ulm university universität
uulm

EVALUATING BENEFITS OF REQUIREMENT CATEGORIZATION IN NATURAL LANGUAGE SPECIFICATIONS FOR REVIEW IMPROVEMENTS

Daniel Ott, Alexander Raschke

Ulmer Informatik-Berichte

Nr. 2013-08
Oktober 2013

Evaluating Benefits of Requirement Categorization in Natural Language Specifications for Review Improvements

Daniel Ott
Research and Development
Daimler AG
P.O. Box 2360, 89013 Ulm, Germany
daniel.ott@daimler.com

Alexander Raschke
Inst. of Software Engineering
University of Ulm
Ulm, Germany
alexander.raschke@uni-ulm.de

***Abstract*—One of the most common ways to ensure the quality of industry specifications is technical review, as the documents are typically written in natural language. Unfortunately, review activities tends to be less effective because of the increasing size and complexity of the specifications. For example at Mercedes-Benz, a specification and its referenced documents often sums up to 3.000 pages. Given such large specifications, reviewers have major problems in finding defects, especially consistency or completeness defects, between requirements with related information that are spread over large or even different documents.**

The classification of each requirement according to related topics is one possibility to improve the review efficiency. The reviewers can filter the overall document set according to particular topics to check consistency and completeness between the requirements within one topic.

In this paper, we investigate whether this approach really can help to improve the review situation by presenting an experiment with students reviewing specifications originating from Mercedes-Benz with and without such a classification.

In addition, we research the experiment participants' acceptance of an automatic classification derived from text classification algorithms compared to a manual classification and how much manual effort is needed to improve the automatic classification.

The results of this experiment, combined with the results of previous research, lead us to the conclusion that an automatic pre-classification is an useful aid in review tasks for finding consistency and completeness defects.

***Keywords*-experimental software engineering; review; topic; topic landscape; classified requirements; inspection**

I. Introduction

Current industry specifications get more and more complex and voluminous, and it is still common that they are written in natural language (NL) [1]. For example in the automotive industry, in this case by Mercedes-Benz, a natural language specification and their referenced supplementary specifications, often have more than 3,000 pages [2]. Supplementary specifications can be, for example, internal or external standards. A typical specification at Mercedes-Benz refers to 30-300 of these documents [2]. The information related to one requirement can be spread across many documents. The common way to ensure the quality in these specifications is technical review, but because of the above reasons, it is difficult or nearly impossible for a reviewer to find consistency and completeness defects in the specification and between the specification and referenced supplementary specifications. This is also reported in a recent analysis of the defect distribution in current Mercedes-Benz specifications [3].

Considering the huge amount of requirements, it is obvious that the identification of topics and the classification of requirements to these topics must be done automatically to be of practical use. In this paper, we present a tool-supported approach to automatically classify requirements with related information and to visualize the resulting requirement classes. The classification is done by applying text classification algorithms like Support Vector Machines (details see Section II-B), which use experience from previously classified requirement documents. The framework for this automatic classification and visualisation of requirements from various documents is called ReCaRe standing for "**Re**view with **Ca**tegorized **Re**quirements".

In previous work [4], we evaluated text classification algorithms in ReCaRe on two large Mercedes-Benz specifications and investigated possible improvements. The results of this evaluation showed that an automatic classification to various topics is feasible with high accuracy.

In the current work, we investigate, whether or not a manual classification of requirements actually helps reviewers to find special kinds of defects. We further research, if an automatic classification is acceptable for reviewers, because it will not necessarily find all relevant requirements, but will almost always add additional, not relevant requirements. Finally, we will investigate, how accurate the results of an automatic classification really need to be to aid in review tasks and if these results can be strongly adjusted with little manual efforts during the review. We investigate these points in an experiment with ten students reviewing three automotive specifications originating from real Mercedes-Benz specifications.

Section II provides an overview of the approach of collecting requirements of related information into classes - we

call this concept “topic landscape”. We also present the tool ReCaRe, which realizes the topic landscape, and its concepts e.g. the classification algorithms. Section III presents experiment goals, structure and results. These results are discussed in Section IV. In Section V we discuss related work and finally, in Section VI we conclude with a summary of the contents of this work and describe our planned next steps.

II. The Topic Landscape Approach

The topic landscape aims at supporting the review process by classifying the requirements of the inspected specification and its additional documents into topics. A topic is defined by one or more key words. For instance, the topic “temperature” is defined by key words like “hot”, “cold”, “heat”, “°C”, “Kelvin” or the word “temperature” itself.

All requirements classified into a particular topic can be grouped for a specific review session. Due to this separation of the specification and its additional documents into smaller parts with content related requirements, a human inspector can more easily check these requirements for content quality criteria like consistency or completeness, without searching every single relevant document.

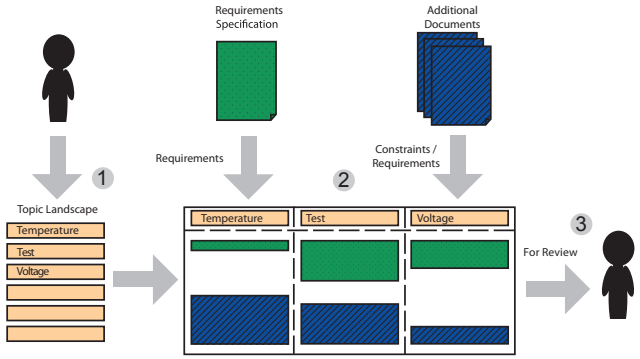

Figure 1: Illustration of the Topic Landscape Approach

Figure 1 illustrates the individual steps in order to use topic landscape:

1) The user/author creates the topic landscape as a container of relevant topics for this particular specification. Each topic is described by one or more keywords.
2) Each requirement of the specification and the requirements/constraints of the additional documents are classified into individual topics.
3) The inspector chooses one topic from the topic landscape and checks all requirements assigned to the chosen topic for defects.

In this work, we research the needed performance of classifiers to automatically perform Step 2. Step 1 could also be performed semi-automatically by a sophisticated algorithm, but this remains future work.

The content of a topic may not be considered disjoint from other topics since a requirement normally includes information on different topics and thus will be assigned to several of them. For instance, the requirement “The vehicle doors must be unlocked when the accident detection system is not available.” highlights many topics including, but not limited to, accident detection, accident, detection, availability, locking, vehicle door, door, security, door control, and functionality.

A. ReCaRe

The tool ReCaRe (**Re**view with **Ca**tegorized **Re**quirements) is the realization of the topic landscape approach. Since ReCaRe is still a prototype, we focused ReCaRe on the basic use case of classifying text. Currently, ReCaRe cannot extract information from figures or tables. Our later investigated specifications contain some requirements, which consist only of figures or tables (see also Section III-B), so these requirements cannot be classified correctly with the current version of ReCaRe.

The general user interaction options available with ReCaRe are described in Section II-C.

Figure 2 shows the individual processing steps of ReCaRe. The chosen pre-processing, post-processing and classification steps have many alternatives, but after a comparison, we got the best results in previous work [4] with the illustrated setting for German natural-language specifications from Mercedes-Benz. We reuse this setting in the current work, since the evaluated specifications in the experiment have the same characteristics.

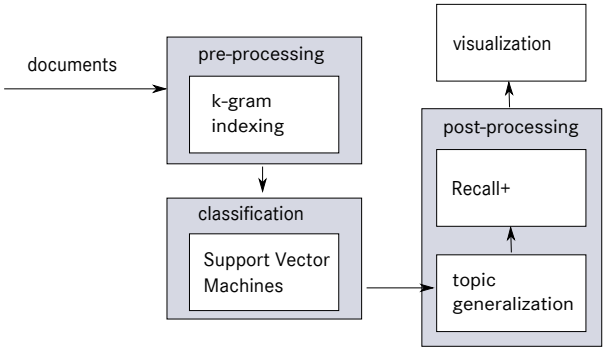

Figure 2: Processing Steps in ReCaRe

In ReCaRe we assume that a requirement can be classified to multiple topics. Therefore, we train a binary classifier for each topic, which decides if a requirement is relevant or not for a certain topic. The used classification algorithm called support vector machines is described in Section II-B. The classifier is based on the work of Witten et al. [5] and more details to the classifier can be found there.

