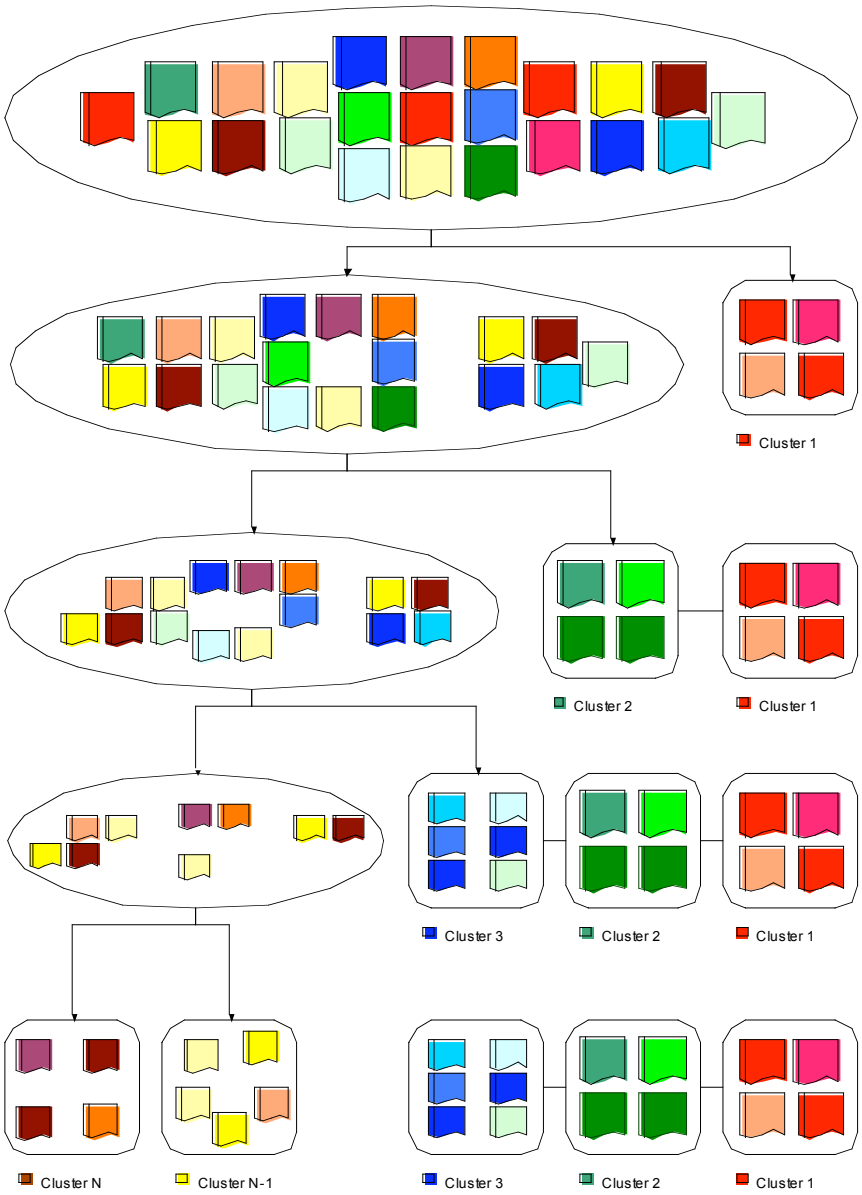

Figure 4. Adaptive hierarchical clustering algorithm.

5.5 The experiment

Experimental material. We have taken 2830 scientific articles from the domain of computer dependability and security.

For the experiment we have used different settings, that can be grouped as follows:

- A document setting. A document can appear only once in a single cluster, meaning that when documents are grouped according to the clustering algorithm, one document is assigned to one cluster only. Otherwise a document can appear in many clusters.
- Similarity measurements can be calculated either 1 time or 2 times (as presented in section 2).

The threshold $Wert was selected by experts. As an optimal threshold 0.7 was chosen. The motivation was that a smaller value threshold delivers too big clusters, i.e., an irrelevant document is more likely assigned to a cluster. On the other hand, when the threshold value is set too high, clusters tend to be very small which is unwanted for the purpose of searching in RKB. As a side effect, quite many documents remain unclustered. Results of the experiment are presented in Table 4 and Figure 5. Columns represent

different experimental settings, i.e. 1-pass correlation and document appearing in many clusters, 1-pass correlation and document appearing in a single cluster, 2-pass correlation and document appearing in many clusters, and 2-pass correlation and document appearing in single cluster. For the purposes of RKB, the method that is able to assign the majority of the documents into clusters is considered better, as well as the method that finds relatively a lot of clusters that contain 4-10 documents per cluster. A distribution with a lot of small clusters, i.e., with 2 or 3 documents, or large clusters, i.e. 50, 100 and more, is unwanted. Therefore the 2-pass correlation method allowing a document to appear in a single cluster, was the most appropriate.