As shown in Figure 2, we consider the following pre- and post-processing steps to improve the classification results:

Hollink et al. [6] describe k-gram indexing in detail. In short, each word of the requirement is separated in each ongoing combination of k letters and the classifier is then

trained with these k-grams instead of the whole words. For example, a k-gram indexing with k = 4 separates the word “require” to “requ”, “equi”, “quir”, “uire”. For the evaluated specifications in [4], k = 4 led to the best results.

The first post-processing step called “topic generalization” takes the structure of Mercedes-Benz specifications into account. All specifications at Mercedes-Benz are written using a template, which provides a generic structure and general requirements, and are later filled with system specific contents. Because of this structure, we assume that if a heading was assigned to a topic, then we can also assign each requirement and subheading under the heading to this topic. This is also the only way, besides the thereafter following “Recall+” approach, to correctly assign requirements to topics, which only consist of a figure or table, because ReCaRe has currently no potential to get information out of figures or tables.

Finally, there is also a possible review or ReCaRe specific enhancement: As part of the of topic visualisation, we also need to provide the ReCaRe-user with the context around of each requirement in each topic, so that the reviewer understands where in the document the specific requirement comes from. This is done by linking the requirement of the topic to the full document. So, the reader has also an awareness of the surrounding requirements during the review. Because of this, we assume that if in a later stage of the analyses an unclassified requirement is within a certain structural distance to correctly classified requirements, we can also count this requirement as classified. We call this assumption “Recall+” because it influences only this specific measure later in the evaluation. Until now, Recall+ is not proven in experiments with ReCaRe-users, therefore we also want to evaluate this thesis in the current experiment. As explained, this post-processing step is only an enhancement to the performance analysis of the ReCaRe classification and doesn’t improve the classification algorithms itself as, for example, the topic generalization.

B. Support Vector Machines (SVM)

The support vector machine approach works in ReCaRe as follows (based on [5], [7]) : A nonlinear mapping is used to transform the training data into a higher dimension. Within this new dimension, the classifier searches for the optimal separating hyperplane, which separates the class of topic relevant and topic irrelevant requirements. If a sufficiently high dimension is used, data from two classes can always be separated by a hyperplane. The SVM finds the maximum-margin hyperplane using support vectors and margins. The maximum-margin hyperplane is the one with the greatest separation between the two classes.

The maximum-margin hyperplane can be written as [5]:

$$x = b + \sum_{i \text{ is support vector}} \alpha_i * y_i * a(i) \cdot a$$

Here, y_i is the class value of training instance $a(i)$, while b and α_i are numeric parameters that have to be determined by the SVM. $a(i)$ and a are vectors. The vector a represents a test instance, which shall be classified by the SVM.

C. User Interaction with ReCaRe

After the start of ReCaRe, the user chooses the documents to review and defines a list of topics. Thereafter, these documents are loaded and their objects (requirements or headings) are automatically classified to topics using the previous described mechanisms. Then, the user gets a statistic table of all topics, with their number of objects, considered documents and later annotated defects. In the next working step, the user chooses an interesting topic to review and gets a new view, called “topic view” (see Figure 3), including a table with all assigned requirements (1). In the topic view, he can fade in the whole document to an interesting requirement in an additional view, called “context view” (3) and can also document defects (2) concerning one or more requirements.

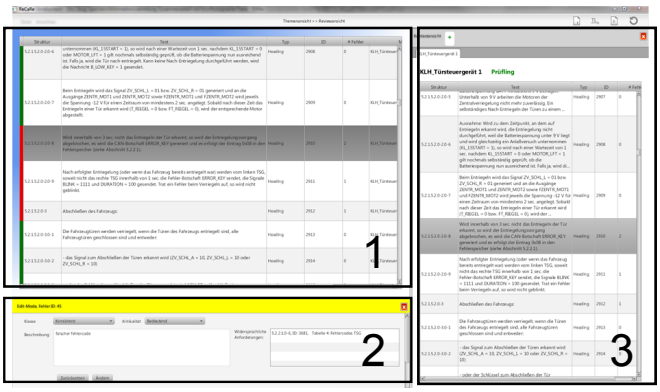

Figure 3: Illustration of Topic View

III. Evaluation

After the introduction of the topic landscape and the tool ReCaRe this section describes an evaluation of the method and the tool. This evaluation was conducted with ten students. We were able to use original specifications of Mercedes-Benz, which were only little modified due to reasons of confidentiality. Based on three questions that should be answered by this evaluation and environmental constraints, we designed several experiments as explained in Section III-B.

A. Evaluation Questions

We define the following three evaluation questions for our experiment:

- (Q1) What is the benefit of using the topic landscape for review activities?
- (Q2) Is the typical accuracy of an automatic classification acceptable and what is the at least needed recall for reviewers?

Table I: Overview of Conducted Experiments

experiment	task	data set(s)	classification	duration
1	conventional technical review	PWE, DCU	—	20 hours
2	review with topic landscape using ReCaRe	DCU, PWE	manual	20 hours
3	review in consideration of particular questions	IC	basic & best practice automatic	2 hours
4	review in consideration of particular questions with the possibility to improve classification	IC	basic & best practice automatic	4 hours

- (Q3) What is the influence of additional manual efforts to the results of an automatic classification?

In a previous experiment with students [9] we also evaluated Q1, but without the tool ReCaRe. Instead, we used the IBM Rational DOORS filter mechanism to show only requirements related to a specific topic. Unfortunately, this first experiment got us no statistically significant evidence for or against the benefit of the topic landscape approach (see [9]). Reasons are the insufficient tool-support with DOORS and other problems, like motivation loss of participants during the review. Nevertheless, we could extract many improvement ideas and lessons learned of this first experiment, which are now included in the current.

B. Evaluation Design

The evaluation of the evaluation questions mentioned in the previous Section III-A was organized with ten students at the University of Ulm. During a time period of ten weeks, four different experiments were conducted. Each of these experiments included review tasks with and without the topic landscape. Table I gives an overview of the conducted experiments, which are explained in detail in this section.

All reviews are done on three data sets based on real specifications of Mercedes-Benz. The original specifications have to be altered due to confidentiality reasons such that the description of the functionality and the interfaces is still similar but contains dummy parameters and values, and the specifications are of reduced size. Each data set consists of a component specification (C-SP), two supplementary specifications and a reduced system specification (S-SP). The systems specified by the three data sets are a door control unit (DCU), an instrument cluster (IC) and a parctronic warn element (PWE). Table II shows some statistics on these data sets. The addition of the number of requirements and the number of headings is not the number of the DOORS objects, because a DOORS object can be both heading and requirement at the same time. Compared to the data sets described in [4] and [9], some improvements mainly concerning the layout of Tables are made so that the statistics slightly differ.

In previous work [3], the distribution of different defect types is investigated based on many original review protocols of Mercedes-Benz. According to this distribution of defect types we injected 100 defects of different types into each data set described in Table II. The categorization in defect types is done using a quality model. This quality model is described in detail in [3]. Nevertheless, in this evaluation we concentrate on correctness, consistency, and completeness defects.

In contrast to the last experiment, where the students did the classification, we define the topics for each data set on our own and classify all documents manually. The identification of topics is done by two persons independently and then is merged into one list of topics for each data set. The classification of the requirements is done in a similar way. The two classifications are synchronized in a review session using Cohen's Kappa [10] as a help. Cohen's Kappa is a statistical measure to calculate the inter-rater agreement between two raters who each classify n items to x categories. The results of the manual classification are also described in Table II in the last five rows. For example, for the DCU C-SP we categorized 880 of the 901 DOORS objects to topics. Since an object can be categorized to multiple topics, we made 5947 topic assignments to objects and we identified 141 topics for the whole DCU data set. The last two lines describe the number of objects with figures and tables and how manual annotations are concerned by these objects. This information is particularly interesting for experiment three and four, because ReCaRe can hardly classify these objects correct to topics, as described in Section II-A.

The student group was a mixture of three master students and seven last-year bachelor students. For the master students, the participation was a relevant course achievement, the bachelor students were reimbursed with money for their work. After the review tasks, the students have to model parts of the specifications with statecharts. These executable models are exchanged between student groups and mutually tested. This was another incentive for a good review performance, in order to find all noticeable problems in advance. Thus, although the participation was voluntary, the motivation was (also) driven by an external source.

Before the evaluation phase itself, the students get an introduction to the process of reviews and the tool ReCaRe. Using a small example specification, the students have to find some defects, enter them in the tool ReCaRe and discuss them in a review meeting. The students also get an explanation of the underlying quality model (see above) and

Table II: Data Set Statistics

data set	Door Control Unit (DCU)				Instrument Cluster (IC)				Parctronic Warning Element (PWE)			
document	**C-SP**	**AWR**	**CTR**	**S-SP**	**C-SP**	**DR**	**CTR**	**S-SP**	**C-SP**	**DR**	**CTR**	**S-SP**
requirements	782	70	71	346	580	96	71	349	569	96	71	115
words / requirement	13.0	30.5	16.0	11.7	13.3	30.9	16.0	12.5	12.9	30.9	16.0	16.6
headings	121	13	18	173	176	27	18	164	179	27	18	51
words / heading	2.0	2.0	1.9	3.2	2.2	2.9	1.9	2.2	2.5	2.9	1.9	32.8
DOORS objects	901	83	89	519	754	123	89	513	746	123	89	166
objects to topics	880	83	89	419	472	96	71	340	524	96	71	115
topic assignments	5947	438	444	1930	1975	357	254	1198	2092	225	244	513
number of topics	141				99				94			
figures and tables	34	3	0	9	38	5	0	1	31	5	0	2
influenced topic assignments	612	16	0	62	288	36	0	7	225	27	0	11

a detailed description of each defect type. The training phase is necessary to reduce learning effects during the following experiments, to ensure a minimum level of experience with ReCaRe and the quality model, and to answer questions before the experiment starts.

The first experiment is a conventional technical review using ReCaRe to document the defects but without using the topic landscape. The students are divided into two groups and each student reviews one of the two systems DCU or PWE on his own for 20 hours within fourteen days. All defects are gathered in review meetings, one for each data set.

The second experiment is a review using ReCaRe and the topic landscape. The data sets are exchanged between the student groups. Again, each student reviews for 20 hours within fourteen days the documents of one data set accordingly to our quality model. After the conclusive review meeting, the students have to fill out a questionnaire about their assessment of normal reviews compared to a topic landscape review.

The fixed duration of 20 hours is requested in order to allow for an easier comparison of the students' results, because otherwise the time and effort of the students differ too much and it is quite difficult to define reliable metrics like found defects per hour.

In the last two experiments, we evaluate the quality of an automatic categorization derived with the text classification mechanisms of ReCaRe described in Section II. In order to measure the performance of ReCaRe, the standard metrics from data mining and information retrieval research are used: recall and precision [5], [7], [8]. In this context, a perfect precision score of 1.0 means that every requirement that a classifier algorithm labeled with a topic does indeed belong to this topic. A perfect recall score of 1.0 means that for every topic, all requirements related to this topic are assigned with this topic. We will also use the f-measure (for example introduced by Witten et al. [5]) in these experiments, to have a single measure that characterizes the performance changes in the classification. The f-measure is calculated as follows:

$$f\text{-}measure = \frac{2 * recall * precision}{recall + precision}$$

The quality of the machine learning algorithm is measured by the k-fold cross validation, which is a well known validation technique in data mining [11], [5], [12], [7]. The manually classified data sets are shuffled and split into k parts of equal size. k-1 of the parts are then used for training the classifier algorithm. With the trained classifier algorithm the remaining part of the data set is classified for evaluation.

This procedure is executed k times each time with a different part being held back for classification. After that, the complete process is repeated k times. The classification performance averaged over all k parts in k iterations characterizes the classifier. As shown by Witten et al. [5], using a value of 10 for k is common in research, so this value is used in this paper, too.

Since the main focus in the following experiments is on the performance and acceptance of the automatic classifying algorithm, the reviews are not carried out with the complete specification, but only with parts of it.

In addition, the students are again separated into two groups: One group gets the data set classified with the basic SVM algorithm (without pre- and post-processing), the other group gets the data set classified with SVM using also the k-gram indexing and topic generalization as described in II-A. We call the first setting the "basic" and the second setting the "best practices" approach. We identified this setting as best practices in previous work [4]. We used the third (so far unkown) data set IC for these last experiments.

The third experiment is a short review focused on specific questions like "Is the communication interface between the instrument cluster and other components in the component specification and the system specification consistently documented?". The review is done using the tool ReCaRe with the described automatic classification. For this part of the experiment the students meet in one room reviewing simultaneously for two hours.

The fourth and last experiment is similar to the previous one: The students get new specific questions to focus on during review. This time, the students have four hours of

time within one week and they are allowed to improve the classification manually by removing or adding new requirements to a topic. After a modification, the classification algorithm is run again for these topics to incorporate the changes. After this phase, the students filled out a second questionnaire for the last two experiments.

All four experiments are used to answer evaluation question Q1. Q2 is answered by experiment 3 and 4. The evaluation question Q3 can be answered by the last experiment. In the following subsection the gained results for each evaluation question are described in detail.

C. Evaluation Results

In the following subsections each evaluation question is answered by proving or disproving several hypotheses presented in each subsection.

1) Q1 - What is the benefit of using the topic landscape at review activities?: The evaluation question about the benefit of the topic landscape compared to normal reviews is answered by the following hypotheses:

- (H1) Reviews employing only the topic landscape are a full replacement for normal reviews.
- (H2) Reviews employing only the topic landscape achieve better results than normal reviews.
- (H3) The topic landscape is an useful aid for normal reviews.

The students answered to a question "The normal review can be replaced by a review with topic landscape" using a six-point Likert item (1 - totally agree, 6 - totally disagree) with an average value of 3.6. That means, they do slightly disagree. A closer look to the corresponding free text answer shows their objections: With foreign specifications, it is quite difficult to recognize the context of the requirements filtered by a topic which is important to avoid any misunderstandings. By using the "context view" (see Section II-C), this issue can be improved.

Defects that occur only in one requirement and are not related to other requirements can be found by a review with and without a topic landscape. Defects affecting a wider part of the specification (e.g. consistency defects), tend to be found more easily with the topic landscape (again according to the students' answers).

Some kinds of defects might be easier to find without the topic landscape (e.g. unambiguity). Other kinds like traceability are expected to be much easier found with the topic landscape. Unfortunately, no traces are included in the investigated three data sets.

Hypothesis 2 has to be refuted. We are not able to find a significant performance improvement of reviews with topic landscape. Table III shows the number of found completeness, consistency and correctness defects. Again, one can see a wide range of found defects in each group and data set. Group 2 achieves better results with the topic landscape, group 1 doesn't. Overall, group 1 found less defects compared to group 2 in both data sets. Both groups found at least in the PWE data set more defects than in the DCU data set.

Table III: Found (and accepted) defects per participant of normal review (NR) and review with topic landscape (RTL).

	Group 1		Group 2	
data set	PWE	DCU	DCU	PWE
review kind	NR	RTL	NR	RTL
∅ completeness	4.25	7.75	11.33	13.67
∅ correctness	6.5	6	3.33	3.67
∅ consistency	15.75	8.5	16.67	23.83
∅ defects	26.5	22.25	31.33	41.17
varying from ... to ...	14–48	11–40	24–41	12–61

This result is not very surprising. Because of our earlier experiment [9], we expected this result and therefore added another hypothesis, whether the topic landscape can be used as additional help beside normal reviews. According to the students' answers, this hypothesis can be accepted. The students rated the question "Is it reasonable to use topic landscape as a supplement of a normal review for selective topics with regards to content?" with 2.7 average, again on a six-point Likert scale. They mentioned a more comfortable review experience for not too large topics with less than 70 requirements assigned and a better review performance when looking at requirements from different points of view dependent of the current topic. In addition, the students mention the inspection of requirements from different views depending on the topic and the clustering of relevant requirements on one spot as further benefits.