	1-pass Corr +Clus	1-pass Corr 1-Clus	2-pass Corr +Clus	2-pass Corr 1-Clus
#Clusters	287	199	149	88
#Unclustered documents	399	538	33	52

Table 4. The number of clusters found from 2469 documents, when the similarity threshold is 0.7.

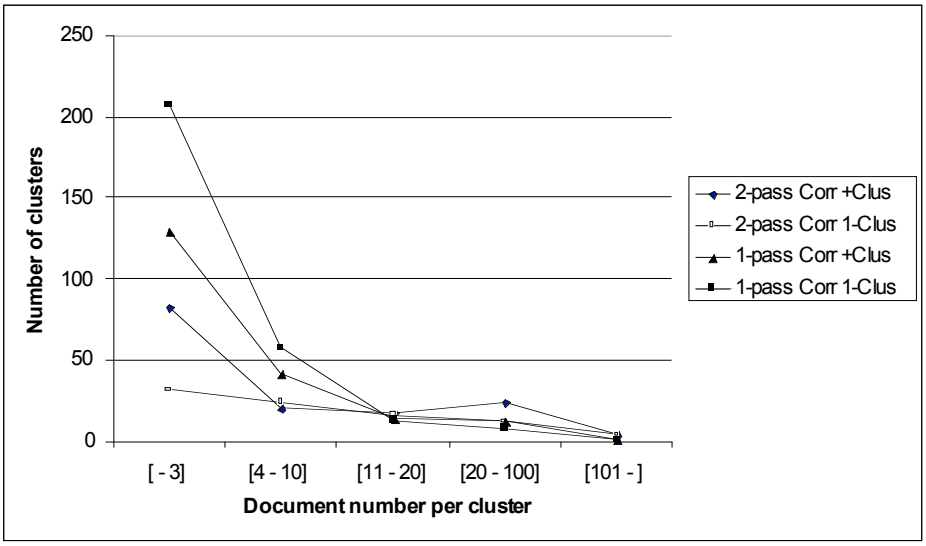

Figure 5. 4 experiments: the distributions of number of documents per cluster.

The results of the experiment in 4 different settings are presented in Figures 6-9. As we see from the diagrams, none of the experimental settings was perfect in a sense of producing even clusters (see Figures 6-9, y-axes – the number of document per cluster, x-axes - clusters), as in all 4 experiments large clusters were found. The 1-pass correlation method allowing a document to appear in many clusters performed quite good in this sense, however more than 50% of clusters found by this method contain 2-3 documents.

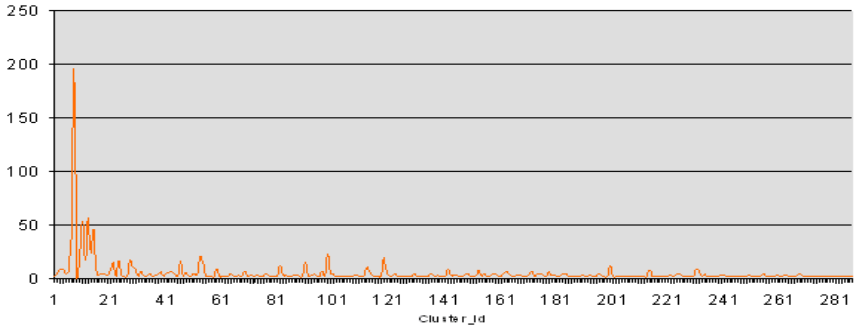

Figure 6. The size of clusters learned by 1-pass Corr +Clus method.

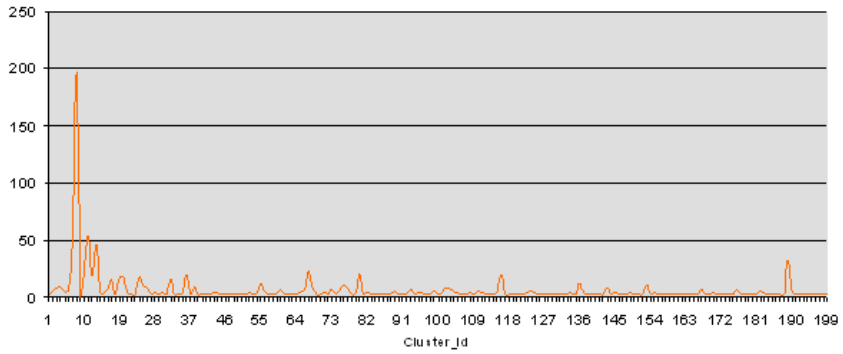

Figure 7. The size of clusters learned by 1-pass Corr 1-Clus method.