2) Q2 - Is the typical accuracy of an automatic classification acceptable and what is the at least needed recall for reviewers?: Question Q2 evaluates the automatic classification algorithm for a topic landscape. Since it is hardly possible to achieve 100 % recall and/or precision.

The hypotheses regarding to the evaluation question Q2 are the following:

- (H4) A recall of 100 % is not necessary because of the context in "Recall+".
- (H5) Typical recall and precision values of 80 % and 60 % are sufficient to be accepted by the users.

As described earlier (in Section III-B), the automatic classification of the data set IC using SVM is done in two different ways: a basic and a best-practices approach. The overall recall and precision of the basic approach is 0.68 and 0.73. For the best-practices approach it is 0.80 and 0.50.

After the two-hour review, the students declared they used in average ten topics for the review. Five students complained that they did not find the necessary requirements in the supposed topics (lack of recall). This problem is absorbed by using the context view of each requirement, which supports hypothesis H4. The students indicate they needed 6 to 22 requirements in average around a requirement to understand it and to review its context. In previous work

with similar data sets [4], we calculated that including only three requirements using the described "Recall+" mechanism (see Section II-A) already improves the recall by over 10 %. Indeed, recalculating the basic and best practices results with Recall+ for the participants' minimum of 6, the recall improves for the basic approach to 0.87 and for the best practices to 0.91. If we also consider the objects containing figures and tables, which are difficult to impossible to categorize with ReCaRe (see Table II in Section III-B for details), at least the best practices approach includes almost all needed requirements to topics.

The low precision especially of the best-practices classification results in five statements about too many useless (wrong classified) requirements in the topics. Altogether, the participants are comfortable with the automatic classification, although it could be improved at several points (H5).

3) Q3 - What is the influence of additional manual efforts to the results of an automatic classification?: These possible improvements lead us to the last evaluation question Q3. The corresponding hypothesis is as follows:

- (H6) The manual topic modification by adding to or removing requirements from topics improves the classification algorithm.

The students added and removed several requirements to and from topics during their review activity. After each modification, the classification algorithm was started again. All participants added 290 requirements to and removed 96 requirements from 47 topics. Out of these 47 modified topics, recall and precision are improved for 29 topics with an average increase of 32,3 % for the f-measure (details to the f-measure see Section III-B). For nine topics, the modifications caused only very little changes (f-measure changes less then 1 %) and nine modified topics result in a worsening of recall and precision (with an average decrease of 14,5 % for the f-measure).

IV. Discussion

In this chapter we discuss the results of our experiment, the applicability in industrial practices of the topic landscape approach and threats to validity to our experiment.

A. Interpretation of Results

As with many software engineering topics, it is obvious that it is not possible to gain statistically significant results with ten participants. The results depend heavily on the individual performance of each person. Table IV shows the found (and by us accepted) defects of experiment 3 (two hour review). Since this experiment was done in one room at the same time while we observed the students, we are sure no student reviewed shorter or longer than expected. Nevertheless, the results vary from 5 to 20 defects which is a difference of factor four. By chance, the averages are comparable which is obviously not generalizable. E.g., for the four hour review (results see Table V), the average number of found defects of group 2 is again 11, but of group 1 it is only 6.8.

Another issue is that the problems treated by the topic landscape only occur in reviews of voluminous specifications. Therefore, it takes much longer to review such a specification. Thus, it is not possible to conduct many small inexpensive experiments instead of a few large experiments in the research area of review improvements (see also [9]).

In consequence, we did not (only) focus on the obtained numbers but we emphasized on the two questionnaires filled out by the students (See Section III-B).

Table IV: Found (and accepted) defects per participant of third experiment (two hour review).

	Group 1					Group 2				
setting	basic					best practices				
participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
completeness	2	6	4	3	2	1	7	10	2	1
correctness	5	1	1	2	0	0	1	1	1	0
consistency	8	6	5	6	3	5	7	9	7	3
sum	15	13	10	11	5	6	15	20	10	4
average	10.8					11				

Table V: Found (and accepted) defects per participant of fourth experiment (four hour review).

	Group 1					Group 2				
setting	basic					best practices				
participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
completeness	3	3	0	6	0	3	9	13	5	0
correctness	2	0	2	0	0	0	1	2	1	0
consistency	5	1	7	3	2	3	4	6	3	5
sum	10	4	9	9	2	6	14	21	9	5
average	6.8					11				

Concerning Q1 we got the results that with some advantages and disadvantages for some defect types the topic landscape can replace the normal review. Unfortunately, there is no evidence for enough benefit to do a complete replacement. This is not surprising, since for many defect types, like unambiguity, correctness, testability, atomicity, a linear reading of the documents and searching for defects is the best choice. On the opposite, we got evidence from the experiment that a linear review with a mechanism, where you stop at one requirement and search the whole document for a particular topic related to the actual requirement helps to find certain defect types, like inconsistencies, easier.

We are aware that this evidence is not really strong, but this is due to the experiment environment with students not surprising. As we have already discussed before and will discuss in the threats to validity, this experiment setting leads to certain disadvantages. One of the main problem is that the topic landscape approach is developed for far bigger document sets with many documents. In the experiment data sets, which are rather small and contain only four documents to not overburden the students, it is still possible for the students to remember most of the requirements by linear

reading and find inconsistencies and incompleteness between them. In a real industrial setting with up to 3.000 pages this is not possible for the reviewer. This circumstance could not be simulated in the experiment.

Nevertheless, we believe, that the current experiment justifies further experiments with the topic landscape, especially in industrial environments.

B. Applicability in Industrial Practice

Questions Q2 and Q3 addressed the applicability of our approach in industrial practice: To use the topic landscape in real world projects, it is necessary that the classification of requirements to topics is done automatically. In a previous work [4] we showed that such an automatic classification is possible with ReCaRe with sufficient results, at least in comparison to other research concerning text classification and information retrieval. This evaluation now indicates that the automatic classification is also sufficient for review tasks. Although this evidence is biased by the above discussed problems of the comparability between the experiment and industrial environment, we believe that ReCaRe can be applied in industry. Especially, because we also got the result that the topic classification can be strongly improved with only minor manual changes.

C. Threats to Validity

In this section, the threats to validity are discussed. Therefore, we use the classification of validity aspects from Runeson et al. [13] on construction validity, internal validity, external validity, and reliability.

Construction & Internal Validity. One obvious threat is the manual classification. It is questionable - there is no unique classification and it is reviewer dependent - which requirements must be considered as belonging to a topic. We mitigated this threat, by introducing a process using Cohen's Cappa for the manual classification tasks (See Section II-A) for details

Furthermore, the students' efficiency and performance depend on each individual within a broad range. Some people perform 10–20 times better than others. This discrepancy can be best equalized with many participants. According to our experience, it is difficult to find a large amount of students for a voluntary course.

Also, the participants might have different experiences with the review process itself. We preceded the experiment with a short introduction phase to face this problem. The students had to pass a review of a small example specification document in order to get used to ReCaRe, our quality model and the review process itself.

Concerning also the experience, during the empirical study, the students certainly gain experience during the multiple review tasks in the experiment. In order to avoid a too large learning effect, we prepared three different data sets and exchanged these sets among the students after each phase. Nevertheless, there will be a small learning effect, because e.g. all three data sets are within the same domain and describe a component in a similar structure.

Another point, is the missing quality information about the data sets, since it is not possible to avoid mistakes in such large textual documents during the writing phase. And indeed, the students found a lot of defects not injected by us. We do not see, how to avoid this aspect when using real documents without spending a tremendous amount of time in a pre quality check. A similar problem is, that we can only do a subjective estimation, whether all three data sets and the injected defects are equal difficult and complex. This circumstance results in a more realistic study but also introduces some uncontrollable variables.

Last, ReCaRe is still considered as a prototype. Therefore, some helpful additional features are missing, for example a search function. This resulted in some motivation losses for the participants, but didn't considerably bias the results.

External Validity. There are limitations in the transferability of our results on German, natural language specifications drawn from the Mercedes-Benz passenger car development to specifications from other industries because of different specification structures, the content and complexity of the specifications, and other company specific factors.

Reliability. The topic landscape and the manual classification is person dependent. So a replication of the experiment would result in a slightly different number of topics and classification of requirements to them.

V. Related Work

In this section, we discuss research on reviews and approaches to support or improve the review process. Afterwards, we present existing research on the classification of requirements and talk about the different use cases and benefits to do these classifications.

The initial work about reviews was done by Fagan [14]. Since then, there have been many further developments of the review process. Aurum et al. [15] give an overview of the progress in the review process from Fagan's initial work until 2002. Gilb and Graham [16] provide a thorough discussion about reviews, including case studies from organizations using reviews in practice.

As stated before, the benefit of the review of natural language specifications becomes limited because of the increasing size and complexity of the documents to be checked. To overcome these obstacles, much research has been done until now to automatically search for special kinds of defects in the natural language specification or to support the review process with preliminary analyses. Some examples are listed below:

The ARM tool by Wilson et al. [17] automatically measures and analyzes indicators to predict the quality of the documents. These indicators are separated in categories for

individual specification statements (e.g. imperatives, directives, weak phrases) and categories for the entire specification (e.g. size, readability, specification depth).

The tool QuARS by Gnesi et al. [18] automatically detects linguistic defects like ambiguities, using an initial parsing of the requirements.

The tool CARL from Gervasi and Zowghi [19] automatically identifies and analyzes inconsistencies of specifications in controlled natural language. This is done by automatic parsing of natural language sentences into propositional logic formulae. The approach is limited by the controlled language and the set of defined consistency rules.

Similar to Gervasi and Zowghi, Moser et al. [20] automatically inspect requirements with rule-based checks for inconsistencies. Unfortunately, in their approach the specifications must be written in controlled natural language.

The following research focuses on the classification of requirements for multiple purposes:

Moser et al. [20] are using a classification of requirements as an intermediate step during the check of requirements with regard to inconsistencies.

Gnesi et al. [18] create a categorization of requirements to topics as a byproduct during the detection of linguistic defects.

Hussain et al. [21] developed the tool LASR that supports users in annotation tasks. To do this, LASR automatically classifies requirements to certain annotations and presents the candidates to the user for the final decision.

Song and Hwong [22] report about their experiences with manual categorizations of requirements in a contract-based system integration project. The contract for this project contains over 4,000 clauses, which are mostly contract requirements. They state the need of categorization of these requirements for the following purposes: The identification of requirements of different kinds (e.g. technical requirements) and to have specific guidelines for developing and analyzing these requirement types. The identification of non-functional requirements for architectural decisions and to identify the needed equipment, its quantity and permitted suppliers. To identify dependencies among requirements, especially to detect risks and for scheduling needs during the project.

In addition, Knauss et al. [11] report the importance for many specifications nowadays, to classify the security-related requirements early in the project, to prevent substantial security problems later. Therefore, they automatically classify security relevant requirements in specifications with Naive Bayesian Classifiers. They got the results that using the same specification as training and testing leads to satisfying results. The problem is getting sufficient training data for a new specification from other/older specifications in order to get useful results in practice.

One probably feasible way to get sufficient training data is the approach of Ko et al. [23]. They use Naive Bayesian Classifiers to automatically classify requirements to topics, but they also automatically create the training data to do that. The idea is to define each topic with a few keywords and then use a cluster algorithm for each topic to get resulting requirements, which are then used to train the classifiers. The evaluation results of this approach are promising, but the evaluation was only done by small English and Korean specifications (less than 200 sentences).

VI. Summary & Future Work

During this work, we showed the essential problem of ensuring the quality in large and complex natural language requirement specifications with reviews and, with the topic landscape, we presented (realized in ReCaRe) a promising solution to mitigate this problem. ReCaRe automatically classifies requirements to topics and therefore supports the reviewer by finding defects, especially completeness and consistency defects.

In an experiment with ten students investigating three data sets originating from Mercedes-Benz, we evaluated first, if the topic landscape has benefits for reviewing tasks. There, we got evidence that the topic landscape is an useful aid to normal review activities. Further, we investigated the usability of this approach in practice: Is the performance of an automatic classification, as commonly derived with ReCaRe, acceptable for reviewers' tasks? This is a valid question, because an automatic classification does not necessary assign each relevant requirement to each topic and will always assign additionally not relevant requirements. We also got positive results to this thesis.

Consequently, we got enough positive indicators from this work to justify a pilot study with ReCaRe in an industrial environment and will therefore conduct an experiment in cooperation with Mercedes-Benz with developers reviewing real specification in the near future.

References

[1] L. Mich, M. Franch, and I. Novi, "Market research for requirements analysis using linguistic tools," *Requirements Engineering*, vol. 9, no. 2, pp. 40–56, 2004.

[2] F. Houdek, "Challenges in Automotive Requirements Engineering," *Industrial Presentations by REFSQ 2010, Essen*.

[3] D. Ott, "Defects in natural language requirement specifications at mercedes-benz: An investigation using a combination of legacy data and expert opinion," in *Requirements Engineering Conference (RE), 2012 20th IEEE International*. IEEE, 2012, pp. 291–296.

[4] ——, "Automatic requirement categorization of large natural language specifications at mercedes-benz for review improvements," in *Requirements Engineering: Foundation for Software Quality*, 2013.

[5] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[6] V. Hollink, J. Kamps, C. Monz, and M. De Rijke, "Monolingual document retrieval for european languages," *Information retrieval*, vol. 7, no. 1, pp. 33–52, 2004.

[7] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*, 3rd, Ed. Morgan Kaufmann, 2012.

[8] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York., 1999, vol. 463.

[9] D. Ott and A. Raschke, "Review improvement by requirements classification at mercedes-benz: Limits of empirical studies in educational environments," in *Empirical Requirements Engineering (EmpiRE), 2012 IEEE Second International Workshop on*. IEEE, 2012, pp. 1–8.

[10] J. Carletta, "Squibs and discussions assessing agreement on classification tasks: The kappa statistic," *Computational linguistics*, vol. 22, no. 2, pp. 249–254, 1996.

[11] E. Knauss, S. Houmb, K. Schneider, S. Islam, and J. Jürjens, "Supporting requirements engineers in recognising security issues," *Requirements Engineering: Foundation for Software Quality*, pp. 4–18, 2011.

[12] S. Wang and C. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," *ACL (2)*, pp. 90 – 94, 2012.

[13] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[14] M. Fagan, "Design and code inspections to reduce errors in program development," *IBM Journal of Research and Development*, vol. 15, no. 3, p. 182, 1976.

[15] A. Aurum, H. Petersson, and C. Wohlin, "State-of-the-art: Software Inspections after 25 Years," *Software Testing, Verification and Reliability*, vol. 12, no. 3, pp. 133–154, 2002.

[16] T. Gilb and D. Graham, *Software Inspection*, S. Finzi, Ed. Addison-Wesley, 1994.

[17] W. Wilson, L. Rosenberg, and L. Hyatt, "Automated analysis of requirement specifications," in *Proceedings of the 19th International Conference on Software Engineering (ICSE '97)*. IEEE, 1997, pp. 161–171.

[18] S. Gnesi, G. Lami, G. Trentanni, F. Fabbrini, and M. Fusani, "An automatic tool for the analysis of natural language requirements," *International Journal of Computer Systems Science & Engineering*, vol. 20, no. 1, pp. 53–62, 2005.

[19] V. Gervasi and D. Zowghi, "Reasoning about inconsistencies in natural language requirements," *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 3, pp. 277–330, 2005.

[20] T. Moser, D. Winkler, M. Heindl, and S. Biffl, "Requirements Management with Semantic Technology: An Empirical Study on Automated Requirements Categorization and Conflict Analysis," in *Advanced Information Systems Engineering*. Springer, 2011.

[21] I. Hussain, O. Ormandjieva, and L. Kosseim, "Lasr: A tool for large scale annotation of software requirements," in *Empirical Requirements Engineering (EmpiRE), 2012 IEEE Second International Workshop on*. IEEE, 2012, pp. 57–60.

[22] X. Song and B. Hwong, "Categorizing requirements for a contract-based system integration project," in *Requirements Engineering Conference (RE), 2012 20th IEEE International*. IEEE, 2012, pp. 279–284.