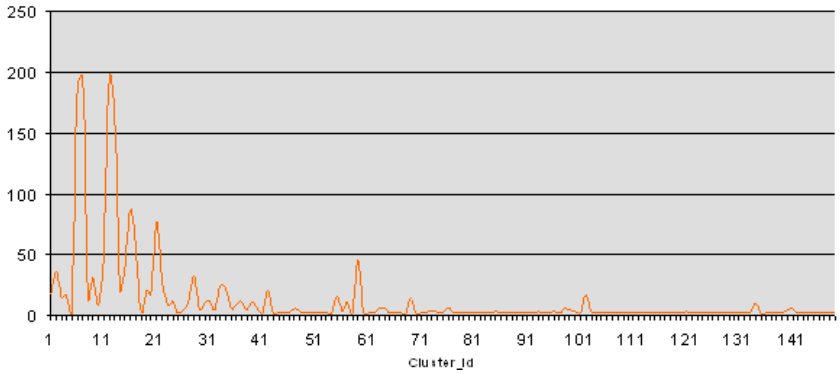

Figure 8. The size of clusters learned by 2-pass Corr +Clus method.

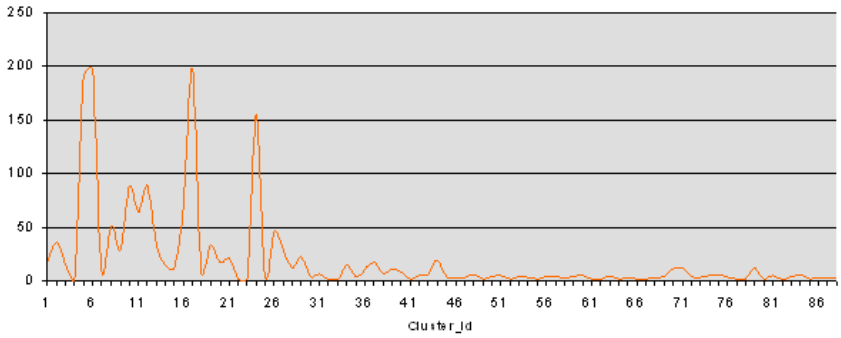

Figure 9. The size of clusters learned by 2-pass Corr 1-Clus method.

The evaluation of 10 % of the experiment results was performed manually by experts of the domain. The most relevant clusters were created with the 2-pass correlation method and allowing a document to appear in many clusters. These settings worked best out of the 4 previously described settings. The results of this experiment were applied in the RKB for providing the list of the most relevant publications related to the browsed document.

6. The resilience ontology

An ontology is a formal representation of a set of relevant concepts and relationships of a domain. The basic terms of the domain of resilient computing are characterized in the "ALRL" paper (Avižienis et. al. 2004) that provides in depth descriptions and classifications of threats, means, and attributes of dependability and security mostly by text and by some diagrams. This widely accepted scheme is an excellent blueprint for building an ontological representation of this domain.

6.1 Initial resilience ontology

An effort to create an ontology corresponding to the ALRL paper, within the ReSIST activity, was initially carried out by Brian Randell at the University of Newcastle upon Tyne in September 2006[3]. In the following however, the classification scheme is discussed whether it appropriately reflects the implicitly given characterizations of the underlying paper.

Our own analysis of that part of the ontology dealing with the various types of faults instantly revealed that this hierarchy contained almost no multiple inheritance, i.e., that the sub-fault relationship spanned a tree rather than a graph as shown in Figure 10Figure (the arrows – from right to left – represent the sub-fault relationships). This results in an inappropriate and sometimes misleading categorization of faults. For instance, "Fault-Phase" is defined to be the direct, more general fault of "Development-Fault". Obviously, these two concepts refer to disjoint aspects of a fault categorization (time vs. kind).