[23] Y. Ko, S. Park, J. Seo, and S. Choi, "Using classification techniques for informal requirements in the requirements analysis-supporting system," *Information and Software Technology*, vol. 49, pp. 1128–1140, 2007.

Liste der bisher erschienenen Ulmer Informatik-Berichte

Einige davon sind per FTP von `ftp.informatik.uni-ulm.de` erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm

Some of them are available by FTP from `ftp.informatik.uni-ulm.de`
Reports marked with * are out of print

91-01 *Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe*
Instance Complexity

*91-02** *K. Gladitz, H. Fassbender, H. Vogler*
Compiler-Based Implementation of Syntax-Directed Functional Programming

*91-03** *Alfons Geser*
Relative Termination

*91-04** *J. Köbler, U. Schöning, J. Toran*
Graph Isomorphism is low for PP

91-05 *Johannes Köbler, Thomas Thierauf*
Complexity Restricted Advice Functions

*91-06** *Uwe Schöning*
Recent Highlights in Structural Complexity Theory

*91-07** *F. Green, J. Köbler, J. Toran*
The Power of Middle Bit

*91-08** *V.Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf*
Reductions for Sets of Low Information Content

*92-01** *Vikraman Arvind, Johannes Köbler, Martin Mundhenk*
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

*92-02** *Thomas Noll, Heiko Vogler*
Top-down Parsing with Simulataneous Evaluation of Noncircular Attribute Grammars

92-03 *Fakultät für Informatik*
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

*92-04** *V. Arvind, J. Köbler, M. Mundhenk*
Lowness and the Complexity of Sparse and Tally Descriptions

*92-05** *Johannes Köbler*
Locating P/poly Optimally in the Extended Low Hierarchy

*92-06** *Armin Kühnemann, Heiko Vogler*
Synthesized and inherited functions -a new computational model for syntax-directed semantics

*92-07** *Heinz Fassbender, Heiko Vogler*
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

*92-08** *Uwe Schöning*
On Random Reductions from Sparse Sets to Tally Sets

*92-09** *Hermann von Hasseln, Laura Martignon*
Consistency in Stochastic Network

92-10 *Michael Schmitt*
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11 *Johannes Köbler, Seinosuke Toda*
On the Power of Generalized MOD-Classes

92-12 *V. Arvind, J. Köbler, M. Mundhenk*
Reliable Reductions, High Sets and Low Sets

92-13 *Alfons Geser*
On a monotonic semantic path ordering

*92-14** *Joost Engelfriet, Heiko Vogler*
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01 *Alfred Lupper, Konrad Froitzheim*
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*
The COCOON Object Model

93-03 *Thomas Thierauf, Seinosuke Toda, Osamu Watanabe*
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04 *Jin-Yi Cai, Frederic Green, Thomas Thierauf*
On the Correlation of Symmetric Functions

93-05 *K.Kuhn, M.Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam*
A Conceptual Approach to an Open Hospital Information System

93-06 *Klaus Gaßner*
Rechnerunterstützung für die konzeptuelle Modellierung

93-07 *Ullrich Keßler, Peter Dadam*
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01 *Michael Schmitt*
On the Complexity of Consistency Problems for Neurons with Binary Weights

94-02 *Armin Kühnemann, Heiko Vogler*
A Pumping Lemma for Output Languages of Attributed Tree Transducers

94-03 *Harry Buhrman, Jim Kadin, Thomas Thierauf*
On Functions Computable with Nonadaptive Queries to NP

94-04 *Heinz Faßbender, Heiko Vogler, Andrea Wedel*
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers

94-05 *V. Arvind, J. Köbler, R. Schuler*
On Helping and Interactive Proof Systems

94-06 *Christian Kalus, Peter Dadam*
Incorporating record subtyping into a relational data model

94-07 *Markus Tresch, Marc H. Scholl*
A Classification of Multi-Database Languages

94-08 *Friedrich von Henke, Harald Rueß*
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge

94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*
Construction and Deduction Methods for the Formal Development of Software

94-10 *Axel Dold*
Formalisierung schematischer Algorithmen

94-11 *Johannes Köbler, Osamu Watanabe*
New Collapse Consequences of NP Having Small Circuits

94-12 *Rainer Schuler*
On Average Polynomial Time

94-13 *Rainer Schuler, Osamu Watanabe*
Towards Average-Case Complexity Analysis of NP Optimization Problems

94-14 *Wolfram Schulte, Ton Vullinghs*
Linking Reactive Software to the X-Window System

94-15 *Alfred Lupper*
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

94-16 *Robert Regn*
Verteilte Unix-Betriebssysteme

94-17 *Helmuth Partsch*
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming

94-18 *Helmuth Partsch*
Transformational Development of Data-Parallel Algorithms: an Example

95-01 *Oleg Verbitsky*
On the Largest Common Subgraph Problem

95-02 *Uwe Schöning*
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

95-03 *Harry Buhrman,Thomas Thierauf*
The Complexity of Generating and Checking Proofs of Membership

95-04 *Rainer Schuler, Tomoyuki Yamakami*
Structural Average Case Complexity

95-05 *Klaus Achatz, Wolfram Schulte*
Architecture Indepentent Massive Parallelization of Divide-And-Conquer Algorithms

95-06 *Christoph Karg, Rainer Schuler*
Structure in Average Case Complexity

95-07 *P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe*
ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen

95-08 *Jürgen Kehrer, Peter Schulthess*
Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik

95-09 *Hans-Jörg Burtschick, Wolfgang Lindner*
On Sets Turing Reducible to P-Selective Sets

95-10 *Boris Hartmann*
Berücksichtigung lokaler Randbedingung bei globaler Zieloptimierung mit neuronalen Netzen am Beispiel Truck Backer-Upper

95-11 *Thomas Beuter, Peter Dadam*:
Prinzipien der Replikationskontrolle in verteilten Systemen

95-12 *Klaus Achatz, Wolfram Schulte*
Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists

95-13 *Andrea Mößle, Heiko Vogler*
Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes

95-14 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
A Generic Specification for Verifying Peephole Optimizations

96-01 *Ercüment Canver, Jan-Tecker Gayen, Adam Moik*
Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE

96-02 *Bernhard Nebel*
Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class

96-03 *Ton Vullinghs, Wolfram Schulte, Thilo Schwinn*
An Introduction to TkGofer

96-04 *Thomas Beuter, Peter Dadam*
Anwendungsspezifische Anforderungen an Workflow-Mangement-Systeme am Beispiel der Domäne Concurrent-Engineering

96-05 *Gerhard Schellhorn, Wolfgang Ahrendt*
Verification of a Prolog Compiler - First Steps with KIV

96-06 *Manindra Agrawal, Thomas Thierauf*
Satisfiability Problems

96-07 *Vikraman Arvind, Jacobo Torán*
A nonadaptive NC Checker for Permutation Group Intersection

96-08 *David Cyrluk, Oliver Möller, Harald Rueß*
An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and Extraction

96-09 *Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte*
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT–Ansätzen

96-10 *Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Formalizing Fixed-Point Theory in PVS

96-11 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Mechanized Semantics of Simple Imperative Programming Constructs

96-12 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Generic Compilation Schemes for Simple Programming Constructs

96-13 *Klaus Achatz, Helmuth Partsch*
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its Applications and Variants

97-01 *Jochen Messner*
Pattern Matching in Trace Monoids

97-02 *Wolfgang Lindner, Rainer Schuler*
A Small Span Theorem within P

97-03 *Thomas Bauer, Peter Dadam*
A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration

97-04 *Christian Heinlein, Peter Dadam*
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies

97-05 *Vikraman Arvind, Johannes Köbler*
On Pseudorandomness and Resource-Bounded Measure

97-06 *Gerhard Partsch*
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den digitalen Mobilfunkstandard DECT

97-07 *Manfred Reichert, Peter Dadam*
$ADEPT_{flex}$ - Supporting Dynamic Changes of Workflows Without Loosing Control

97-08 *Hans Braxmeier, Dietmar Ernst, Andrea Mößle, Heiko Vogler*
The Project NoName - A functional programming language with its development environment

97-09 *Christian Heinlein*
Grundlagen von Interaktionsausdrücken

97-10 *Christian Heinlein*
Graphische Repräsentation von Interaktionsausdrücken

97-11 *Christian Heinlein*
Sprachtheoretische Semantik von Interaktionsausdrücken

97-12 *Gerhard Schellhorn, Wolfgang Reif*
Proving Properties of Finite Enumerations: A Problem Set for Automated Theorem Provers

97-13 *Dietmar Ernst, Frank Houdek, Wolfram Schulte, Thilo Schwinn*
Experimenteller Vergleich statischer und dynamischer Softwareprüfung für eingebettete Systeme

97-14 *Wolfgang Reif, Gerhard Schellhorn*
Theorem Proving in Large Theories

97-15 *Thomas Wennekers*
Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen

97-16 *Peter Dadam, Klaus Kuhn, Manfred Reichert*
Clinical Workflows - The Killer Application for Process-oriented Information Systems?

97-17 *Mohammad Ali Livani, Jörg Kaiser*
EDF Consensus on CAN Bus Access in Dynamic Real-Time Applications

97-18 *Johannes Köbler,Rainer Schuler*
Using Efficient Average-Case Algorithms to Collapse Worst-Case Complexity Classes

98-01 *Daniela Damm, Lutz Claes, Friedrich W. von Henke, Alexander Seitz, Adelinde Uhrmacher, Steffen Wolf*
Ein fallbasiertes System für die Interpretation von Literatur zur Knochenheilung

98-02 *Thomas Bauer, Peter Dadam*
Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse

98-03 *Marko Luther, Martin Strecker*
A guided tour through *Typelab*

98-04 *Heiko Neumann, Luiz Pessoa*
Visual Filling-in and Surface Property Reconstruction

98-05 *Ercüment Canver*
Formal Verification of a Coordinated Atomic Action Based Design

98-06 *Andreas Küchler*
On the Correspondence between Neural Folding Architectures and Tree Automata

98-07 *Heiko Neumann, Thorsten Hansen, Luiz Pessoa*
Interaction of ON and OFF Pathways for Visual Contrast Measurement

98-08 *Thomas Wennekers*
Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons

98-09 *Thomas Bauer, Peter Dadam*
Variable Migration von Workflows in *ADEPT*

98-10 *Heiko Neumann, Wolfgang Sepp*
Recurrent V1 – V2 Interaction in Early Visual Boundary Processing

98-11 *Frank Houdek, Dietmar Ernst, Thilo Schwinn*
Prüfen von C–Code und Statmate/Matlab–Spezifikationen: Ein Experiment

98-12 *Gerhard Schellhorn*
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers

98-13 *Gerhard Schellhorn, Wolfgang Reif*
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers

98-14 *Mohammad Ali Livani*
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN

98-15 *Mohammad Ali Livani, Jörg Kaiser*
Predictable Atomic Multicast in the Controller Area Network (CAN)

99-01 *Susanne Boll, Wolfgang Klas, Utz Westermann*
A Comparison of Multimedia Document Models Concerning Advanced Requirements

99-02 *Thomas Bauer, Peter Dadam*
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation

99-03 *Uwe Schöning*
On the Complexity of Constraint Satisfaction

99-04 *Ercument Canver*
Model-Checking zur Analyse von Message Sequence Charts über Statecharts

99-05 *Johannes Köbler, Wolfgang Lindner, Rainer Schuler*
Derandomizing RP if Boolean Circuits are not Learnable

99-06 *Utz Westermann, Wolfgang Klas*
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets

99-07 *Peter Dadam, Manfred Reichert*
Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999, GI–Workshop Proceedings, Informatik ’99

99-08 *Vikraman Arvind, Johannes Köbler*
Graph Isomorphism is Low for ZPP^{NP} and other Lowness results

99-09 *Thomas Bauer, Peter Dadam*
Efficient Distributed Workflow Management Based on Variable Server Assignments

2000-02 *Thomas Bauer, Peter Dadam*
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT

2000-03 *Gregory Baratoff, Christian Toepfer, Heiko Neumann*
Combined space-variant maps for optical flow based navigation

2000-04 *Wolfgang Gehring*
Ein Rahmenwerk zur Einführung von Leistungspunktsystemen

2000-05 *Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel*
Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos

2000-06 *Wolfgang Reif, Gerhard Schellhorn, Andreas Thums*
Fehlersuche in Formalen Spezifikationen

2000-07 *Gerhard Schellhorn, Wolfgang Reif (eds.)*
FM-Tools 2000: The 4th Workshop on Tools for System Design and Verification

2000-08 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Effiziente Durchführung von Prozessmigrationen in verteilten Workflow-Management-Systemen

2000-09 *Thomas Bauer, Peter Dadam*
Vermeidung von Überlastsituationen durch Replikation von Workflow-Servern in ADEPT

2000-10 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Adaptives und verteiltes Workflow-Management

2000-11 *Christian Heinlein*
Workflow and Process Synchronization with Interaction Expressions and Graphs

2001-01 *Hubert Hug, Rainer Schuler*
DNA-based parallel computation of simple arithmetic

2001-02 *Friedhelm Schwenker, Hans A. Kestler, Günther Palm*
3-D Visual Object Classification with Hierarchical Radial Basis Function Networks

2001-03 *Hans A. Kestler, Friedhelm Schwenker, Günther Palm*
RBF network classification of ECGs as a potential marker for sudden cardiac death

2001-04 *Christian Dietrich, Friedhelm Schwenker, Klaus Riede, Günther Palm*
Classification of Bioacoustic Time Series Utilizing Pulse Detection, Time and Frequency Features and Data Fusion

2002-01 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata

2002-02 *Walter Guttmann*
Deriving an Applicative Heapsort Algorithm

2002-03 *Axel Dold, Friedrich W. von Henke, Vincent Vialard, Wolfgang Goerigk*
A Mechanically Verified Compiling Specification for a Realistic Compiler

2003-01 *Manfred Reichert, Stefanie Rinderle, Peter Dadam*
A Formal Framework for Workflow Type and Instance Changes Under Correctness Checks

2003-02 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Supporting Workflow Schema Evolution By Efficient Compliance Checks

2003-03 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values

2003-04 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
On Dealing With Semantically Conflicting Business Process Changes.

2003-05 *Christian Heinlein*
Dynamic Class Methods in Java

2003-06 *Christian Heinlein*
Vertical, Horizontal, and Behavioural Extensibility of Software Systems

2003-07 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values
(Corrected Version)

2003-08 *Changling Liu, Jörg Kaiser*
Survey of Mobile Ad Hoc Network Routing Protocols)

2004-01 *Thom Frühwirth, Marc Meister (eds.)*
First Workshop on Constraint Handling Rules

2004-02 *Christian Heinlein*
Concept and Implementation of C+++, an Extension of C++ to Support User-Defined Operator Symbols and Control Structures

2004-03 *Susanne Biundo, Thom Frühwirth, Günther Palm(eds.)*
Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence

2005-01 *Armin Wolf, Thom Frühwirth, Marc Meister (eds.)*
19th Workshop on (Constraint) Logic Programming

2005-02 *Wolfgang Lindner (Hg.), Universität Ulm , Christopher Wolf (Hg.) KU Leuven*
2. Krypto-Tag – Workshop über Kryptographie, Universität Ulm

2005-03 *Walter Guttmann, Markus Maucher*
Constrained Ordering

2006-01 *Stefan Sarstedt*
Model-Driven Development with ACTIVECHARTS, Tutorial

2006-02 *Alexander Raschke, Ramin Tavakoli Kolagari*
Ein experimenteller Vergleich zwischen einer plan-getriebenen und einer leichtgewichtigen Entwicklungsmethode zur Spezifikation von eingebetteten Systemen

2006-03 *Jens Kohlmeyer, Alexander Raschke, Ramin Tavakoli Kolagari*
Eine qualitative Untersuchung zur Produktlinien-Integration über Organisationsgrenzen hinweg

2006-04 *Thorsten Liebig*
Reasoning with OWL - System Support and Insights –

2008-01 *H.A. Kestler, J. Messner, A. Müller, R. Schuler*
On the complexity of intersecting multiple circles for graphical display