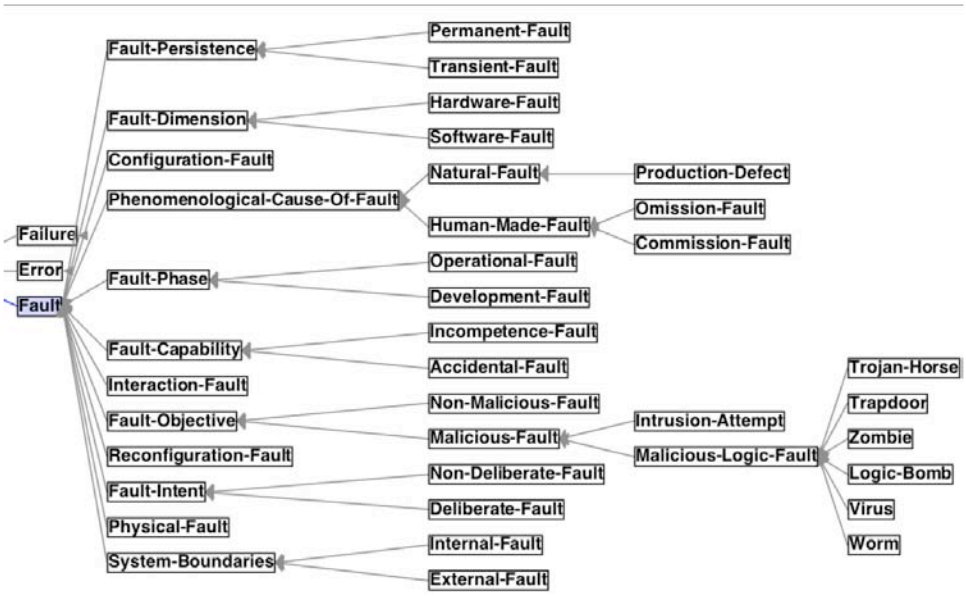

Figure 10: Fault categories as of ALRL ontology (wrt. sub-fault relationship)

Another example of inappropriate modeling can be found in Section 2.1 of (Avižienis et. al. 2004) which deals with basic concepts of the domain such as the structure of systems. Within this section, a "System-Specification" is specified as something that "describes" a system. However, in the given ontology a "System-Specification" is defined as a special system (subclass of System) as one can see in the following DAG representation of Figure 11.

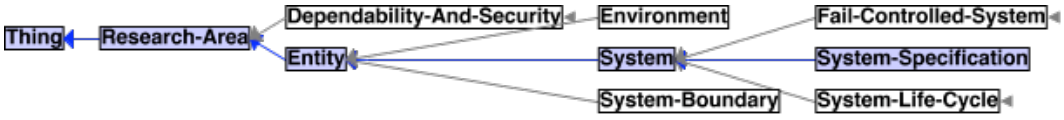

Figure 11: System categories as of ALRL ontology

Moreover, on a more detailed level components are stated to be systems by their own (3rd paragraph). An atomic component furthermore is described as being non-decomposable. It may go too far for the purpose of this ontology but from an ontology modeling perspective

[3] http://resist.ecs.soton.ac.uk/ontologies/resist.owl

one would define those concepts as follows (written in a syntax mostly following OWL abstract syntax):

```
Component = (and System (some is-part-of System))
Atomic-Component = (and Component (= 0 has-part))
```

Depicted in our ontology modeling environment OntoTrack the system sub-categories would look like the description given in Figure 12 where is-part-of is the inverse relationship to has-part.

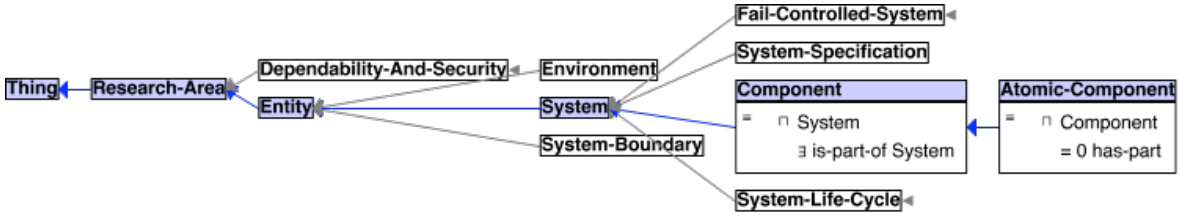

Figure 12: Fraction of revised system categories

Furthermore, System-Life-Cycle is specified as sub-concept of System (see hierarchy excerpt above) whereas in sec. 3.1 it is stated that a System "has" a life cycle which "consists" of certain phases.

6.2 Revised resilience ontology

Concerning the categorization of faults, (Avižienis et. al. 2004) accounts for eight basic viewpoints which lead to various overlapping groupings. Figure 13 shows the eight viewpoints on the left whose possible combinations lead to 31 fault classes (bottom row).

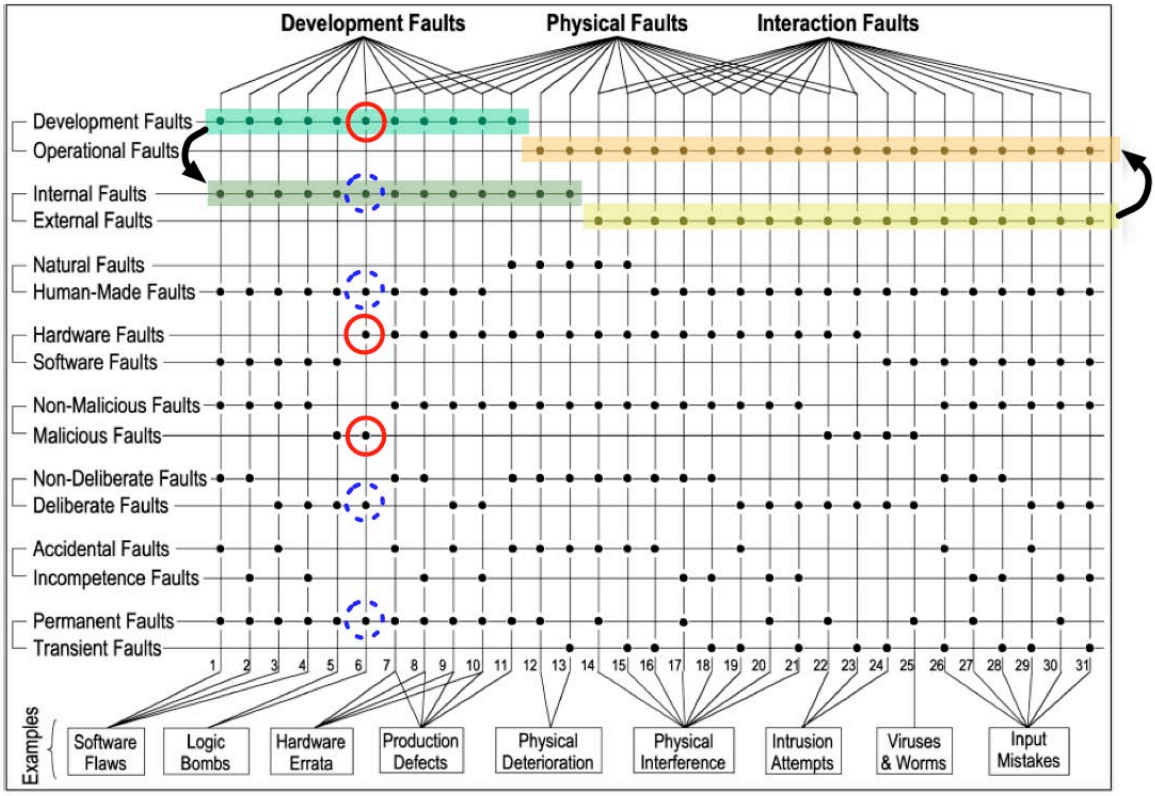

Figure 13: Fault categories as of Fig. 5a in (Avižienis et. al. 2004)

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 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 14. The arrows represent the sub-concept relationships and semantically equivalent faults are drawn within a surrounding box.

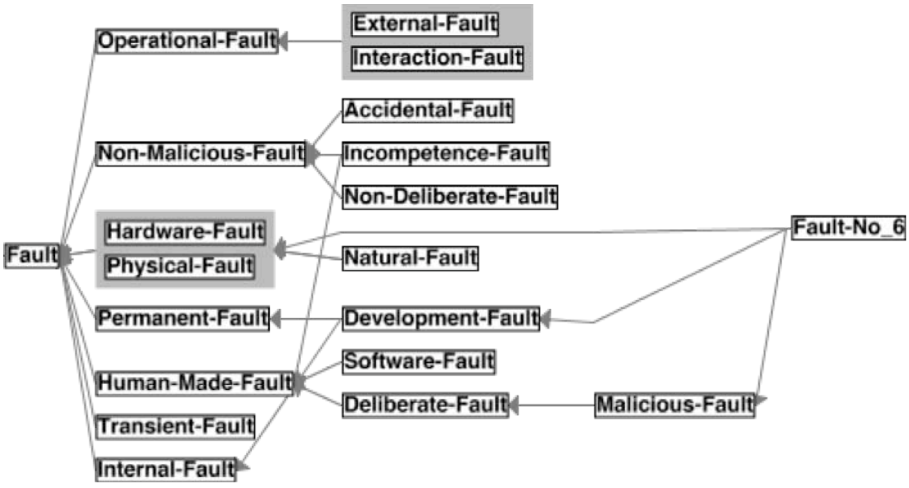

Figure 14: Revised fault hierarchy

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. A lack of obvious domain facts was identified as one major obstacle to successful knowledge based systems in a survey of different efforts of formalizing knowledge **Erreur ! Source du renvoi introuvable.**(Fiedland et. al. 2004).

Here, we consider one of the 31 elementary fault classes (fault no. 6 "logic bomb") to show its position within the revised ALRL fault ontology. The classification of fault no. 6 can also be deduced from analyzing the fault matrix of (Avižienis et. al. 2004) in Figure 13 and taking the sub-fault relationships into account. For instance, fault no. 6 is marked to be a "Development" as well as "Internal Fault". Since the former is a sub-fault of the latter (see Figure 14) it is completely sufficient to make fault no. 6 a specialization of "Development Fault" from a semantical perspective. Figure 13 distinguishes the semantically required and redundant sub-fault relationships with red resp. dashed green circles for fault no. 6. All other 30 elementary faults can be classified analogously.

7. Thesaurus mapping

7.1 Introduction

To sum up, we are facing two different approaches for organizing knowledge in the domain of resilient computing: on one side, the resilience ontology has been carefully constructed and re-structured after revision in order to correctly reflect the domain knowledge and to improve its usage. This process of refinement and revision is the result of various discussions by domain experts. The ontology is mainly handcrafted and it is consistent with respect to the underlying logical formalism which allows us to detect, among other things, inconsistencies. On the other hand, the thesaurus has been primarily compiled from a wide range of documents in an unsupervised and automatic manner. However, such an approach leads to a thesaurus which contains not only redundant but also irrelevant – with respect to the resilience domain – and rarely used (or even misused) terms. For instance, some terms are too general to be included in a thesaurus for the domain of resilient computing, but they should rather be part of a vocabulary of the domain of computer science in general. If a term is rarely used in the text corpus it does not necessarily mean that one can discard it. Feedback from domain experts is needed again.

In order to benefit from both kinds of knowledge we need to bridge the results of both approaches. In the next section we describe the process of mapping terms from the thesaurus and descriptions from the ontology. Then we present a user-friendly plug-in for our ontology authoring environment to support the task of mapping.

7.2 Mapping process

Numerically speaking, about 8000 thesaurus terms are facing about 180 well-defined concepts in the ontology. However, we observed that only a small set of relevant terms actually need to be mapped because the primary structural element of the thesaurus as well as the ontology is the hypernym-hyponym (aka. sub-set or sub-concept) relationship. Here tool support is recommended which semi-automatically supports the user in finding the best mapping.

Depending on the relevance of terms with respect to the resilience domain we discovered several different levels of granularity: branch versus leaf mapping. Leaf mapping means to map single terms to concepts in the ontology. This can be considered as a one-to-one mapping. Utilizing a branch mapping a more general term is mapped to a specific concept. Due to the sub-set relationship all sub-terms are also mapped to the given concept. For instance, the terms related to notion of faults are key concepts in the domain of resilient computing. Therefore they are typically mapped one-by-one. However, other terms such as *algorithmic circuit verification* or *online fault diagnosis* can either be included in the ontology or be mapped via their corresponding and already existing hypernym (i.e. *verification, diagnosis*). Here again, domain experts need to agree whether a term is to specific to be included in the ontology.

We identified the following four kinds of mappings:

(1) Creating one-to-one mapping between term and concept: By mapping a term to a concept (or, respectively, a concept to a term) one establishes 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 domain of resilient computing. The links do not necessarily form an one-to-one relationship: the same term can be linked to several concepts.

(2) Introducing equivalence between terms. Current thesaurus creation process does not consider the synonymy issue. 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.

(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.

(4) Discarding non-relevant terms. Revising the thesaurus terms by discarding non-relevant terms wrt. 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 semantic of some terms is not clear (i. e. combinations of faults) or does not refer to any term in the domain of resilient computing. It is clear that this can only be decided with the help of domain experts.

7.3 Mapping plug-in

In this section we introduce our mapping component that is implemented as a plug-in for the knowledge workbench OntoTrack. OntoTrack uses modern visualization and interaction techniques in combination with logical reasoning to efficiently support the user in understanding ontologies, to sniff out possible modelling errors, as well as to provide an intuitive access to ontology modelling. Therefore, our plug-in for the mapping tasks benefits from various reasoning and consistence checking capabilities to assist the user by

automatically detecting possible problems, as well as from its easy use even for novices in ontology authoring. For instance, mapping a term to different concepts within the same sub-hierarchy (with respect to the hypernym-hyponym relationship) can be simplified to mapping which contains only the most specific concept(s), e.g. a mapping from a specific fault term in the thesaurus to both the most general concept fault and the specific concept related to the fault term can be simplified by only considering the mapping of the specific term/concept because of the sub-set inclusion.

In order to provide an initial mapping between thesaurus and ontology, our tool utilizes quick and simple NLP techniques such as *entity matching* and *hyphen or whitespace* character recognition. Here, one can also consider more advanced techniques such as variant detection. Utilizing the plug-in management of OntoTrack it is quite easy to enhance our mapping plug-in with further NLP techniques.

Following OntoTrack's mantra to focus on the user and to make interactions useful and understandable for the user our plug-in is seamlessly integrated into the ontology authoring environment: By dragging concepts from the ontology view provided by OntoTrack (see right hand side of Figure 15) to the hierarchy of terms (left-hand side) users can easily establish an one-to-one link. Utilizing state-of-the-art Tablet PCs with pen-like or similar input devices an expert can intuitively perform this task without remembering a workflow of consecutive mouse-clicks, context menus etc. In contrast, such a simple metaphor may also eliminate a possible expert's dislike to utilize the plug-in.

Following the simple drag-'n-drop approach for establishing mappings, an equivalence relationship between thesaurus terms (see mapping (2) in Section 7.2) is created by dropping a term to another term in the thesaurus. Enriching the ontology by adding terms as new concepts to the ontology means adding this term as a sub-concept to an existing one. Moreover, whole sub-hierarchies of terms can be marked and mapped via one single operation that adds all terms of this hierarchy to the ontology while preserving the sub-set relationship. As a matter of course, concepts that are already contained in the hierarchy are re-used instead of being duplicated.

To increase the tool's acceptance by experts it is important that they could immediately see their benefit – or if it is not possible, they should get feedback about their progress. Even if the initial mapping automatically downsizes the number of terms in the thesaurus, there are still a lot of terms. Therefore, all already mapped terms can be made invisible which results in a tree-structured list that becomes smaller and smaller.

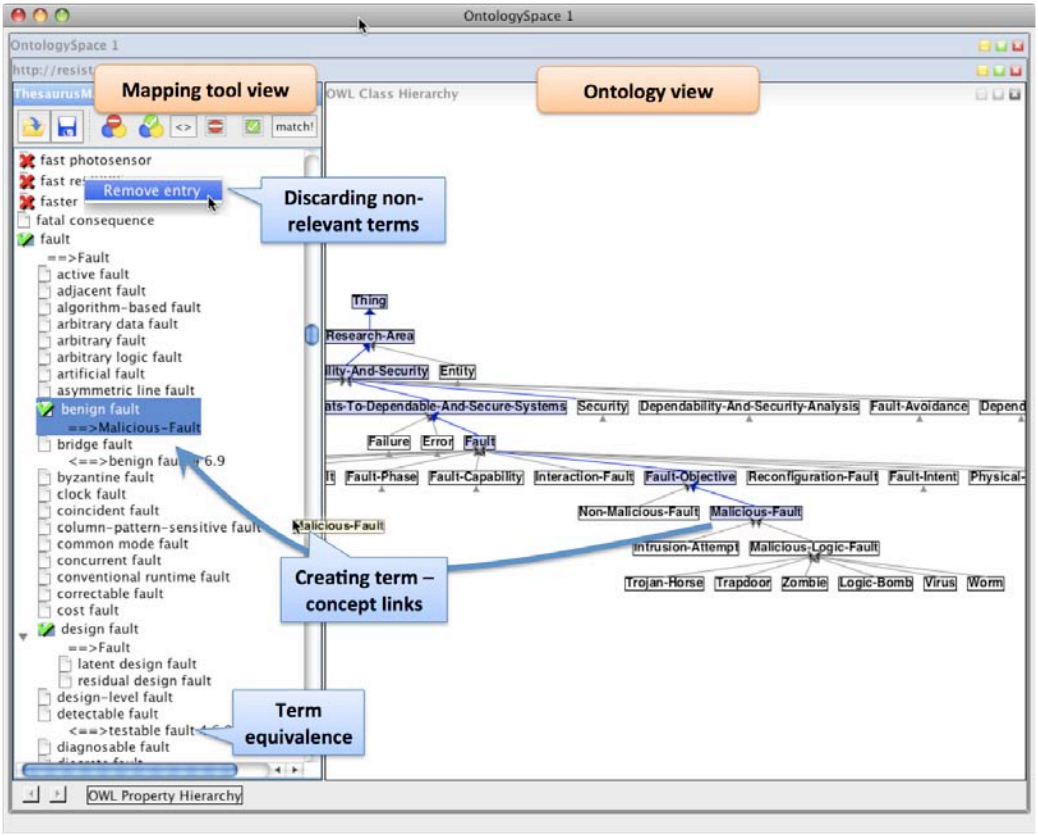

Figure 15: Mapping tool.

8. Conclusions

The approach of our work 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 include:

- A *thesaurus of the domain*, which was constructed automatically from the corpus of the resilience domain. The thesaurus contains 7,974 terms. Terms are structured via hypernym-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. 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.
- An *interactive mapping tool* implemented as a plugin to our ontology authoring framework OntoTrack. The mapping tool supports domain experts in establishing links between the thesaurus and concepts of the resilience ontology.

- 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, it is especially beneficial for domains that have no prior knowledge resources, i.e. glossaries, thesauri, organized bases of domain documents. There are several promising applications for the results of this research:

- Automatic annotation of texts that are submitted to be reviewed for publication
- Automatic identification of potentially related publications
- Focused intelligent search in large document sets
- Mediation between different dialects of a domain with several near-synonyms

The contributors to this research effort intend to continue the work with sponsorship of IFIP and thus open the participation to members of a world-wide organization of researchers. Another direction of future work is the creation of a thesaurus and an ontology of the entire field of informatics, or computer science and engineering.

9. Acknowledgments

This report presents the research performed by several members of the SIG ResOn of the ReSIST project. The report has been edited by Gintare Grigonyte (VMU Kaunas). The principal authors are Algirdas Avizienis, Gintare Grigonyte (both VMU Kaunas), Friedrich von Henke, Thorsten Liebig, Olaf Noppens (all Ulm University), Johann Haller, Mahmoud Gindyeh (both Affiliate members, IAI Saarbruecken). Additional contributions have been made at various times by ReSIST members Hugh Glaser, Jean-Claude Laprie, Ruta Marcinkeviciene, Ian Millard, Brian Randell, and Robert Stroud.

10. References

Avižienis, A., Grigonyte, G., Haller, H., von Henke, F., Liebig, T., Noppens, O. 2009. Organizing Knowledge in the Domain of Resilience Computing by Means of Natural Language Processing and Ontologies - An Experience Report -. Proceedings of FLAIRS-22 , Sanibel Island, FL, USA. AAAI Press, May 2009.

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.

Bennet N. 1999. Extracting Noun Phrases for all of MEDLIN. In Proc. American Medical Informatics Assoc. Symposium, AMIA.

Bourigault D., Jacquemin C., and M.-C. L'Homme, editors. 2001. Recent Advances in Computational Terminology. John Benjamins Publishing Company.

Bourigault, D. 1992. Surface grammatical analysis for the extraction of terminological noun phrases. In Proceedings of COLING-92, 977-981.

Carl, M., Haller, J., Horschmann. C, Theofilidis, A. 2002. A Hybrid Example-Based Approach for Detecting Terminological Variants in Documents and Lists of Terms. In 6. KONVENS, Saarbrücken.

Carl, M., and Schmidt-Wigger, A. 1998. Shallow post-morphological processing with KURD. Proceedings of the Conference on new methods in language processing (NeMLaP).

Daille, B., Gaussier, E., and Langé, J.M. 1994. Towards automatic extraction of monolingual and bilingual terminology. In Proceedings of COLING-94., 515-521.

Fiedland, N. S.; Allen, P. G.; Witbrock, M.; Matthews, G.; Salay, N.; Miraglia, P.; Angele, J.; Stab, S.; Israel, D.; Chaudhri, V.; Porter, B.; Barker, K.; and Clark, P. 2004. Towards a Quantitative, Plattform-Independent Analysis of Knowledge Systems. In Proc. of the Ninth International Conference on Principles of Knowledge Representation and Reasoning, 507–514. Whistler, BC, Canada: AAAI Press.

Frantzi K. and Ananiadou S.. 1999. The c-value/nc-value domain independent method for multiword term extraction. Journal of Natural Language Processing, 6(3):145–179

Glaser, H., Millard, I., Rodriguez-Castro, B. and Jaffri, A. 2007. Knowledge-Enabled Research Infrastructure. In: 4th ESWC, Austria.

Haller, J. 2006. Multiperspektivische Fragestellungen der Translation in der Romania. Frankfurt: Peter Lang Verlag.

Haller, J., Schmidt, P. 2006. AUTINDEX - Automatische Indexierung. Zeitschrift für Bibliothekswesen und Bibliographie: Sonderheft 89, Klostermann, Frankfurt am Main S. 104-114.

Hong ,M., Fissaha, S., and Haller, J. 2001. Hybrid filtering for extraction of term candidates from German technical texts. Proceedings of the Int. Conference on Terminology and Artificial Intelligence (TIA 2001), 223-232.

Jisong Chen. 2006. A Multi-word Term Extraction System. PRICAI 2006: Trends in Artificial Intelligence, Springer, 1160-1165

Johnson S. C. 1967. Hierarchical Clustering Schemes. Psychometrika, 2:241-254.

Kouomou, A., Berti-Équille, L., Morin, A. 2005. Optimizing progressive query-by-example over pre-clustered large image databases, Proceedings of the 2nd international workshop on Computer vision meets databases, Baltimore, MD.

Laprie, J.-C. 2008. From Dependability to Resilience. LAAS Report no. 08001. LAAS-CNRS, Toulouse, France.

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 Woerter.

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

Nakagawa, H. 2001. Experimental Evaluation of Ranking and Selection Methods in Term Extraction. In: Recent Advances in Computational Terminology. John Benjamins Publishing Company, 303-325.

Paulo, J. L. et al. 2002. Using Morphological, Syntactical, and Statistical Information for Automatic Term Acquisition. In E. Ranchhod and N. Mamede (eds.), Advances in Natural Language Processing, PorTAL. Springer-Verlag, LNAI 2389: 219-227

Schiller, A. 1996. Multilingual Finite-state noun phrase extraction. In: Proc. ECAI-96 Workshop on Extended Finite State Models of Language.