2008-02 *Manfred Reichert, Peter Dadam, Martin Jurisch,l Ulrich Kreher, Kevin Göser, Markus Lauer*
Architectural Design of Flexible Process Management Technology

2008-03 *Frank Raiser*
Semi-Automatic Generation of CHR Solvers from Global Constraint Automata

2008-04 *Ramin Tavakoli Kolagari, Alexander Raschke, Matthias Schneiderhan, Ian Alexander*
Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse

2008-05 *Markus Kalb, Claudia Dittrich, Peter Dadam*
Support of Relationships Among Moving Objects on Networks

2008-06 *Matthias Frank, Frank Kargl, Burkhard Stiller (Hg.)*
WMAN 2008 – KuVS Fachgespräch über Mobile Ad-hoc Netzwerke

2008-07 *M. Maucher, U. Schöning, H.A. Kestler*
An empirical assessment of local and population based search methods with different degrees of pseudorandomness

2008-08 *Henning Wunderlich*
Covers have structure

2008-09 *Karl-Heinz Niggl, Henning Wunderlich*
Implicit characterization of FPTIME and NC revisited

2008-10 *Henning Wunderlich*
On span-P^{cc} and related classes in structural communication complexity

2008-11 *M. Maucher, U. Schöning, H.A. Kestler*
On the different notions of pseudorandomness

2008-12 *Henning Wunderlich*
On Toda's Theorem in structural communication complexity

2008-13 *Manfred Reichert, Peter Dadam*
Realizing Adaptive Process-aware Information Systems with ADEPT2

2009-01 *Peter Dadam, Manfred Reichert*
The ADEPT Project: A Decade of Research and Development for Robust and Fexible Process Support
Challenges and Achievements

2009-02 *Peter Dadam, Manfred Reichert, Stefanie Rinderle-Ma, Kevin Göser, Ulrich Kreher, Martin Jurisch*
Von ADEPT zur AristaFlow® BPM Suite – Eine Vision wird Realität "Correctness by Construction" und flexible, robuste Ausführung von Unternehmensprozessen

2009-03 *Alena Hallerbach, Thomas Bauer, Manfred Reichert*
Correct Configuration of Process Variants in Provop

2009-04 *Martin Bader*
On Reversal and Transposition Medians

2009-05 *Barbara Weber, Andreas Lanz, Manfred Reichert*
Time Patterns for Process-aware Information Systems: A Pattern-based Analysis

2009-06 *Stefanie Rinderle-Ma, Manfred Reichert*
Adjustment Strategies for Non-Compliant Process Instances

2009-07 *H.A. Kestler, B. Lausen, H. Binder H.-P. Klenk. F. Leisch, M. Schmid*
Statistical Computing 2009 – Abstracts der 41. Arbeitstagung

2009-08 *Ulrich Kreher, Manfred Reichert, Stefanie Rinderle-Ma, Peter Dadam*
Effiziente Repräsentation von Vorlagen- und Instanzdaten in Prozess-Management-Systemen

2009-09 *Dammertz, Holger, Alexander Keller, Hendrik P.A. Lensch*
Progressive Point-Light-Based Global Illumination

2009-10 *Dao Zhou, Christoph Müssel, Ludwig Lausser, Martin Hopfensitz, Michael Kühl, Hans A. Kestler*
Boolean networks for modeling and analysis of gene regulation

2009-11 *J. Hanika, H.P.A. Lensch, A. Keller*
Two-Level Ray Tracing with Recordering for Highly Complex Scenes

2009-12 *Stephan Buchwald, Thomas Bauer, Manfred Reichert*
Durchgängige Modellierung von Geschäftsprozessen durch Einführung eines Abbildungsmodells: Ansätze, Konzepte, Notationen

2010-01 *Hariolf Betz, Frank Raiser, Thom Frühwirth*
A Complete and Terminating Execution Model for Constraint Handling Rules

2010-02 *Ulrich Kreher, Manfred Reichert*
Speichereffiziente Repräsentation instanzspezifischer Änderungen in Prozess-Management-Systemen

2010-03 *Patrick Frey*
Case Study: Engine Control Application

2010-04 *Matthias Lohrmann und Manfred Reichert*
Basic Considerations on Business Process Quality

2010-05 *HA Kestler, H Binder, B Lausen, H-P Klenk, M Schmid, F Leisch (eds):*
Statistical Computing 2010 - Abstracts der 42. Arbeitstagung

2010-06 *Vera Künzle, Barbara Weber, Manfred Reichert*
Object-aware Business Processes: Properties, Requirements, Existing Approaches

2011-01 *Stephan Buchwald, Thomas Bauer, Manfred Reichert*
Flexibilisierung Service-orientierter Architekturen

2011-02 *Johannes Hanika, Holger Dammertz, Hendrik Lensch*
Edge-Optimized À-Trous Wavelets for Local Contrast Enhancement with Robust Denoising

2011-03 *Stefanie Kaiser, Manfred Reichert*
Datenflussvarianten in Prozessmodellen: Szenarien, Herausforderungen, Ansätze

2011-04 *Hans A. Kestler, Harald Binder, Matthias Schmid, Friedrich Leisch, Johann M. Kraus (eds):*
Statistical Computing 2011 - Abstracts der 43. Arbeitstagung

2011-05 *Vera Künzle, Manfred Reichert*
PHILharmonicFlows: Research and Design Methodology

2011-06 *David Knuplesch, Manfred Reichert*
Ensuring Business Process Compliance Along the Process Life Cycle

2011-07 *Marcel Dausend*
Towards a UML Profile on Formal Semantics for Modeling Multimodal Interactive Systems

2011-08 *Dominik Gessenharter*
Model-Driven Software Development with ACTIVECHARTS - A Case Study

2012-01 *Andreas Steigmiller, Thorsten Liebig, Birte Glimm*
Extended Caching, Backjumping and Merging for Expressive Description Logics

2012-02 *Hans A. Kestler, Harald Binder, Matthias Schmid, Johann M. Kraus (eds):*
Statistical Computing 2012 - Abstracts der 44. Arbeitstagung

2012-03 *Felix Schüssel, Frank Honold, Michael Weber*
Influencing Factors on Multimodal Interaction at Selection Tasks

2012-04 *Jens Kolb, Paul Hübner, Manfred Reichert*
Model-Driven User Interface Generation and Adaption in Process-Aware Information Systems

2012-05 *Matthias Lohrmann, Manfred Reichert*
Formalizing Concepts for Efficacy-aware Business Process Modeling

2012-06 *David Knuplesch, Rüdiger Pryss, Manfred Reichert*
A Formal Framework for Data-Aware Process Interaction Models

2012-07 *Clara Ayora, Victoria Torres, Barbara Weber, Manfred Reichert, Vicente Pelechano*
Dealing with Variability in Process-Aware Information Systems: Language Requirements, Features, and Existing Proposals

2013-01 *Frank Kargl*
Abstract Proceedings of the 7th Workshop on Wireless and Mobile Ad-Hoc Networks (WMAN 2013)

2013-02 *Andreas Lanz, Manfred Reichert, Barbara Weber*
A Formal Semantics of Time Patterns for Process-aware Information Systems

2013-03 *Matthias Lohrmann, Manfred Reichert*
Demonstrating the Effectiveness of Process Improvement Patterns with Mining Results

2013-04 *Semra Catalkaya, David Knuplesch, Manfred Reichert*
Bringing More Semantics to XOR-Split Gateways in Business Process Models Based on Decision Rules

2013-05 *David Knuplesch, Manfred Reichert, Linh Thao Ly, Akhil Kumar, Stefanie Rinderle-Ma*
On the Formal Semantics of the Extended Compliance Rule Graph

2013-06 *Andreas Steigmiller, Birte Glimm*
Nominal Schema Absorption

2013-07 *Hans A. Kestler, Matthias Schmid, Florian Schmid, Dr. Markus Maucher, Johann M. Kraus (eds)*
Statistical Computing 2013 - Abstracts der 45. Arbeitstagung

2013-08 *Daniel Ott, Alexander Raschke*
Evaluating Benefits of Requirement Categorization in Natural Language Specifications for Review Improvements

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm