ASADO - The Analysis and Structuring of Aviation Documents

Final Report of a
Study Project held at the
Institute of Cognitive Science
University of Osnabrück
and
Institute of Applied Linguistics
University of Hildesheim

Dr. Helmar Gust, Prof. Dr. Christa Hauenschild, Dr. Petra Ludewig, Martin Bleichner, Eugenie Giesbrecht, Eva-Maria Leicht, Dr. Sabine Möller, Wiebke Müller, Moritz Stefaner, Egon Stemle

October 14, 2006

Contents

1 Introduction **6**

1.1 One Year Study Projects 6

1.2 The Global Task 7

1.2.1 Starting Point 8

1.2.2 Sketch of the ASADO Approach 10

2 Industrial Requirements **13**

2.1 Requirements Analysis 13

2.1.1 Introduction 13

2.1.2 Methodological Considerations of the Requirement Analysis 13

2.1.3 Undertaking of the Workshop 15

2.1.4 Results of the Requirement Analysis 15

2.2 Software ergonomics 20

2.2.1 Introduction 20

2.2.2 State of the Art 20

2.2.2.1 Usability Engineering 21

2.2.2.2 Iterative Inspection Method 22

2.2.2.3 Cognitive Walkthrough 23

2.2.2.4 Heuristic Expert Inspection 23

2.2.3 Application 24

2.2.3.1 Restriction of Criteria 24

2.2.3.2 Cognitive Walkthrough and Inspection 24

2.2.3.3 Feedback Costumer 25

2.2.3.4 Second Version of Ascent Capture 25

2.2.3.5 Industrial Psychological Point of View 25

2.2.3.6 Job Design . . . 26
2.2.3.7 Motivation . . . 26
2.2.4 Conclusion . . . 27

3 Document Processing and Representation 28
3.1 Detecting Text Structure . . . 28
3.1.1 Starting Point . . . 28
3.1.2 Characteristics of the Document Repository . . . 28
3.1.2.1 Textual Content per Document . . . 29
3.1.2.2 Textual Content per Page . . . 29
3.1.3 Preprocessing . . . 30
3.2 Linguistic Processing . . . 31
3.2.1 Language Identification . . . 32
3.2.2 Tokenization, Lemmatization, and Part-of-Speech Tagging 33
3.2.3 Chunking . . . 35
3.3 Terminology Extraction . . . 38
3.3.1 Introduction . . . 38
3.3.2 Constituent Analysis . . . 38
3.3.3 Extended Constituent Analysis . . . 40
3.3.4 Qualitative Evaluation of the Terminology . . . 42
3.3.5 Summary and Discussion . . . 44
3.4 Vector based Document Representation . . . 46
3.4.1 Construction of the Vector Representation . . . 46
3.4.1.1 Document Indexing . . . 47
3.4.1.2 Term Weighting . . . 47
3.4.2 Feature Reduction . . . 48
3.4.3 Putting it all together . . . 49

4 Analyzing and Presenting the Document Space 50
4.1 How Document Maps Can Help in Information Retrieval . . . 50
4.2 Clustering . . . 53
4.2.1 Introduction . . . 53
4.2.2 Procedure . . . 53
4.2.2.1 Prerequisites . . . 53

4.2.2.2 Clustering Methods 54
4.2.2.3 Algorithm and Parameter Settings 55
4.2.2.4 Hypotheses 57
4.2.2.5 Experiments and Evaluation 58
4.2.3 Perspectives 63
4.3 The ASADO visualization module 65
4.3.1 General interaction principles 65
4.3.2 The framework 66
4.3.3 Scatterplot 67
4.3.4 Map 68
4.3.5 Additional panels 68
4.4 Mapping techniques 70
4.5 Empirical results 71
4.6 Labeling 75

5 Summary and Conclusions **77**

References **85**

List of Figures

1.1 Global situational setting of the project 9
1.2 Architecture-like overview of the ASADO approach 10

2.1 Usability engineering 21
2.2 Iterative inspection method 22

4.1 Clustering 54
4.2 Bisecting K-Means 56
4.3 Document example 59
4.4 Average internal cluster similarities for different K 60
4.5 Average external cluster similarities for different K 61
4.6 Average internal cluster validity values for K=40 62
4.7 H2 for different document vector representations 63
4.8 A screenshot of the ASADO system 67
4.9 The scatterplot view 68
4.10 The map view 69
4.11 Slide control for coordinate mixture 69
4.12 Cluster labels and document tooltips are presented on mouse rollover 69
4.13 Test projections 72
4.14 Biplots of original vs. projected distances 74

Chapter 1

Introduction

1.1 One Year Study Projects

In order to assess the quality and the significance of the results presented in the following paper, it might be helpful to realize the context in which they were achieved: The underlying work was done in a one year study project carried out in close cooperation with five bachelor and two master students of cognitive science advised by two lectures of the Institute of Cognitive Science of the University of Osnabrück and with one Magister student of international information management advised by one professor of the Institute of Applied Linguistics of the University of Hildesheim.

One year study projects are a core type of teaching and learning within the master's[1] study program "Cognitive Science" (cf. (28) for further details). That is to say, study projects are part of the ongoing academic teaching. During twelve months six to ten students define a research task within a larger domain, plan it and carry out the required work and finally present their results in talks and articles. The global domain should preferably of interdisciplinary character and usually overlaps with research currently undertaken by institute members. This is why these study projects are usually supervised by two scientists. While supervisors define the global context, the students are required to determine the

[1] ASADO is the first study project for which most of the students belong to the bachelor and not yet to the master program of cognitive science.

actual task by themselves which is quite challenging. The task should be scientifically interesting and bear some practical relevance but it also has as well to be tractable within the given time.

Sometimes study projects address topics of industrial relevance or their members belong to locally distributed teams. That is, students of the University of Osnabrück cooperate with students of other universities. Both conditions apply to ASADO. This certainly makes working and learning somewhat harder but is, without doubt, also more exciting. In ASADO, the concrete task to be defined had to be fixed by the students of both cooperating universities in close discussion and agreement with their industrial partners, namely with Frithjof Weber (Airbus: knowledge management) and Katja Wilke (AAS Aviation Archive Service GmbH[1] : archiving).

Besides, administrative challenges had to be met. Signing a non-disclosure agreement, getting the set of documents analyzed[2], transferring money in order to cover the traveling costs incurred, as well as organizing a group discussion with Airbus members in order to identify user requirements made striking demands on the time of the (academic and industrial) leading members. Furthermore, it took some time to match both the needs and interests of the industrial partners and the research interest of the academic project members, since this step assumed that each group understood the core techniques, interests, potentials, and problems of the other one. However, after a period of six months, a good understanding seemed to have been reached between the industrial and academic partners involved. Thus, ideas for future cooperations were raised quite easily in the second half of the project.

1.2 The Global Task

Within the project ZAMIZ EDG (Zentralarchiv mit interaktivem Zugriff für Engineering-Dokumente Germany) at Airbus Deutschland GmbH about 30 million pages are to be archived electronically. These pages are printed engineering

[1] outside supplier company

[2] Unfortunately the given document selection turned out not to be optimally suitable for the clustering task addressed within ASADO.

documents produced during the development of different aircrafts in the last three decades. They contain texts, tables, schemes, technical drawings and a great number of sketches.

There are basically two different scenarios of how to use the archive. First, design and development engineers may want to search for documents as part of their daily routine work. Second, members of the "product safety" department need to search for relevant documents in case an aviation accident or incident has occurred and liability issues have to be dealt with. The latter group will typically search for documents under a very high pressure regarding time and correctness of the search result. Traditionally, search quality is measured as a combination of recall and precision. The demands on recall are that as many relevant documents as possible are found. The precision criterion means that there should be as few irrelevant documents displayed as possible. In the safety scenario recall is of paramount importance for correctness. Precision has an important impact on search time aspects.

Also, there is the field of knowledge management where intelligent and powerful tools are needed for managing the immense, heterogeneous and highly valuable knowledge resources spread over the whole Airbus Company. These resources have to be made easily and reliably accessible for many different user groups, especially for engineers. The global task the ASADO project decided to tackle was to support these archiving, knowledge managing and data searching processes and to connect these three domains of interest (see Figure 1.1) in a functional way. And, to some extent, it is a merit of this project, that the tight connections between the existing archiving project ZAMIZ EDG and more global tasks in the domain of knowledge management were unambiguously revealed.

1.2.1 Starting Point

Presently, the mentioned documents, which have previously been scanned and treated with OCR software for optical character recognition, are classified manually at **Airbus** by a reading team within the ZAMIZ EDG project. The classification task requires taking 12 to 15 cascading multi-valued decisions and there

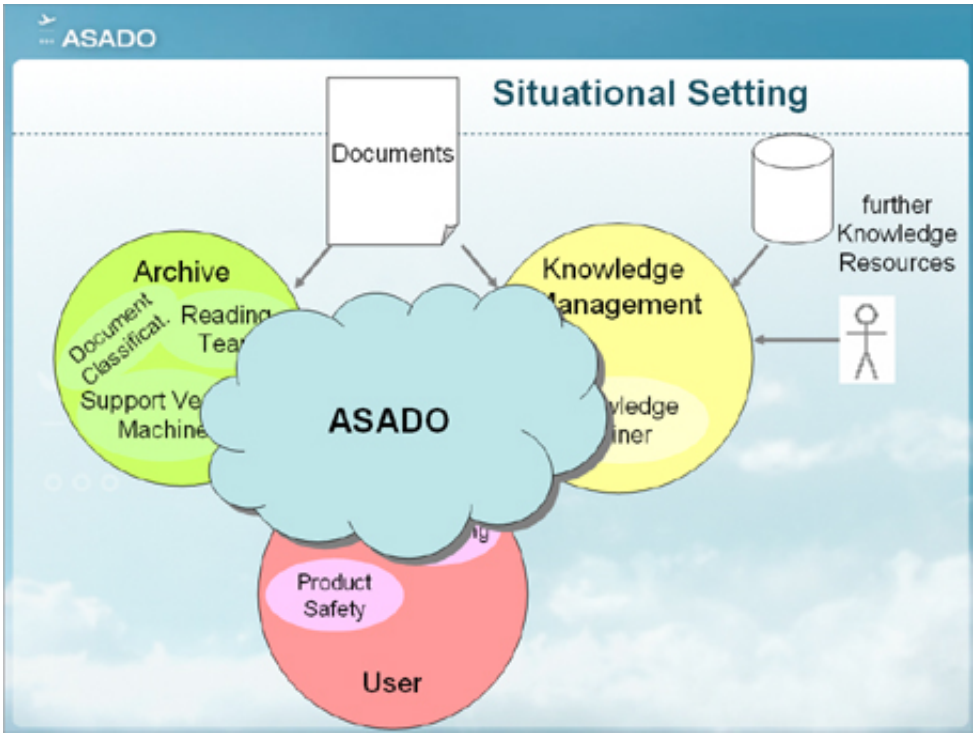


Figure 1.1: Global situational setting of the project

are ideas of using automatic classification techniques called "support vector machines" in order to automatically generate proposals for some of the document attributes that, later on, can be accepted or corrected by the members of the reading team.

Within the context of knowledge management a professional software called KnowledgeMiner provided by the company USU is evaluated by members of Airbus in order to assess whether this tool would be sufficiently helpful for knowledge management at Airbus Industries.

The **USU KnowledgeMiner** is a meta search engine that, among other things, permits to make use of (manually rated) former search results when making new but similar queries. To this end, it also tries to make use of terminological and ontological knowledge in order to optimize search results. Naturally, this background knowledge varies with respect to the given application scenario of the KnowledgeMiner. Therefore, in order to use the KnowledgeMiner for knowledge

management at Airbus, terminological and ontological knowledge about aviation will have to be compiled. Up to now, this can only be done manually by domain experts, an approach which, in fact, is both, time consuming and expensive.

For nearly all subproblems addressed within ASADO there are solutions available. Thus, in most cases, quite well **established techniques** are used within the implemented prototype. Sometimes, they are adapted or refined. The convincing aspect is the application of all these techniques to aviation documents which has a high synergy effect multiplying the positive impact of each single analysis and representation step.

1.2.2 Sketch of the ASADO Approach

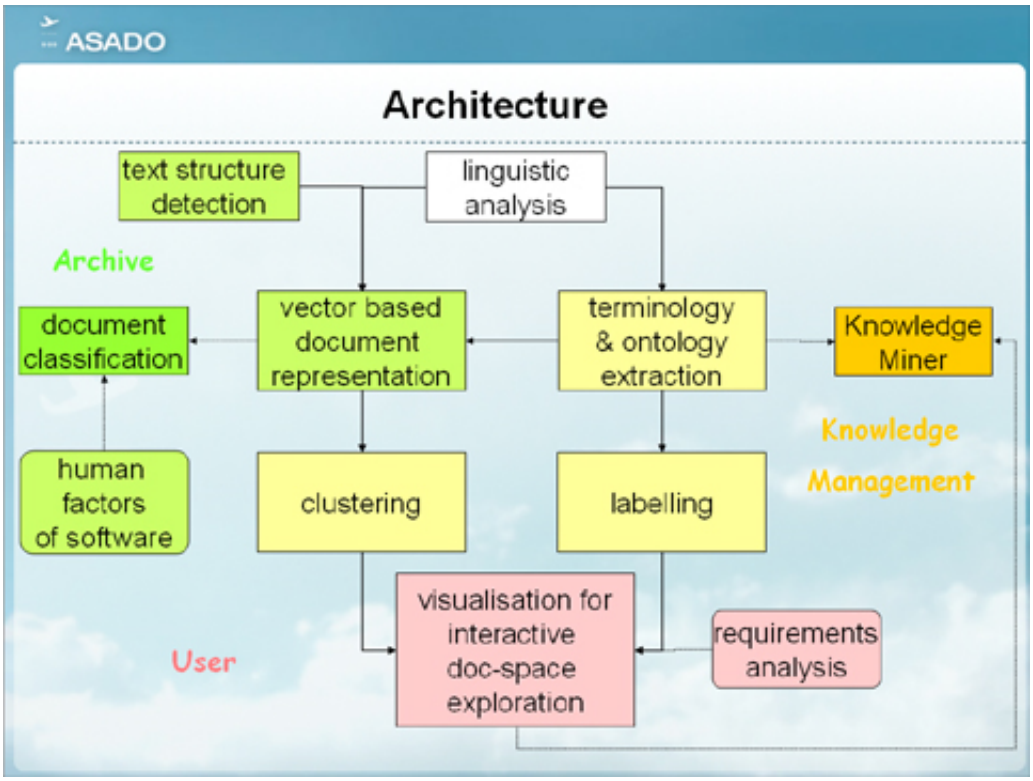


Figure 1.2: Architecture-like overview of the ASADO approach

Figure 1.2 gives an architecture-like overview of the ASADO approach. The modules are colored corresponding to their proximity to the three given domains

archive, knowledge management, and user interface. The modules "document classification" and "KnowledgeMiner" depicted slightly more dark in figure 2 are not part of the ASADO project but are supported by the results obtained within the project. Frames with rounded edges do also not represent implemented program modules but indicate important supplementary soft studies carried out by project members in close discussion with Airbus and AAS members. These studies try to identify **requirements of interfaces for the reading team and for the different types of end-users.**

One core aim of the project is to (linguistically) analyze and to represent aviation documents in such a way as to enable an interactive visualization of the document set that makes search of information more efficient and effective.

After transferring the documents into a tractable format, a series of well established **computational linguistic techniques** like tokenization, part of speech tagging, lemmatization and chunk parsing for noun phrase detection are applied to the given documents. In particular, the identified noun phrases serve as input for the following **terminology extraction**. Besides, heuristics are used in order to **identify the (logical) text structure** of documents since, for example, words written in large letters, might be parts of headings and, thus, be of more importance for content identification than words printed smaller.

All three results feed the **vector based document representation**. Very roughly speaking, each coordinate of a vector corresponds to a given term and their numerical value represents the frequency or importance of this item in the corresponding document. The core idea leads to the assumption that similar documents should roughly have similar vectors and vice versa.

Of course such vectors can also be used for **document classification** procedures like support vector machines. Nevertheless, the ASADO vector representations are primarily used as input to a **clustering algorithm** that tries to group those documents together that are, with respect to content, most similar to each other.

In a further step, the computed clusters of the document space should be **visualized** in order to facilitate document retrieval. The general idea is to represent documents as dots in a two-dimensional map, where the distance/proximity of two dots should indicate the semantic (dis)similarity of the corresponding documents.

Dot agglomerations thus indicate collections of semantically similar documents. The created clusters clusters should be **labeled** by their relevant terminology. For providing these labels vector representations and extracted terminology/ontology is resorted. The extracted terminology can also be used as input for the KnowledgeMiner and reduce manual terminology work. Besides, query results produced by the KnowledgeMiner could also be presented by an ASADO-like visualization tool.

In section 2 the results of the accompanying requirements studies carried out within the context of ASADO are presented. Section 3 treats primarily analysis and representation on the document level, whereas analysis and representation on the document set level is mainly described in section 4. Section 5 gives a summary of the ASADO results and indicates future perspectives.

Chapter 2

Industrial Requirements

2.1 Requirements Analysis

2.1.1 Introduction

In order to determine the users' needs regarding the visualization of ZAMIZ EDG search results it had been decided to undertake a requirement analysis within the framework of the ASADO project. This analysis was meant to answer the question in which use cases users would employ the currently available search functions. New insights were supposed to be gained for the improvement of search options of the prototype developed in the ASADO project. The analysis of use cases concentrated on the pre-defined user contexts "Product Safety" and "Engineering" which were already regarded to be the principle ones in the ZAMIZ EDG project.

2.1.2 Methodological Considerations of the Requirement Analysis

There are various methods to undertake a requirement analysis, e.g. questionnaires, interviews or group discussions. An important decision for our project was to employ a qualitative method instead of a quantitative one. Given the nature of the field to be investigated this decision seemed to be especially sensible. Only little detail is known about the use cases and therefore a questionnaire would

not have been an appropriate instrument. On the one hand, using a questionnaire would actually have been time saving for the interviewees but on the other hand the given questions would have restricted the variety of possible answers too much. This would have implied the risk of overlooking core aspects of the area of application, especially since the members of the ASADO project had rather restricted knowledge of the working or operating conditions at **Airbus** (16; 6) . A research method suitable for a qualitative requirement analysis is the qualitative interview. The range of qualitative interviews can be distinguished by different levels of standardization and by the interview style. There are differences between interviews with rigidly organized Q&A sequences and interviews with relatively freely organized communication about a topic with large amount of narrative elements. One particularly important aspect of an interview is the preparation of pre-formulated questions.

For the success of research planned by means of a requirement analysis it is vital to integrate questions related to the business context of the company and to the content and structure of the user interface. Therefore the preparation of conversation guidelines and pre-formulated questions should not be neglected (7). For our interview we thus predefined the content and also the sequencing of the topic categories. As to the technical carrying out of the interview Hopf describes a ”compromise” interview style that tries to find a compromise between the two extremes of openness and rigidness: the ”semi-structured interview” (13). The advantage of this method is that the interviewer can interact with the participants and modify the interview style, if necessary. He or she therefore reaches a high degree of openness towards new aspects that may come up during the conversation and should be integrated in the data record.

The semi-structured interview can be implemented by interviewing single persons or groups. The latter places higher demands on the interviewer when it comes to the integration of discussion members and to the structuring of the group interaction. At the same time, in group discussions, the interview situation can be stimulated by the exchange and discussion of differing viewpoints by the participants. This is widely seen as an advantage of group discussions (19). Respecting also the interviewees limited time resources in the AIRBUS organization, the following decisions have been made with regard to the requirement analysis.

- Focusing on two relevant user groups, i.e. engineers and employees in the field of "Product Safety".

- Undertaking a semi-structured group discussion during a workshop at AIRBUS in Hamburg.

2.1.3 Undertaking of the Workshop

The participants in the workshop were:

- 4 employees of the Engineering Division (e.g. development of the A 400 M APU and product management long range)

- 2 employees of Product Safety (document discovery)

Beside these participants, 4 employees of the Archive Service Division (ASS GmbH) took part in the workshop.

The first part of the workshop consisted of three steps which occasionally merged into each other:

- introducing the participants' work areas

- identifying typical use cases

- classifying the use cases with regard to the definition of common and individual characteristics for the different use cases

In the second part of the meeting, the prototype of the ASADO visualization module was explained and presented to the future users. While evaluating the different search options for the previously defined use cases, a connection was established between the software engineering and the user perspective.

2.1.4 Results of the Requirement Analysis

Before specifying the user oriented requirements it should be mentioned that a new insight in the constitution of the user groups was gained through the identification of single use cases for each participant. It can be said that the

core aspect for identifying the characteristics of use cases is not the affiliation to the pre-defined user groups (”Engineering” and ”Product Safety”). Instead, the collected data has revealed that it is important to consider the function of the document or document group in the working process of the user. This function is crucial for the differentiation of classes of similar use cases.

Respecting this altered viewpoint, an adequate classification of use cases can be put at the basis of the reflection of user requirements. The first category contains use cases frequently occurring in the context of Product Safety. In this approach to the document search, the users' task is to retrieve and analyze information about Quality Management issues. But this task is in some cases also part of the daily work of an engineer. An example is the internal project management where the identification of responsibilities for project modules is an essential part of the research activity. The characterizing factor is that the document(s) that are being searched for, represent evidence for judicial considerations in the context of Quality Management. Therefore, the category for the corresponding use cases will be named Quality Management and needs to be distinguished from the following one.

In the second category, various use cases are similar with respect to the technical purpose of the document search. The crucial point with this aspect is that it is less interesting in which document the technical content is found or by whom it had originally been produced. As soon as a valid source for the needed information has been found, the research task is fully implemented. This characteristic of document search applied to most of the participating employees working in the engineering departments. The essential fact is that the function of the search document is exclusively the transmission of technical content.

The two different use case categories are therefore characterized according to the function of the document in the work process of the user: **Quality Management** and **Technical Problem Solving**.

Further criteria for the classification of relevant use cases have been deducted from the data record. The corresponding questions could read as follows:

- To which extend does the user have knowledge about the required documents, regarding the content and/ or the meta data?

- How much knowledge does the user have about the work processes which are relevant in the search context?

- Are the documents related or unrelated?

- What result type does a user expect - single documents or sets of documents?

- How many pages do the documents contain?

The criteria for the classification and an approximation for possible values have been summarized in the following table:

criterion	**value**		
Function of doc. in work process	Quality Management	Technical Problem Solving	
Knowledge about doc.	Doc. unknown	Few information	Doc. known
Knowledge about work process	Work process unknown	Few information	Work process known
Relation between documents	Unrelated	Related	
Type of single result (nb. of documents)	Single doc.	Set of documents	
Nb. of pages of doc.	Small	Medium	Large

These criteria have different consequences for the search options realized in the prototype.

To start with the first criterion, we would like to distinguish the two typical classes of use cases in order to explain single aspects of the requirements for the search devices. The first category is the Quality Management context. In this case, the qualities of the search device have to serve for the identification of large document sets and their organizational or content-based relations between each other. This is connected to the challenge of finding every possible source of information related to a research task. Therefore, connections and links between different documents, ownerships and responsibilities need to be traced back as thoroughly as possible. Frequently, the employees facing the search tasks of the Quality Management do not have detailed context information about the corresponding work process.

The amount of knowledge about the relevant work processes in specific contexts influences the way in which iterative steps should be offered to support the search process. The required analysis of an unknown document set can hardly be overcome by relying simply on keyword search and browsing ordered lists. In order to improve this situation, the idea is to develop a map-like surface which enables the user to view the inter-document similarities and relations. This display mode can provide a better insight into the statistics of the examined document set (see also section 4.3). The proposed visualization module may help to overcome a lack of orientation in an unknown set of documents, as supplementary information of single documents can be extracted and displayed.

The integration of new paradigms for document set visualization is therefore regarded as helpful for creating effective and efficient search alternatives for the document discovery in the Quality Management context.

Coming to the second typical class of use cases, a slightly different situation can be observed. In the context of Technical Problem Solving, the user often has very detailed contextual information about the work process related to his or her search question. Therefore, data about the relevant business unit or the author of the document is not needed during a search. In the ”technical” use cases mentioned in the workshop, the starting point for a document request often contains pieces of information such as the aircraft type or technical specifications of aircraft components. The participants stated that the amount of information available for a search request may vary in multiple ways.

The amount of knowledge the user has about the wanted documents influences the type of the used search device. The search for a known set of documents or a search based on vague ideas of the document content (keywords) could be mentioned as two extreme use cases. In the latter case a cluster based and labeled pre-structuring of the document presentation would be helpful in order to give an idea of the possible relevant search results.

This shows that use cases within the two extremes need to be considered for the design of the search instruments. It is therefore recommended to provide various search functions, ranging from the simple keyword search to ranked lists and meta data filtering that enable the choice of an appropriate search method

for the user. As to the map-oriented display of large document sets, it should be mentioned that the use of this search function can equally support the use cases related to Technical Problem Solving. The possibility of quick scanning of large sets of documents might be simplified by the spatial display of search results and help to quickly identify relevant information sources.

Further detail of the different approaches to the visualization module can be found in the parts of the report dealing with the Interactive Exploration of documents (see section 4.3 "The ASADO Visualization Module").

2.2 Software ergonomics

2.2.1 Introduction

This part of the ASADO project report is about the ergonomic aspects of the scanning and indexing software Ascent Capture provided by the company T-Systems.

At the beginning of the project the ZAMIZ EDG Reading Team of the Airbus Archive Service in Hamburg had just started to use Ascent Capture. The goal of the ergonomics study is to evaluate whether this software fits the ergonomic standards and the physical, perceptoric and personal needs of the people working with it every day. Furthermore, it tried to give some proposals, where possible, how the software interface could be customized to the needs of the specific task of the Reading Team in Hamburg.

Software ergonomics is a topic that can take very different aspects into account: not only functional correctness, economic use, a clear design and usability of the user interface itself but also topics of the environmental factors that influence motivation, cognitive processes and the perceptoric and motoric performance of users. Basically software as a tool (here Ascent Capture) should be evaluated with regard to the degree to which it is adjusted to the human users and their tasks. The second approach is to regard the whole working environment from an industrial psychological point of view. The efficiency of the system should be increased whereas the system's performance should exceed the sum of the technical and human components (9).

2.2.2 State of the Art

There are two different approaches to software ergonomics: on the one hand usability engineering, taking place during the conceptual and developing stage (see figure 2.1) and on the other hand the evaluation and iterative inspection and redesign method, appropriate when the user interface is already in use (see figure 2.2) (24; 32; 35)

2.2.2.1 Usability Engineering

The action of designing a new software system involves uniting the imaginations of at least 2 different groups of interest: the users or costumers on the on hand and the designer on the other hand (in figure 2.1 displayed as yellow rectangles).

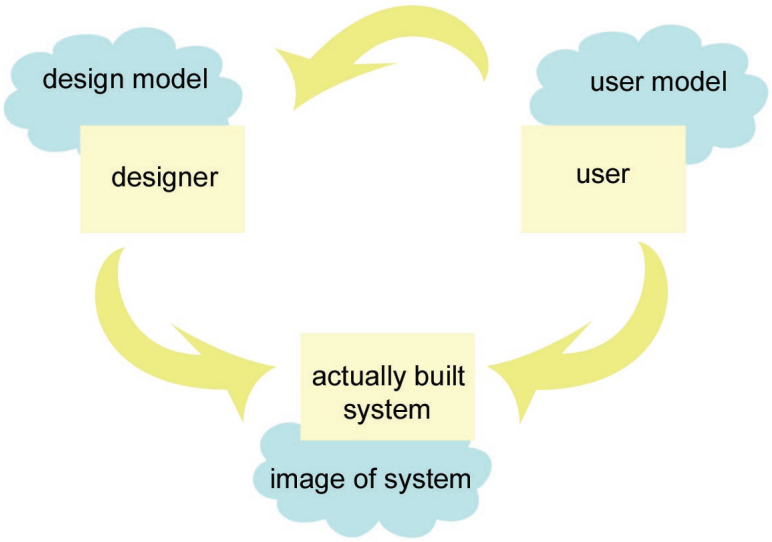


figure 2.1: usability engineering, taking place during the conceptual and developing stage

Figure 2.1: Usability engineering

The users of the software system have a certain idea about the software: they build a mental model, which is pictured as a blue cloud in figure 2.1. The designer as well builds a mental model out of the information provided by the user and out of his or her experience.

Here from the software system is actually built. For the constructed system there is also a model or image which has not to be the very same that is actually realizable in every little detail. The higher the congruence between the three models and the actually built system is the better is the software system.

2.2.2.2 Iterative Inspection Method

The development of a software system in n versions can be seen as an iterative inspection and redesign process. It starts out with the delivery of the first version of the software, next is the review by the customer which results go back to the designer as feedback. The developer checks in how far the single issues of the feedback can be realized and implements the second version of the software system.

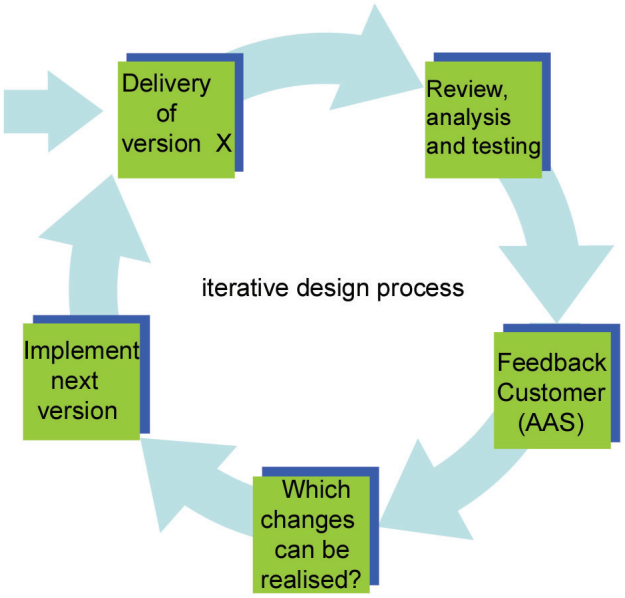


figure 2.2: iterative inspectio method (accoring to Thaller, 2002)

Figure 2.2: Iterative inspection method

The optimal procedure is a combination of both approaches. In our case, where the conceptual stage took place at T-Systems earlier and the software is already in use, we start out with the iterative inspection method. This study tries to support the customer at the stage of ”review, analysis and testing”.

For the methods the combination of a cognitive walkthrough and usability testing would have been the most interesting. For economic and legal reasons

it was not realizable to construct a version of Ascent Capture in the usability lab of the University of Osnabrück. It would have been hard to realize to have a considerable number of AAS employees use a software mock-up for indexing partly confidential documents in the University's laboratory.

Therefore this ergonomics study concentrates on the methods cognitive walkthrough and heuristic expert inspection.

2.2.2.3 Cognitive Walkthrough

In a cognitive walkthrough a human-computer-interface is evaluated with respect to the cognitive processes required for handling the software system. Experts who know the tasks and the context for solving this task well (reading team members) go through the procedure necessary to solve a given task with the help of the software system and formulate their thoughts and problems at every step of interaction with the interface.

2.2.2.4 Heuristic Expert Inspection

Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics") (24). The goal of heuristic evaluation is to find the usability problems in a user interface design so that they can be attended to as part of the iterative design process.

The DIN EN ISO 9241 is the international standard for the design of human computer interaction (HCI). The requirements defined in ISO 9241 are of course also taken into account in this study.

To get the best possible outcome for the whole organizational department in which the software in question is used, a holistic approach, where also psychological issues of the working environment and work structuring are considered, is recommended (9; 32).

2.2.3 Application

2.2.3.1 Restriction of Criteria

First of all the criteria, in respect to which, the evaluation should take place were restricted out of the variety of possible catalogues of criteria proposed in ISO 9241 and other relevant sources (11). This revealed that the following three values are most important for the needs of the Reading Team in the ZAMIZ EDG project, also according to the project management at AAS:

- Effectiveness: The correctness and completeness of the indexing of documents should be as high as possible

- Efficiency: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible

- Satisfaction: The system should be pleasant to use, so that users are subjectively satisfied when using it. (resulting in a high acceptance of the software tool and a contribution to the motivation of the reading team)

These three criteria are also rated to highest priority in ISO 9241-11, which can be interpreted as being on the right track and consistent with the international standard.

2.2.3.2 Cognitive Walkthrough and Inspection

The second step was a cognitive walkthrough of the scanning and the indexing module of Ascent Capture. For this, screenshots of every view of the first version of the software were used and an expert user described the single steps necessary during the process of archiving a single pile of documents with the help of Ascent Capture. Also her thoughts and opinions about each step were encouraged to reveal with the help of systematic question and recorded for analysis.

Additional the design of the interface was analyzed with the method of heuristic expert inspection in respect to usability principles, ergonomic icon-design, arrangement, colors and consistency to well known software packages with the help of screenshots.

The result was, that there were many time consuming problems in the first version.

2.2.3.3 Feedback Costumer

After a short period of working with the interface the whole reading team and an ASADO team member met with a representative of the developer of Ascent Capture and brought forward their wishes, concerns and suggestions about the first version. A reading team member also had designed a graphic as a suggestion for the second version of the validation interface, which was implemented by the developer.

2.2.3.4 Second Version of Ascent Capture

When the second version of Ascent Capture was delivered and installed, a second analysis with cognitive walkthrough and heuristic inspection was made. This revealed that some but not all improvement suggestions had been implemented by T-Systems and that there were still very time-consuming problems, which seemed technically solvable from the outside. At this time the implementation of the third version of ascent capture is in progress at T-Systems and will soon be delivered. So the circles of the iterative redesign process still keep turning.
Even though another system, Image Master, due to technical backbone problems with the installation of Ascent Capture on the workstations is used right now. It seems that the third, improved version of Ascent Capture could be closer to fitting the needs of the AAS Reading team.

2.2.3.5 Industrial Psychological Point of View

As already mentioned in the beginning, the second approach of this ergonomics study was to look at the whole working environment of the reading team in Hamburg in an industrial psychological view:
There exists coherence between the demand of the tasks and want of personal development (30). The task of the reading team, archiving engineering documents in different languages, can be categorized as a demanding task, which also can be

assumed due to the fact that only employees with academic degree are chosen to be members of the reading team at AAS right now.

2.2.3.6 Job Design

The task itself is the most psychologically important part of the working conditions (30), so possibilities for job design have to be kept in mind in order to achieve optimal results in our archiving project. Basically there are three kinds of job design: corrective, preventive and prospective design (Ulich, 1980). For all three concepts there was a short analysis of actions that had already been implemented in the Reading Team and a few suggestions of additional arrangements to further improve the working conditions of the employees by designing their tasks.

For reasons of shortness not all suggestions should be stated in this report. One example for the concept of preventive job design would be:

Preventive job design means that organizational psychological concepts and rules are regarded on the conceptual stage of operational procedures to prevent possible problems of health and well being caused by the functional interaction of humans and computers. For example could one working Team member look after one archiving box with documents. He or she could accompany the box from the customer, who produced the documents to the final location in the physical archive. All necessary working steps in between should be included as well. For the Reading Team in Hamburg a this is a possibility to get a holistic approach to their task on the on hand and on the other hand change the employees load in between sitting in front of a computer screen and archiving documents an other demanding tasks.

2.2.3.7 Motivation

Further on there are 5 core issues to support personal development and intrinsic motivation (30). These 5 issues are autonomy, holistic account and understanding of the task, diversity of requirements, possibility of social interaction and the opportunity for learning and self-development.

The Reading Team already had components in their internal structure that would cover these issues important for intrinsic motivation, but as there never can

be enough motivation for excellent work some additional ideas were suggested to the team management of AAS as well.

A special emphasis was set on the learning aspect and the recruitment and education of Reading Team members as trainer as there are more Reading Teams to come in the future in other locations.

On example for a suggestion in the motivational area would be the introduction of a tear-off calendar for the whole team. For every 10 or 15 meter of digital archived physical folders on page would be torn off and the new mileage for the whole team is visible for everybody. For full numbers like every five hundred meters there could be a kind of team ritual introduced like having a brunch together.

A special emphasis was set on the learning aspect and the recruitment and education of Reading Team members as trainer as there are more Reading Teams to come in the future in other locations.

2.2.4 Conclusion

Both topics of this ergonomics study, the iterative design process of the interface Ascent Capture and the optimization of the motivational and learning concept for the reading team have not been completed yet, due to the time limit of the ASADO project and the natural organizational processes that are of course still ongoing. The learning concept especially is still in the conceptual phase and without doubt it would be very challenging to continue studying its further development.

Chapter 3

Document Processing and Representation

3.1 Detecting Text Structure

3.1.1 Starting Point

The ASADO project received a set of documents to be analyzed on 3 DVDs. The set consists of 7,229 documents and each document is available as a multi-page TIFF(@TI) file and as an ISO-8859-1(@IS) encoded plain text file. The TIFF files are the result of scanning the printed documents and the corresponding plain text files are the output from the OCR software used at ZAMIZ EDG. Additionally, each DVD contains one Excel(@MS) file that holds the results of the manual classification performed by the members of the ZAMIZ EDG reading team for the documents on the particular DVD.

3.1.2 Characteristics of the Document Repository

The 7,229 documents comprise a total of 148,248 pages where one document has 20 pages on average and the largest document has 1,799 pages. They contain handwritten and typewritten text, tables, sketches and photos. The text is composed in one or two of the following three languages, namely, English, German, or French – the language may, indeed, switch multiple-times within a document,

i.e. parts of the text within a document are composed in one language, other parts in another language, and yet another part is composed again in the first language. According to the manual classification on the DVDs the languages are distributed as follows: 90,0% English, 9,7% German, and 0,3% French documents.

The average number of pages is quite appropriate; 20 pages are likely to contain enough textual material on one topic for the following processing stages to have a reasonable basis. On the other hand, it is unlikely that parts of one document would already be dissimilar to other parts of the same document – this happens when two or more distinct topics are covered within one document.

3.1.2.1 Textual Content per Document

The prototypical structure of a document is the following: The first page contains a distribution list. Hereafter, comes a cover letter of a few pages. The remainder of the document – and thus the actual content – is attached in the appendix. However, half of the documents only consist of 3 or fewer pages. Furthermore, we looked into the 10% non-English documents and observed that very often only part of a document is non-English, i.e. there are still some parts in English.

3.1.2.2 Textual Content per Page

A rough estimation about the textual content per page hints that – on average – we can expect 240 words[1] per page. As a reference we did the same calculation for a set of printer manuals, i.e. semi-technical material containing text, tables, and sketches. There, we can expect at least 580 words per page.

Intermediate Summary: Taking all the characteristics of the document set into account our data set is quite unsatisfactory for the tasks under consideration. Half of the documents are too short to contain a substantial amount of textual information and, additionally, the pages are only meagerly filled with words.

[1] leaving aside some details about what actually is considered a *word*

3.1.3 Preprocessing

Even though we had been given the extracted text of the documents in the plain text format we decided to reprocess the TIFF files, i.e. to produce our own OCR output documents.

We decided to use the end user application equipped with the same OCR engine that is built into the software used at ZAMIZ EDG, i.e. our software and the software that produced the OCR output on the DVDs share their central part. However, our end user version uses a more recent version of the OCR engine.

Now, why should we reprocess the TIFF files with an almost identical OCR software? For two reasons: a) we wanted to see whether the newer version of the engine performed better and b) the engine is capable of producing a variety of output formats, of which plain text is just one of them. Indeed, being able to choose another output format was the central issue in deciding to reprocess the TIFF files.

The output format we chose is XML[1](@XM). This format has received extensive attention lately because of some key features: it is platform-independent, supports Unicode(@UN) encoding, is self-documenting, and therefore not only machine-readable but is at the same time human-readable.

Furthermore, the XML output describes the physical layout information of a document, i.e. we are able to extract the exact position of words conjoined with their font sizes, lines, paragraphs, and tables within the particular page. We use the font size as a first, very rough estimation about what might constitute a heading, i.e. a sequence of larger words compared to the mean of font size values on a page. This facilitates the process of term weighting (see 3.4). The physical layout information could, then, in a second step, be interpreted to get a handle on the reconstruction of the logical layout of a document, i.e. augment textual parts with labels like heading of section, heading of subsection, enumeration, item list, etc. Assuming that an author makes use of logical elements to support the natural language transfer of content we would, in this vein, improve both the understanding of and the ability to classify the content.

[1]It takes only one step to readout the textual information from the XML file and construct the corresponding plain text file.

Two Problems during OCR: Generally, the OCR output might be erroneous. An error in the physical layout analysis might disrupt paragraphs because headers, footers, and floating material[1] could break up the paragraphs of the text; or the OCR software tries to recognize text in floating material, however, only partly successful, which results in corrupted words; or the OCR software confuses the usual gadflies $\{i, 1, I, l\}$, $\{h, lz\}$, $\{c, o\}$, $\{S, 5\}$, $\{O, 0\}$.

Secondly, our software often crashed when processing TIFF files that contained large drawings or folder tabs and, unfortunately, did not always recover from the crash, i.e. manual interaction was necessary to have the software continue its work. People at ZAMIZ EDG told us that their OCR software behaves similar in this respect.

Comparing our OCR output with the one from ZAMIZ EDG we can say that, as an impression, ours is slightly better. The new version of the OCR engine has, on average, a slightly better recognition rate, i.e. on average it commits less recognition errors but, still, some words that were correctly recognized by the other engine were now miss-recognized.

Summary: As an average worst case scenario concerning the textual content to pass on to the linguistic processing we have 1.8 pages sparsely filled with words of which some might be defective. Again, this holds for about 50% of the documents and, hence, the available data is quite unsatisfactory.

3.2 Linguistic Processing

This stage in our processing pipeline takes as input the XML documents from the OCR software and prepares lemmatized noun-phrase (NP) chunks (see 3.2.3) for the terminology extraction (see 3.3) and the vector based document representation.

Using linguistic insights about the inner structure of language and retaining, generally speaking[2], a more feasible approach towards terminology extraction we decided to start with the extraction of NP chunks.

[1] for example tables, figures, and footnotes

[2] After all, we might be dealing with three or more languages.

3.2.1 Language Identification

In order for the following linguistic processing to operate optimally on given textual content, it is necessary to identify the language the document is written in. In our case things are even more complicated for some of our texts *are* composed in more than one language. We decided to tackle this problem by identifying the language on a page basis, i.e. we assume that a language does *not* change within a page but only from one page to another. Indeed, this is quite arbitrary; however, the other options were assigning a language to each paragraph, or to each document. The former turned out to be inappropriate because paragraphs, often enough, have too few words for a proper identification, and documents would be too coarse to capture the mentioned fragmentation of languages across one document. After all, the intuition is that the language used in a document only changes from one section to another, and that sections would start on new pages.

We chose TextCat(@TC) because it works reliably on "spelling and grammatical errors in e-mail, and character recognition errors in documents that come through OCR."(2) It covers all the needed languages and, also very importantly, is freeware[1].

Problems during Language Identification: At a first glance, English texts were identified quite successfully but French and German ones were regularly misidentified. We had a look into the data and the problems seemed to be that some pages did only contain a few words, and others mostly special characters. We addressed the latter problem by replacing all special characters by a space character and then only considered words of two or more characters. The poor identification stayed.

A more exhaustive inquiry of the problem revealed that traits of English in the technical language used at AIRBUS seemed to be the truly intrinsic problem. German and French are hard to identify because within the frame of these two languages there are strong traits of English technical terminology. Therefore,

[1]We also performed test with online-versions of some commercial products – without a significant difference in the result.

there was no simple way to discriminate between the primary document language and the incidental English terminology.

Summary: Without spending more time on this issue we decided to assign English to *all* pages and continue in our pipeline, for the time being, monolingually. We put up with the fact that 10% of the documents will undergo an erroneously tailored linguistic analysis.

3.2.2 Tokenization, Lemmatization, and Part-of-Speech Tagging

In order to extract NP chunks we need to know which words *are* nouns. The process that tags all words in a given sequence with their part-of-speech is called part-of-speech (POS) tagging - or for the remainder of this paper just tagging. However, prior to tagging we need to split up the sequence of characters in a document into distinct meaningful units, called tokens – typically the words.

In addition, it is usually considered beneficial to conflate all possible word forms to what is called its *lemma* or *lexeme*, e.g. the forms of the English verb *processing*, *processed*, *processes* would all be lemmatized to *process*. Intuitively, the lemma corresponds to the entry in a lexicon that subsumes all inflected forms of a word and this directly connects to terminology extraction – the terms in a terminology should also be listed in their lemmatized forms. Importantly, the vector based document representation benefits because, it would seem rather odd if merely different word forms of one lemma rendered two documents more dissimilar.

We chose TreeTagger(29),(@TR) from the Institute for Natural Language Processing (@IM) in Stuttgart. It is an integrated framework that can be used for all three steps, tokenization, lemmatization, and tagging, is available for English, German, and French, and is free for academic usage.

Illustration: The following is a tokenized, lemmatized, and tagged example sentence. At first the example sentence is given, followed by the output where ... mark ellipsis, in both the example sentence and the output. Each token from

the input is returned on a single line where the lemma is followed by its POS tag, separated by `/`, and the word form from the input. POS tags beginning with `N` representing nouns, `JJ` adjectives, and `DT` determiners; `@card@` represents a number, `<unknown>` a lemma unknown to the tagger, and `<gibberish>` stands in for a character sequence that our processing pipeline considered odd and did not even pass it to the tagger.

```
Having a look into the Airworthiness Working Paper, 5th Issue,
dated 02.04.91, ... working with a load factor fwd = 6.0 g, ...

have/VHG Having      (1)
a/DT a
look/NN look
into/IN into
the/DT the
airworthiness/NN Airworthiness
Working/NP Working   (2)
Paper/NP Paper ,     (3)
5th/JJ 5th
issue/NN Issue ,     (3)
dated/VVN dated
@card@/CD 02.04.91 , (3,4)
...
<unknown>/NN fwd     (5)
<gibberish> =        (6)
@card@/CD 6.0        (4)
<gibberish> g,       (7)
```

(1) the inflected word form `Having` is reduced to its lemma `have`; (2) `Working` could also be a verb but contextual information resolves this ambiguity; (3) a punctuation mark is properly recognized and, hence, detached from the word; (4) a number – in a very broad sense – is properly recognized; (5) the POS tag is guessed correctly although no lemma can be assigned; (6) an OCR error is assumed and the single character is not passed to the tokenizer; (7) similar to 6 but, as a consequence, the punctuation mark is not recognized either.

Following are two more example lines of text, though, without the according analyses. The first illustrates the problem of sentence-boundary and word-boundary

detection: `3 hours to temp lower than` could be interpreted as an insertion between dashes – which is usually a sentence in itself or could be removed such that the surrounding sentence continues without interruption – although it should actually be interpreted as `2-3` and `-20°C`[1]. The second line illustrates how difficult it might be even for humans to properly name the tokens within a sequence of characters.

```
... exposed longer than 2 - 3 hours to temp lower than - 20°C ...

Ref.:AI/EA-A N° 412.0121/92
```

3.2.3 Chunking

An often found definition states that *chunks* are non-recursive, non-overlapping, flat structures of a sequence of tokens, i.e. a sentence might contain more than one chunk but they would not have any words in common; and, in any case, the inner structure of a chunk is simple.

We deviate from this view in the following way: we assume a specific – but still simple – inner structure and unroll it *recursively*, which entails that our chunks *have* words in common. The structure we assume is that noun phrases (NPs) may be *conjoined* or NPs may contain a preposition with another NP attached to it. Unrolling the structure, then, is done in three stages: the first stage detects all largest non-overlapping NPs within a sentence (SplitNone), the second stage splits these chunks at predefined POS-tags (SplitOnce), and the third chops off words from left to right (SplitAll). Chopping off words from left to right corresponds to the observation that nominal compounds in English and German are usually right-headed[2].

Considering recent work (cf. (3), (20), and (21)), phrases containing adjective, adverbs, and verbs should also be taken into consideration, at least, for terminology extraction because, as L'Homme puts it, they "provide clues to the meaning

[1] Typographic convention, indeed, favors this spelling.

[2] Note that French compounds are left-headed and Spanish compounds are both left- and right-headed.

of terms, [and] others are morphologically and semantically related to terms in noun form". Extending our NP chunks in this manner should be considered an option for further work.

We chose the Natural Language Toolkit(@NL) for this task because it includes a regular-expression based chunk parser and is freeware.

Illustration: Following is an illustration of the unrolling process for the elaborated example from above. The initial step is to find sequences of tokens that could possibly constitute a NP and then, in an initial filtering step, all determiners are deleted, i.e. *a look into the paper* and *the look into a paper* would both be conflated to *look into paper*.

These are the automatically extracted NP candidates. We see that all determiners have been deleted and that ',' precedes or succeeds some of the candidates.

```
[look/NN into/IN airworthiness/NN Working/NP Paper/NP
 ,/CON 5th/JJ issue/NN ,/CON]
[with/IN load/NN factor/NN fwd/NN]
```

First stage of the unrolling process, what will be referred to as *SplitNone*: the longest possible NP chunk has been extracted and candidates that did not constitute a full NP have been discarded.

```
look into airworthiness working paper , 5th issue

load factor fwd
```

Second stage of the unrolling process, what will be referred to as *SplitOnce*: only the longest chunk could be split into three sub-chunks; the splits were done at ',' and `into`.

```
look into airworthiness working paper , 5th issue
look
airworthiness working paper
5th issue

load factor fwd
```

Third stage of the unrolling process, what will be referred to as *SplitAll*: no more splits were possible but the sub-chunks are reduced from left to right until there is only one word left.

```
look into airworthiness working paper , 5th issue
look
airworthiness working paper
working paper
paper
5th issue
issue

load factor fwd
factor fwd
fwd
```

3.3 Terminology Extraction

3.3.1 Introduction

The terminology used in technical documents of industrial research and development is highly branch specific. Therefore, when industrial research and development paper documents are converted into an electronic database, it needs this specific terminology as a basis for document indexing and labeling in order to retrieve relevant documents in a meaningful way later on. Also, for building up a useful ontology for knowledge bases, the terminology needs to be defined thoroughly by intellectual means (22). General methods to define a terminology are specified in the international standards ISO 1087 (36) and ISO 704 (26), and indexing of documents is standardized by the German Industrial standard DIN 31623 (5). These standards refer to intellectual or semi-intellectual methods by experienced personnel for both terminology extraction and indexing. Such a terminology analysis is problematic due to the fact that the intellectual procedure is time-consuming and always subjective. To overcome subjectivity, expert teams are usually in charge of finding a common agreement on the relevant terms. The quality and relevance of the terminology created this way is very high but so are the costs of manpower. Therefore, starting with a set of noun phrases, a simple constituent analysis has been performed and, in a second step, an extended constituent analysis was developed to extract the terminology from research and development documents of aviation technology industry. An analysis of the relevance and quality of the terms obtained was performed semi-automatically to reduce manpower costs. Within this evaluation, the terms obtained could be classified and the classes may serve as a basis for a generic ontology.

3.3.2 Constituent Analysis

The data base for the constituent analysis was a set of 13221 noun phrases (NPs) as extracted from OCR-processed data files (see 3.2). The number of words per NP varied between 1 and 72. None of the NPs contained one-letter or two-letter words except previously defined relevant two-letter words like ”TV”. From these NPs, the following constituents were extracted automatically:

- head of NP (example: ”test”) denoted as ”S” for single constituents,
- head and 1st modifier (example: ”load test”) denoted as ”D”for a double constituent, and
- head and 1st plus 2nd modifier (example: ”dynamic load test”) denoted as ”T” for a triple constituent.

Noun phrases of more than three constituents, for example a quadruple consisting of head, first, second and third modifier, are usually extremely specific in the sense that they appear only in a specific context or only in one single document. Therefore, the simple constituent analysis has been limited to the heads and the first two modifiers of the noun phrases. Any modifier of higher order has been cut off for this analysis.

By adding up all identical singles S, doubles D and triples T and treating them as identities, a total of 1134 different terms of the Type S, D and T was obtained. They were grouped as follows:

- Group A: It was assumed that triples T and the doubles D and singles S they consist of are of highest specifity. Accordingly, they were grouped together in this group (example: T = dynamic load test, D = load test and S = test; ”test” being the head of the triple and ”load” and ”dynamic” being the 1st and 2nd modifier respectively).
- Group B: As assumed above, the constituents of a triple T are of highest specifity. Consequently, any double D not appearing in group A but having a head which appears as a head of a triple T in group A, should be less specific. Thus, all doubles D which do not appear in group A but their heads being singles S in group A are elements of this group (example: ”hic test”, with ”test” being a single S in group A but ”hic” not appearing as first modifier in group A and ”hic test” not being double D in group A).
- Group C: All singles S appearing as heads of NPs but not as heads in group A (i. e. not as singles in group A), and all doubles D with these heads. These constituents should also be less specific compared to those of the groups A.

The distribution of these 1134 terms was 54.7 percent in group A, 14.3 percent in group B and 31.0 percent in group C. It must be noted here that the elements of group C do not form the complementary of the groups A and B within the total of all NPs due to the following reason: next to those NPs from which the singles S, doubles D and triples T of the groups A, B and C were extracted, numerous NPs exist which consist of only one word and which are not heads of other NPs (e. g. "lbs" which denotes a weight dimension used in pressure measurement and appears 2475 times). This means: although such one-word NPs may appear frequently, they do not contribute as a head to any double D or triple T. Therefore, such one-word NPs were not included into the above groups A, B and C, i. e. they were neglected in the further investigation. The advantage of excluding these simple one-word NPs from the further investigation becomes obvious: all one-word NPs which contribute to doubles and triples - and thus, are singles S in the groups A, B or C - may appear comparably frequent as the excluded one-word NPs. But being elements of the groups A, B and C, they are heads of several different doubles D and triples T and accordingly, they are specified by their modifiers (e. g. the one-word NP "seat" appears 2161 times which is nearly as frequent as "lbs" but "seat" is specified by being head of more than 50 doubles and triples). This way, one-word NPs like "seat" being specified by modifiers contribute to the terminology and non-modified one-word NPs like "lbs" do not. This shows that assuming the most frequent one-word NPs as to be irrelevant due to their low specifity and cutting them off immediately after the noun phrase extraction from the documents may lead to a loss of relevant terms. By performing this simple constituent analysis as described here, frequent one-word NPs which contribute to a terminology can be separated from those which are not terminology relevant.

3.3.3 Extended Constituent Analysis

By rough intellectual comparison of the 13221 NPs and the 1134 terms obtained from the above constituent analysis it turned out that a number of technical expressions appearing as higher order constituents of the noun phrases (e. g. 3rd or higher order modifier) might be useful for the terminology. Therefore, the

constituent analysis has been modified by introducing a ”sliding slot” which can be described as follows: an NP contains n words and thus can be expressed as NP = {np(W1,...,Wi,...,Wn)}, i = 1...n with Wi being the word in the i-th position. Singles S, doubles D and triples T are now defined as

- Si = {np(Wi) | np(W1,..., Wi,...,Wn) ∈ NP} with i= 1. . . n,
- Di= {np(Wi, Wi+1) | np(W1,..., Wi, Wi+1,...,Wn) ∈ NP}, i =1...(n-1), and
- Ti = {np(Wi, Wi+1, Wi+2) | np(W1,..., Wi, Wi+1, Wi+2,...,Wn) ∈ NP}, i = 1...(n-2)

respectively.

The four-word noun phrase NP(test, load, dynamic, vertical) may serve as an example: the constituents are the words W1 = test, W2 = load, W3 = dynamic and W4 = vertical. For i=1, the head of this NP is Wi = W1 = test, its first modifier is Wi+1 = W2 = load and its second modifier is Wi+2 = W3 = dynamic. This delivers the triple T1 = ”dynamic load test”, the double D1 = ”load test” and the single S1 = ”test”. This corresponds to the simple constituent analysis as described in the previous chapter (see 3.3.2). If i = 2, the head of the original NP is neglected (”test” in the example), its 1st modifier becomes the head of the new NP, its originally 2nd modifier becomes the 1st modifier and its originally 3rd modifier becomes the 2nd modifier in the new NP. In the example above, the new noun phrase is NP(W2, W3, W4) = NP(load, dynamic, vertical) which is read as ”vertical dynamic load”. This method is applied to all Si, Di and Ti from i = 1...n. As a result, this method delivers singles Si, doubles Di and triples Ti which can be formed by adjacent constituents of the original NPs. Adding up all identical singles, doubles and triples and choosing an appropriate cut-off limit for rare appearances, this method delivers a total of 2225 singles S, doubles D and triples T. Grouping them into the groups A, B and C results in 815 terms in group A (an increase of 33.6 percent compared to the first constituent analysis), 226 terms in group B (39.5 percent increase) and 460 terms in group C (31.1 percent increase). By this extended constituent analysis method, not only terms of the group A, B and C type were obtained but also additional triples. To find out

whether these additional triples are terminology relevant, they were included into the qualitative evaluation (see 3.3.4) and for this later step, they were grouped into two new types of terms as follows:

- group D: 44 triples T which are not triples in group A but with their heads being single expressions in group A (example: ”double std test” with ”test” being its head and also being head of NPs in group A), and
- group E: 65 remaining triples T which do not appear in the groups A and D.

All single words which do not appear in the groups A, B, C, D and E were neglected in the following due to their low specifity. The remaining 1610 terms were set to be 100 percent and based on this, the total increase of terms compared to the first constituent analysis was 42.0 percent. The term distribution is then 50.6 percent in group A, 14.1 percent in group B, 28.6 percent in group C, 2.7 percent in group D, and 4.0 percent in group E.

3.3.4 Qualitative Evaluation of the Terminology

Since the quality of a terminology is of high importance for the users of a knowledge base (4), the branch-specific relevance of the 1610 terms obtained by the extended constituent analysis had to be evaluated. First of all, singles S from the extended constituent analysis were neglected due to their low specifity, and useless terms, e. g. resulting from OCR-reading errors, were eliminated (in average 2.3 percent of the groups A, D and E, 3.1 percent in group B and 4.6 percent in group C). Also, company names and individual names (in average 6.2 percent in group A, D and E, 1.8 percent in group B and 10.4 percent in group C) were separated. The remaining 1158 doubles D and triples T were used as search criteria for the internet search-engine ”Google” (trademark) in the advanced search mode ”exact phrase” and English language. The search engine quality test was chosen because subjectivity in defining the branch-specific technical terminology and high personnel costs should be avoided. The subjectivity is usually minimized by discussing every terminus in interdisciplinary teams of experienced engineers who release terms of common agreement. This procedure is extremely

time-consuming and still refers to a closed group of some individuals. Therefore, the internet search engine test has been performed to find out which of the terms extracted in ASADO are used by the relevant international authorities and the specific industry branches involved.

The number of search results varied from 0 to 20 million; for triples it rarely exceeded 500, and only a few doubles delivered more than 150.000 results. Since "Google" supplies the results in the order of their relevance only the first 10 results were evaluated (which makes a total of roughly 11 000 results being evaluated). The evaluation was performed by either analyzing the link to the website in case the link was self instructive (e. g. homepage of the International Standardization Organization ISO www.iso.org) or by following the link delivered with the result. The search results were classified during the evaluation process under the aspect of the branch specifity of the terms:

- class 1: no result (zero appearance which means highly specific to Airbus Industries),
- class 2: websites or documents of governmental and non-governmental organizations with legal impact (e. g. administrations, regulation authorities, standardization committees),
- class 3: websites and documents of the special industrial branch of this project (aviation in general),
- class 4: websites and documents of supplier´s industry relevance (electrical, mechanical and safety engineering, material science, etc.), and
- class 5: others (e. g. medical, biotechnology and science relevance or diverse).

The results of the evaluation process are shown in table a below, in which the totals of over 100 percent indicate that some terms revealed results from two or three classes (where only a maximum of two terms per term group related to three classes and none related to four classes). The terms of class 5 by definition relate only to this class (others), i. e. they do not appear in any other class. In table a, the results of the group A, D and E terms were accumulated because the

number of group D and E type terms was low (below 4 percent in each group) and their distribution within the classes was similar to the group A terms.

total=1158	group A,D,E terms	group B terms	group C terms
class 1	159 (22.4)	2 (0.9)	10 (4.3)
class 2	100 (14.0)	23 (10.7)	21 (4.3)
class 3	82 (11.5)	37 (17.2)	32 (13.8)
class 4	233 (32.8)	48 (22.3)	42 (18.1)
class 5	183 (25.7)	127 (59.1)	140(60.4)
total	**757** (106.4)	**237** (110.2)	**245** (105.6)

table a: classification of terms by internet search result evaluation (in brackets: percent)

The class distribution of all triples (column "group A,D,E terms") shows that only 25.7 percent of those belong to class 5 (others) which indicates that ca. 75 percent of all terms in these groups are highly specific for the industrial branches. In the groups B and C which contain doubles but no triples ca. 40 percent are comparably relevant: ca. 60 percent of all terms belong to other branches, i. e. they are element of class 5. This relevance difference in group A, D, E terms compared to the group B and the group C terms results from the higher specifity of triples due to their additional modifier. A significance is found in class 4 of the group A, D, E terms where the number of elements is maximal (233 = 32.8 percent): class 4 contains all terms relating to the supplier industry terminology. This maximum reflects the fact that all original NPs stem from documents concerning the research and development activities for a system which is delivered by the supplier industry.

3.3.5 Summary and Discussion

In this project part, the branch-specific terminology of the aviation industry was extracted from noun phrases derived from research and development documents. Based on a simple constituent analysis, an extended constituent analysis was developed as a new method which reveals terms of two or three constituents forming either noun phrases or composites. The quality of the terminology obtained by the extended constituent analysis was evaluated semi-automatically by using the

terms as search criteria for an internet search engine. The results delivered by the internet search were classified by objective criteria and thus, subjectivity in the relevance analysis of the terms was avoided. Classification led to five classes, two for highly specific terms (class 1: no internet search result and class 3: results from the aviation industry), class 2 relates to legal and standardization issues (e. g. for transport safety) and class 4 to the relevant technological vocabulary used in the system supplier's industry. Class 5 consists of miscellaneous branches, e. g. biomedical research which plays a role in the analysis of injury risks in transport. Also, company names and names of individuals could be separated and these form a further class. Generally, terms consisting of three constituents (i. e. triples in the groups A, D and E) are of higher specific relevance than those consisting of only two constituents (i. e. doubles in the groups A, B and C). In the groups containing triples (A, D and E), nearly 75 percent of the terms relate to aviation industry and its related industrial branches and only 25 percent to others branches. In the case that terms are formed by two constituents (doubles), 40 percent are specific to these industrial branches and the remaining 60 percent contain terms of science, biomedical and other branches. From these results it is concluded that the extracted terminology is of high relevance and quality and can be used as a basis for indexing and labeling.

3.4 Vector based Document Representation

Once the documents are linguistically processed we have to find a representation of the documents the computer can work with. This representation should provide us with the possibility to compare the content of the documents. A widely used approach for representing documents in the area of information management is the vector space model (Salton as cited in (14)). In this model each document is represented as a feature vector. For every document it is determined how often the single features appear in the document. These vectors specify points in a high dimensional space. The spatial proximity of two points in that space are interpreted as the semantic similarity of the documents. The closer two points are the more similar is the content of the corresponding documents. This similarity is based on the occurrence and the frequency of the different terms. Despite the fact that this representation does not consider the order and succession of the terms, it is a close enough approximation to suit our needs. This kind of representation gives us the possibility to handle textual information like numerical data.

3.4.1 Construction of the Vector Representation

Starting point the for the vector representation are the different versions of the XML documents constructed during the linguistic processing (3.1).
The construction of the vector representation can be subdivided into three steps. The first step is indexing of documents also called feature selection. Here it is decided which kind of the textual information should enter the representation. The second step is the weighting. Not all terms in a document contribute the same amount of information to its content. Prepositions for example but also very general terms only carry a minor semantic content. This fact is captured by introducing relevance values based on an importance measure. The last step is the reduction of the dimensionality. With every term we use the dimensionality of the vector space increases and further processing of the data gets computationally more difficult. Therefore reducing the dimension wherever possible is a major concern.

3.4.1.1 Document Indexing

The quality of the vector representation depends on the choice of the features. A feature is an arbitrary unit of text, this either can be a single word(*'seat'*), a multi-word expression(*'static test report'*) or a noun phrase(*'interface load of this seat'*). In a document collection we can identify hundreds of thousand different features. However not all of these terms are helpful for further processing. Words from closed word classes like determiners or conjunctions do not carry information with respect to the content description. As we have seen in 3.1 we represent the content via noun phrases and, hence, they will also constitute the features of the vectors.

3.4.1.2 Term Weighting

Besides identifying the relevant terms we have to consider the fact that terms are of different importance for a document. Counting the frequency of a term can be a good heuristic. Frequent terms tend to be of some importance for the meaning of the document than less frequent ones. However relevance does not necessarily have something to do with the frequency of the terms. The keywords of a document might only be mentioned two or three times. As our main purpose is to arrange the documents according to their mutual relations regarding the content we have to identify those terms that are important for certain groups. Terms that appear frequently in all documents are of little value for discriminating documents. What is needed is a measure that combines the frequency of a term with the distribution of that term over the whole collection. The first part is the *term frequency* (TF), that is simply how often a term t appears in a document d (Luhn as cited in (14)). For the second part we compute the inverse document frequency (IDF) (Sparck Jones as cited in (14)) . Terms that appear in many documents receive a low IDF value than terms that appear only in a few documents. This leads us to a widely used weighting schema, the TFIDF measure:

$$w_{t,d} = tf_{t,d} \times log_2 \frac{N}{n_t}$$

$w_{t,d}$ is the weight the term t in document d receives, N is the number of all documents and n_t is the number of documents where the term t appears in.

Beneath the normal IDF term weighting we saw the need for other weighting methods as well. As we have seen before the usable textual information of the documents is very sparse for most documents. We therefore have to exploit the available content as far as possible. For this reason we considered layout information as well. The two features we have taken into account are the position of the terms as well as the font size of the terms.
For calculating the *page weight* of a term within an document we used a very simple heuristic. Terms that appear on the first 3 pages are considered as being more important than terms that follow later on. The idea behind this is that many documents do have on the first few pages the title as well as an index or an introduction. It has to be clear that this can only be a coarse heuristic, as the documents are very diverse in structure and content. The second feature we take into account is the font size of the terms. We assume that terms that are emphasized in size relative to the surrounding text are more relevant than terms that are not. We therefore introduced a *font size weight* receiving a high value when the font size of a term is larger than the mean font size of the page. The reverse is not true. Terms that are smaller than the average term on a page are not considered less important. When writing a text we normally do not de-emphasize terms by writing them in a smaller font. It just might be the case that they are part of caption, telling us nothing about its relevance.
For both measurements it holds that we average the values over all occurrences of that term in a document.

3.4.2 Feature Reduction

The dimensionality of the vector space increases with every selected feature, that is, for example every word taken into account as vector component. This leads to difficulties for further processing, due to two reasons. The first reason simply is the computational complexity. The more dimensions the vector space has the more calculations have to be done, and as a consequence the whole processing is slowed down. The second reason is the so called curse of dimensionality. This is a problem with high dimensional vector spaces. Points tend to become equidistant from one another, making it difficult to make any statement about the mutual

relations of the documents. This leads to problems for the clustering as well as for the mapping of the documents. Consequently, one has to focus on keeping the dimensionality of the vector space low. This can be done already by the selection of the features as well as in all subsequent steps.

3.4.3 Putting it all together

The linguistic processing provides us with three versions of the documents differing in the features they constitute.

- Pure word forms
- Lemmata
- Nominal Chunks

Those are the starting point for the construction of three different vector representations. We determined the *term frequency* as well as the *inverse document frequency* for all terms. Further we computed the *font size weight* and the *page weight* . These values were combined to a single measure, leaving us with a matrix, where the rows are constituting documents (document vector) and the columns(term vector) terms. Now we applied two dimension reduction methods for scaling down the size of the matrix.
The first pruning step is to omit those terms that appear only in one single document. The second pruning step is based on the variance distribution of the terms over the document collection. After normalizing the document vectors to unit length we computed for each term the variance over the whole collection. Low variance values stand for a equal distribution of a term over the whole collection. We omitted all terms with a low variance value until we received a dimensionality that lead to good results in later processing steps. A magnitude of 4000 dimension turned out to be a reasonable magnitude.
Those different vector representations are the foundation for the further processing steps. They are the starting point for the clustering as well as for the document visualization.

Chapter 4

Analyzing and Presenting the Document Space

4.1 How Document Maps Can Help in Information Retrieval

The advent of the world wide web and digital libraries and catalogues require the development of novel methods to find and access information stored in documents. Currently, the predominant technique for information access in unknown document collections is a **keyword search** with a relevance–ranked list as a result. This works well, if the query terms are unambiguous and the user can formulate a well–defined query. However, if the search does not deliver the desired results or a too heterogeneous document collection, it is up to the user to formulate a better–fitting query. Similar problems arise, if only vague ideas exist about the documents of interest, or due to lexical ambiguities.

Browsing is another important search task. Typically it is regarded a more exploratory activity, with the aim of gaining overview over a document collection or identifying documents of interest without a clear preconception. The tools provided are normally document organization structures like hierarchical folder structures, annotated catalogues, meta–data classification or hyperlinks between documents.

Clearly, **ranked lists** are an effective way to display data ordered by a linear

property — such as relevance to a query or the date of creation of a document. However, they are not well–suited to display complex relationships within the retrieved document set or to support exploratory browsing of the result set.

Fortunately, alternative paradigms for document set presentation are available. The **cluster hypothesis** states that "closely associated documents tend to be relevant to the same requests [..., and] relevant documents tend to be more similar to each other than to non-relevant documents" (10)(p. 2) Findings from information foraging theory support and refine this claim by identifying typical information seeking strategies comparable to animals' food and mate seeking behavior (25). A central notion here is "information scent" , which denotes the cues given to guide a user on his track to the desired results. Again, it is assumed that relevant documents are likely to be found in the vicinity of other relevant documents.

Given these findings, it becomes evident that some information retrieval tasks can greatly benefit if the user is enabled to inspect the **inner similarity structure** of a retrieved document set. First, this facilitates an initial overview of the coarse structure of the result set to identify subgroups and outliers. Further, once a good hit has been identified, users can find further relevant documents more easily by browsing its similarity neighborhood. And moreover, experienced users can instantly evaluate the quality of their search terms supported by visual cues, e.g. how shattered the result set is presented and how clearly clusters are separated.

Maps and map–like displays are a premier candidate to display inter-document similarity structure: The map metaphor is well–known to all users from everyday life. Hence, a plethora of cartographic techniques can be utilized without the need of explanation. It has been shown that the distance-similarity metaphor is adopted effortlessly (23). Moreover, navigation has become the predominant metaphor of hypermedia (33), which further facilitates the introduction of spatial metaphors in document presentation.

A variety of mapping paradigms for document sets have been developed.

- Query–document maps and biplots can be used to visualize document properties, such as their relation to the query terms or meta-data.

- For the display of inter–document similarity, document networks, cluster visualization and Self–Organizing Maps are frequently encountered solutions. These techniques sort documents according to a pre-extracted discrete similarity structure (e.g. groups or map units).
- Projection techniques such as Multi–Dimensional Scaling (MDS) or Principal Component Analysis (PCA) directly produce a coordinate configuration to indicate inter–document similarity via spatial proximity. In contrast to the above mentioned techniques, each document receives an individual location on the map. This allows the user to extract a group structure on his own, according to his needs and perspectives.

4.2 Clustering

4.2.1 Introduction

Clustering is a key concept in automatic knowledge organization. The basic idea of clustering is to assign each object to clusters such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized. Document clustering is the automatic content-based categorization of documents. Clustering allows better and faster browsing and search for relevant documents by means of a grouped presentation of a huge document collection. Furthermore, it improves the user interface by facilitating navigation and organization of document collections.

We've chosen clustering and not classification, as clustering is automatic and reveals hidden or latent similarities between objects which are not obvious from the superficial look at the document titles and summaries but which become obvious at the closer analysis of a document. Consequently, clustering is one of the main steps in knowledge management. Unfortunately, the results of most clustering algorithms still leave much to be desired due to not very intuitive groupings of documents and rather bad labeling. Furthermore, a lot of good clustering algorithms are computationally very expensive.

4.2.2 Procedure

4.2.2.1 Prerequisites

There are two minimal prerequisites to perform clustering on a document collection. First, we need to represent documents in a machine tractable form, namely to represent a collection of documents as a huge sparse matrix. The next point we have to agree upon before applying a clustering algorithm is a similarity measure. What does it mean for one document to be similar to another one? Since the documents are represented as vectors, we need to define a measure of similarity of the vectors in a vector space.

The resulting groupings of one and the same document collection can be very different depending on the document representation and on the clustering method. The important matter in document representation is the choice of features to be included into the vector and their weighting correspondingly, which can have a

dramatic impact on clustering. This can be shown on a simple example (s. Figure 4.1). If we decide that the form of the subject is more important then color, then we either only include the feature 'form' into the representation of the object or we give more weight to this feature in comparison to another one (here: 'color'). As a consequence we get completely different results! Thus, the choice and weighting of features in documents' vector representation is a crucial preprocessing step in the clustering procedure.

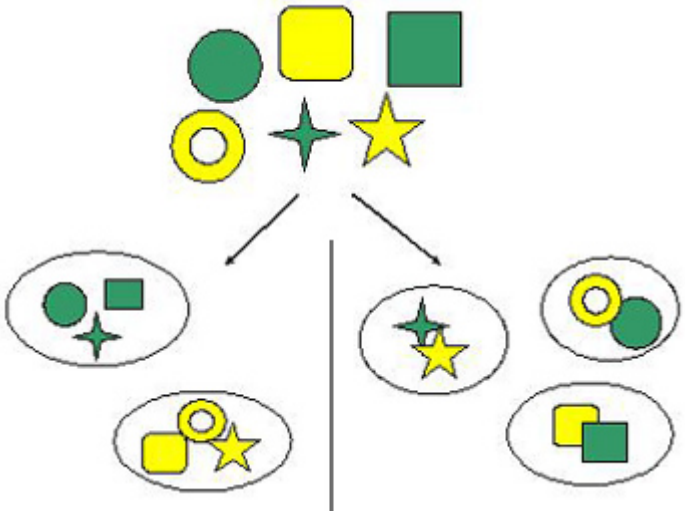


Figure 4.1: Clustering

Furthermore, the resulting clustering can be also different depending on the clustering method.

4.2.2.2 Clustering Methods

Generally, clustering methods are divided into two groups:

1. non-hierarchical methods (also called partitional or flat clustering): find all k clusters at once, where the k apriori-defined number of clusters are being optimized till the elements inside every cluster are as homogeneous as and the ones outside the cluster are as heterogeneous as possible. K-means is the most popular algorithm in cluster analysis. It is a hard clustering

algorithm that defines clusters by the center of mass of their members. Flat clustering is preferably applicable if the efficiency is important in the application, and especially when the dataset is very large. K-means mostly achieves sufficient results.

2. hierarchical methods: produce a set of nested clusters in which each pair of objects or classes of objects is progressively nested in a larger cluster until only one cluster remains. The hierarchical clustering can be further subdivided into:

   - agglomerative (bottom-up): start with the unclustered dataset and gradually merge the objects into a hierarchy (in N-1 steps)
   - divisive methods (top-down – less common): begin with all objects in a single cluster and divide a cluster until each object resides in its own cluster.

Hierarchical methods are better for a detailed data analysis and give more exact information then flat clustering, consequently they are more computationally expensive, which makes them practically out-of-use for large collections of documents (14).

Thus, it would be good to have a method which performs like hierarchical clustering methods and which is, at the same time, at least as efficient as flat clustering. Bisecting K-means offers this double functionality. It was shown experimentally that bisecting K-means can produce clusters of documents that are better than those produced by regular K-means and as good as or better than those produced by hierarchical clustering techniques (34). Thus, bisecting K-means was chosen for our clustering experiments.

4.2.2.3 Algorithm and Parameter Settings

Bisecting K-means is a hybrid approach that combines the advantages of hierarchical and non-hierarchical clustering. The algorithm goes through the following 4 steps (s. 4.2).

1. Start with a single cluster of all the documents ('a bag of documents').

2. Find 2 sub-clusters of a fixed cluster using the basic K-means. There is little difference between the possible methods for selecting a cluster to split (34). In our experiments we split the cluster whose bisection optimizes the value of the overall clustering criterion function:

$$I_2 = maximize \sum_{\iota=1}^{\kappa} \sqrt{\sum_{\upsilon,\omega \in S_\iota} sim(\upsilon, \omega)} \tag{4.1}$$

3. Repeat step 2 - the bisecting step - for a certain number of times and take the split that produces the clustering with the highest overall similarity - in our case the resulting 2-way clustering solution which optimizes the I_2 clustering criterion function. In general, the I_2 criterion function leads to very good clustering solutions (37). Cosine was used as a measure of similarity.

4. Repeat steps 2 and 3 until the desired number of clusters is reached.

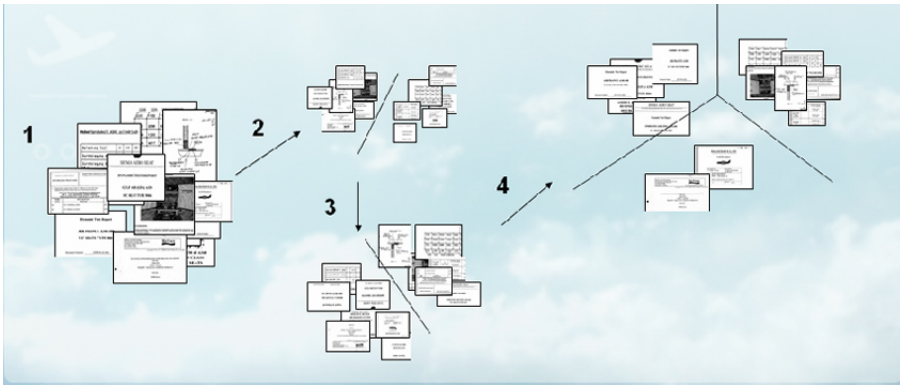


Figure 4.2: Bisecting K-Means

CLUTO clustering toolkit (15) was chosen for our experimenting. CLUTO is a software package developed specially for clustering of low- and high-dimensional datasets and for analyzing the characteristics of the various clusters. Multiple classes of clustering algorithms, including bisecting K-means are implemented in CLUTO, which was the main motivation for us to choose it. Furthermore, the

package is written in C, consequently it is fast and can deal with large collections of documents.

4.2.2.4 Hypotheses

The idea was to perform the 'bisecting K-means'-based clustering using different vector representations of documents discussed in Chapter 3.4 on p. 46. Thus, one of the tasks for clustering was to find out whether the suggested 3 feature selection techniques (pure word forms, lemmas, nominal chunks) as well as 3 different weighting schemes(standard term frequency or term frequency `x` inverted document frequency -TFIDF, font-size, page weighting) could bring the better results and would improve clustering. The standard procedure is to take either pure word forms (rarely) or lemmas as features, and apply TFIDF-weighting. The resulting clustering is unfortunately still far from satisfactory. Thus, the hypotheses to validate were whether we can achieve better clustering results if:

1. we use different weighting for different font-sizes used in the texts with the implication that the fonts of less important information are smaller.

   The font-size weighting as well as page weighting should intuitively have a positive effect on the clustering result, as the textual information available in the technical documentation is very sparse, so that every word literally matters; and if a word contained in the title of the document is given a double weight, then it is to be reflected somehow in the results.

2. we include nominal phrases instead of using only single words or lemmas as features into vectors.

   We would expect that the inclusion of the nominal phrases into the vector representation could induce a better performance as the nominal phrases used in technical documentation are highly application specific; therefore, though one-word terms may overlap in different types of documents, it is very unlikely that compounds or nominal phrases would be shared by documents which have not much in common.

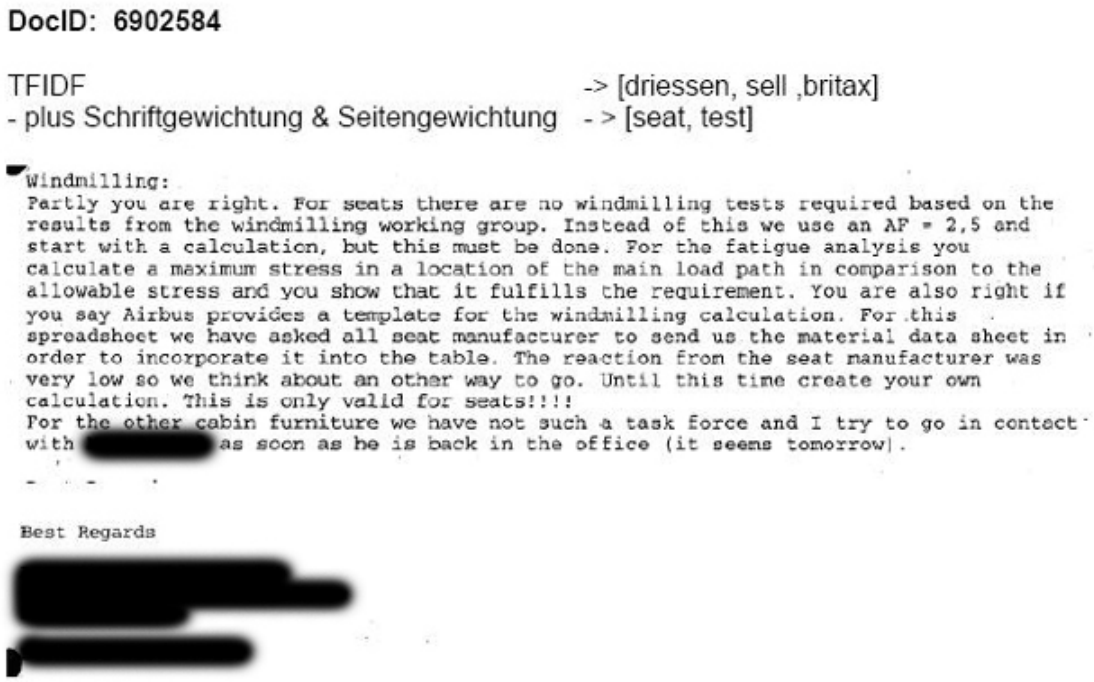
4.2.2.5 Experiments and Evaluation

In the first run, a representative set of documents was chosen out of the collection, so that we could easily test and evaluate our hypotheses. After the necessary preprocessing (s. 3.2 and 3.4) documents were clustered with the help of CLUTO.

To estimate the clustering performance, we first just went through the documents by hand to find out whether the clustering results were reasonable. This was a kind of manual evaluation. It was possible due to the relatively small size of the test set, which was also one of the reasons why we started our experiments with a small dataset. We examined all the documents that were classified as belonging to different clusters depending on the vector representation of these documents. Thus, e.g. emails were classified differently depending on the input vector representation. Emails represented by the traditional TFIDF vectors landed into the clusters different from those where they actually belong to as regards content, whereas when font-size and page weighting were applied, the same documents were ordered into different groups, which they were semantically closer to. Figure 4.3 shows an example of an email which was ordered to some arbitrary cluster when TFIDF was used; in case of using an improved vector representation it was assigned to another group which got the label 'seat test' (and that is what the email is about).

The manual examination of the clustering results was useful in the sense that afterwards we could at least be more or less sure that the algorithm is performing well.

Evaluation is essentially a problem for clustering if no manual content-based classification of the collection is available in advance. For the external cluster evaluation one needs to have a prior classification of the dataset. Though a kind of classification was available for the collection under consideration, this classification was based more on the document type and subtype (*email, report, etc*). Thereby, it was not possible to make a good external evaluation in our case, so we considered another simple but intuitive idea of what we can define as a sort of external evaluation. A mixed set of documents was compiled out of our test set and the documents from an absolutely different area (printer dataset). The


DocID: 6902584

TFIDF -> [driessen, sell ,britax]
- plus Schriftgewichtung & Seitengewichtung - > [seat, test]

Windmilling:
Partly you are right. For seats there are no windmilling tests required based on the results from the windmilling working group. Instead of this we use an AF = 2,5 and start with a calculation, but this must be done. For the fatigue analysis you calculate a maximum stress in a location of the main load path in comparison to the allowable stress and you show that it fulfills the requirement. You are also right if you say Airbus provides a template for the windmilling calculation. For this spreadsheet we have asked all seat manufacturer to send us the material data sheet in order to incorporate it into the table. The reaction from the seat manufacturer was very low so we think about an other way to go. Until this time create your own calculation. This is only valid for seats!!!!
For the other cabin furniture we have not such a task force and I try to go in contact with [redacted] as soon as he is back in the office (it seems tomorrow).

Best Regards

[redacted]

Figure 4.3: Document example

expectation was that the clustering algorithm, if it is good, would be able to differentiate between the two sets. This hypothesis proved to be true; over 97-100% of the documents were correctly classified dependent on the document vector representation. As expected, in this case the font-/page-weighting improved the clustering result. The best outcomes were achieved after the document vectors were pruned with both term-frequency and term-variance reduction techniques combined with font-size and page-weight feature weighting (100% accuracy).

In the end, we clustered the whole document collection (7305 documents). The vectors consisted of the lemmas of the words which occurred at least in two documents in the collection. Further they were normalized. Other kinds of pruning were not tested for this large set.

In this case, of course, it was not possible to check all the documents manually; hence we had to evaluate the results in some other way. Cluster validity is a measure of goodness of clustering performance relative to others created by

other clustering algorithms, or by the same algorithms using different parameter values. Cluster validation is an important issue in cluster analysis as the result of clustering needs to be validated in most applications. To measure the internal validity of the resulting clusters we use the average internal and external cluster similarity values.

Figures 4.4 and 4.5 show the average internal and external cluster similarity values for different numbers of clusters (K) for a collection of 7305 documents. Document vectors were only TFIDF-weighted in this case. Naturally, the intra-cluster similarity is getting better with the larger K. The problem is that the more clusters we have, the more complex is the resulting clustering for humans. The possible way out could be to present the information to the user in "layers", i.e. first to show only, say, 10 super-clusters, and after the user chooses the one she or he is interested in, the further zooming in can be offered. This can be done with bisecting K-means as one can easily build a hierarchy of documents with this method. What is also important is that the inter-cluster similarity stays approximately on the same pretty low level, which is good.

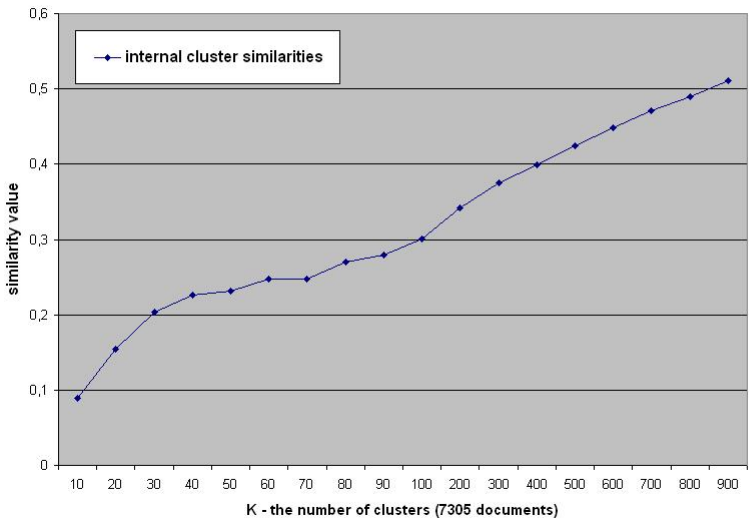


Figure 4.4: Average internal cluster similarities for different K

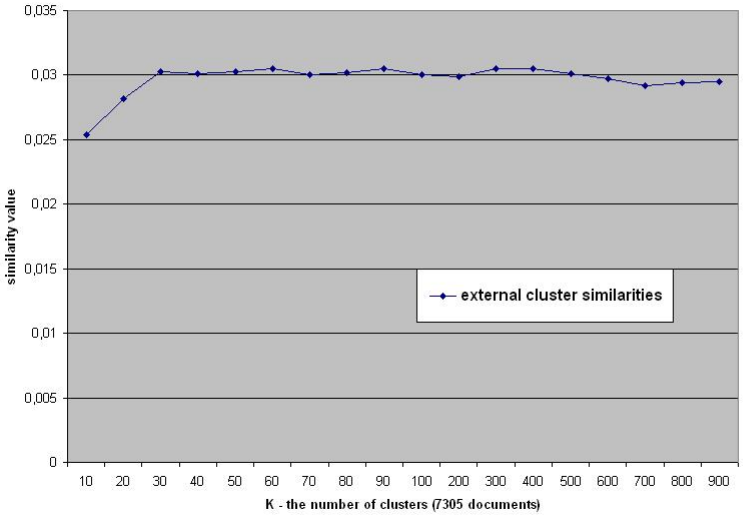


Figure 4.5: Average external cluster similarities for different K

We also tested different vector representation schemes with our large collection of 7305 documents. Figure 4.6 shows the resulting average cluster validity values for K=40. We can see here that:

1. at the first sight, the inclusion of noun phrases into vectors does not really seem to improve the performance in terms of internal similarity values, but, on the other side, we should point out extremely low external similarity values for all the splits, which implies that the inter-cluster similarity in these cases is basically absent!

2. surprisingly, the same outcome is observed for font-size and page weighting in this case, unlike the results we obtained with our small test set. This can be due to the still simplified vector representation we applied for this large collection in comparison to the more sophisticated vector representation of the test set. Moreover, the problem can lie also in the validity measure we applied for the internal evaluation in this case.

We need some value which can smooth out the internal and external cluster similarities as we feel intuitively and we know empirically from our test set ex-

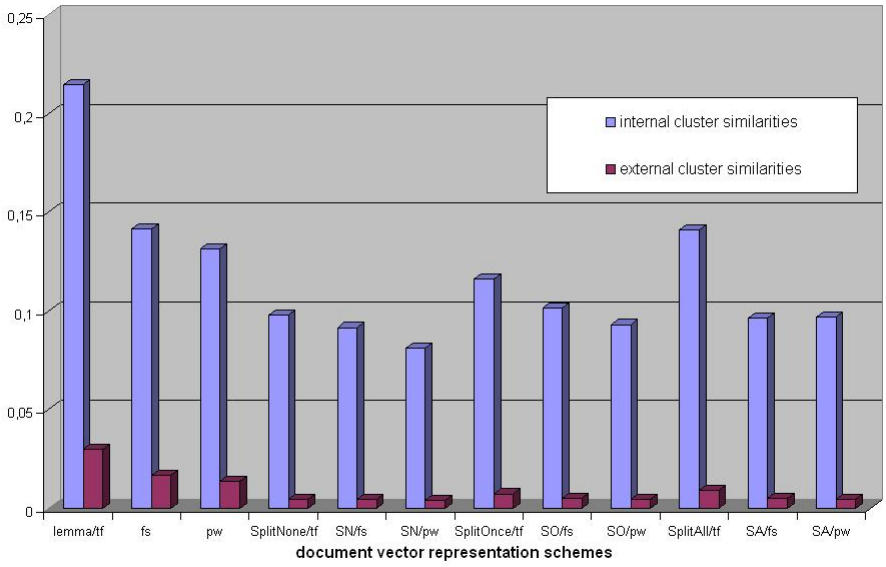


Figure 4.6: Average internal cluster validity values for K=40

periments that our advanced document representation should bring about better clustering results that we unfortunately can not induce from diagram 4.6. We could define, however, the correlation between internal and external cluster similarities with the help of the *H2* hybrid criterion function, which optimizes the global solution. Figure 4.7 demonstrates the *H2* values for the same experiment. The results are motivating as we can observe here definite improvement in the outcome in case of advanced document representations.

Anyway, further tests need to be conducted and, presumably, further cluster validity evaluation is necessary before making generalizations.

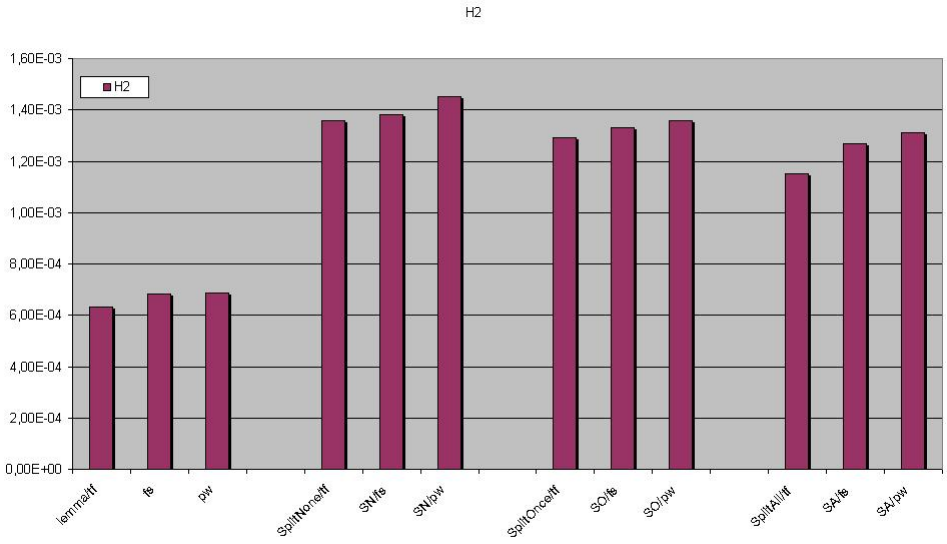


Figure 4.7: H2 for different document vector representations

4.2.3 Perspectives

An the end of our clustering module we get an automatic classification of documents into groups in the form with which our computers are "happy enough", but the resulting clusters look too complex for a human user, which is our ultimate goal. On the other side, it is difficult to interpret clustering results, i.e. to understand why exactly these clusters come into being and not other. Is the resulting grouping optimal for a human user? It is difficult to answer this question as there are different user groups, and what is good for one user may not be good for another as people have different requirements. A member of the knowledge management group and a mechanical engineer read surely differently one and the same document, consequently the content is being appreciated differently by them. The ontology-based clustering could facilitate both the interpretability of clusters and better labeling respectively, as well as some kind of "subjective clustering model". It requires, however, good ontology or even several ontologies (e.g. process- or product-based ontology), which are not available at the moment in the aviation industry. The construction of such ontology could have been the

next step in our framework, but it is out of the scope of this project.

The next stage in our framework is to make the available from clustering information feasible for humans. To bring the information into the user-friendly form one needs a good visual interface. Information visualization enables people to deal with huge amounts of information by taking advantage of our innate visual perception capabilities. By presenting information graphically we make it easier for the human brain to deal with vast amounts of available knowledge, performing it first to the perceptual system for preprocessing, rather than immediately relying entirely on the cognitive system (8).

4.3 The ASADO visualization module

As mentioned above, the motives and information needs during one retrieval session usually alternate. Consequently, tools providing both facilities for directed access and exploratory activity are expected to be most useful (17).
Moreover, an analysis of requirements revealed at least two distinct search scenarios relevant for our client (see 2.1): Exploratory activity is predominant in the "quality management" scenario, while the "Technical Problem Solving" scenario requires a directed, possibly meta–data enhanced search.
For a well-defined, directed search, ordered lists are a simple and effective solution. If the search area can be restricted with the help of document attributes, interactive meta-data filtering can be helpful in combination with other search techniques. Browsing and exploratory activity can profit from a display based on inter-document similarities.

For this reason, we decided to implement an **interactive application**, which integrates facilities for the above mentioned activities in one consistent interface. In the following, its main components will be presented. A demo version of the system is available at
`http://www.cogsci.uni-osnabrueck.de/~ASADO/visualization/`.

4.3.1 General interaction principles

One of the classical paradigms for interactive information visualization, which is especially well-suited for map interaction, is Ben Schneiderman's "Visual Information Seeking Mantra: Overview first, zoom and filter, then details-on-demand" (31) These four tasks constitute the fundamental interaction users expect from an interactive map:

- **Overview**: A zoomed out, coarse view of each variable is presented in the beginning to support quick orientation.
- **Zoom**: Once a region of interest has been identified, users typically want to examine it closer. Both a linear magnification or a non-linear fisheye distortion are popular. Smooth zooming improves keeping a sense of position and context.

- **Filter**: The user should be enabled to hide or disable uninteresting items. The filtering is often based on additional, not yet encoded variables. A rapid display update is the goal in order to indicate the effects of an action immediately.
- **Details-on-demand**: For a set of selected items, additional information should be made available on request. Usually this is achieved via popup windows, tooltips on mouse rollover or a separate details panel with a fixed position.

Further, some supplementary, secondary tasks can be identified (31):

- **Relate**: Enable the user to view relationships between items or compare items.
- **History**: Keep a history of user actions in order to support undo, replay and progressive refinement.
- **Extract**: Allow the extraction of subcollections and corresponding query and filter parameters for later re-use.

Additionally, if multiple views are provided at the same time, linking and brushing is the predominant technique to connect items across visualizations. A selection in one view will mark the selected items in all other views, thus allowing comparison among views and to easily combine the advantages of each offered visualization type.

Most of these principles have been implemented in our prototype. Due to resource constraints, the zoom principle was not integrated. An extension to cover this important task would not be difficult to integrate, however.

4.3.2 The framework

Based on the limited document set (e.g. retrieved by a keyword search or compiled to address a specific recurring retrieval task), the user is offered several possibilities to interact with the contained documents. The central module is a

scatterplot or map display. Several panels group around the visualization in order to support diversity of interaction and provide multiple selection and filtering modes. (see figure 4.8).

Figure 4.8: A screenshot of the ASADO system

4.3.3 Scatterplot

In a scatterplot, the displayed space is spanned by two axes with a pre-defined semantics. Accordingly, any point on the cartesian plan is associated with two variable values. This kind of visualization is both useful to inspect the co–distribution of two variables as well as for quickly filtering value ranges. In the example, two discrete–valued attributes are presented. The resulting rows and columns are scaled to indicate the size of the contained document set. Cells or single documents can be selected for closer inspection. The inner–cell document placement is based by sunflower seed packing algorithms to maximize inter–item space whilst maintaining a visual grouping (see Figure 4.9).

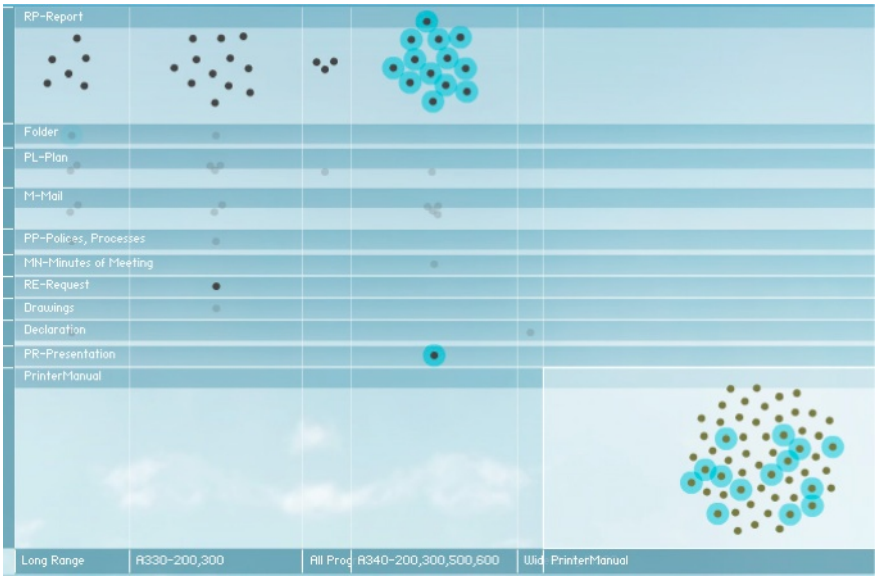


Figure 4.9: The scatterplot view

4.3.4 Map

The map view offers the facility to inspect the inter–document similarity structure (see Figure 4.10). As theoretical comparison and empirical tests revealed, PCA is best for coarse cluster structure inspection, while CCA and Sammon's Map preserve local topology better. Accordingly, both sets of coordinates are computed and the displayed coordinates are a linear mixture of these two coordinate components. The relative contribution of the two coordinate components can be adjusted by the user with a slider control (see Figure 4.11). This allows the user to blow up and shrink the clusters according to their needs.

Cluster are marked by cloud backgrounds. On rollover, automatically computed keywords are displayed to characterize the cluster contents. Single document items reveal their title as a tooltip on rollover (see Figure 4.12).

4.3.5 Additional panels

Around the visualization, several supplemental panels are available: Meta–data filters (see Figure 4.8; upper left) can be used to display only a subset of the

Figure 4.10: The map view

Figure 4.11: Slide control for coordinate mixture

retrieved document set based on meta–data filtering. Selected documents can be compiled in collections for reuse and comparison of different data sets (lower left). New maps should be calculated on–the–fly for these newly created collections; however, this is not implemented yet in the proof–of–concept prototype. A classical list view Ñ– coordinated with selection and filtering processes on the map and in other panels Ñ– enables a quick scanning of document titles. Additionally, sorting mechanisms could be implemented at this spot. A detail view

Figure 4.12: Cluster labels and document tooltips are presented on mouse rollover

(lower right) allows to cycle through all selected documents and retrieve detailed meta–data information or open the document for further inspection.

4.4 Mapping techniques

Besides the implementation of the prototype, much effort was put on the evaluation of different mapping algorithms. From a theoretical perspective, the calculation of a document map can be seen as an optimization problem. Given a formal representation of the quality of a certain coordinate configuration (the error or stress function), the task is minimize this function, resulting in the best possible map according to this criterion. We can distinguish two different methodologies in achieving this goal:

- Techniques like PCA, ICA and Isomap use an algebraic approach to solve the minimization problem. This puts some constraints on the class of computationally feasible error functions, but allows the one-shot calculation of an explicit projection function.

- Neural methods (like MDS and its variants) start with an initial configuration, which is gradually modified according to a heuristics until a stopping criterion is met. These algorithms can in principle optimize any differentiable error function. However, depending on the initial configuration, only a local optimum might be found. The mapping from input to output space is calculatedly only implicitly. Many variants exist, differing in error functions and optimization approaches.

Another important distinction can be made with respect to the capabilities of the algorithms:

- Linear methods like PCA can only apply a linear mapping to the data. Metaphorically speaking, the projection plane can be turned, scaled and skewed in document space, however, it will always remains ”rigid” .

- Non–linear methods techniques allow — to stay with the metaphor — ”elastic” maps which can lie folded, twisted or locally distorted in document space and are then unwrapped onto a plain cartesian map for display purposes.

Since PCA data was readily available from pre-processing, its usage was a computationally inexpensive option. PCA is a robust algebraic algorithm; its output allows an inspection of the coarse cluster structure of the data set. However, local distance relations are often not preserved well. Consequently, we decided to enrich the PCA coordinate information with information computed with the help of non–linear CCA and Sammon's map algorithms. These algorithms put more emphasis on the preservation of local distances, thus preserving the inner–cluster structure more faithfully. This way, the users can benefit from both approaches. In the interface, the amount of contribution of the respective algorithms can be regulated with a slider control (see figure 4.11).

4.5 Empirical results

To find the best–fitting visualization technique for the similarity maps, some empirical tests were conducted: A test set of 57 freely available printer documents was pre–processed by lemmatizing the contained words, discarding lemmata which occurred less than 5 times in all documents together and selecting the 3000 dimensions with the highest variance values. After that, values were weighted both according to font size and IDE value. A PCA dimension reduction on the centered data with a variance threshold of 0.0001 yielded 57 remaining latent dimensions. The first two principal components had a variance of 13.9% and 13,4% of the summed principal component variance, thus capturing together 27,3% of the total variance. Mutual dissimilarities in latent space were computed with the Euclidean distance measure. On this basis, CCA and Sammon's map were computed; both were initialized with the first two PCA coordinates in order to achieve quicker convergence and maintain comparability of the maps.

The resulting maps are plotted in figure 4.13.

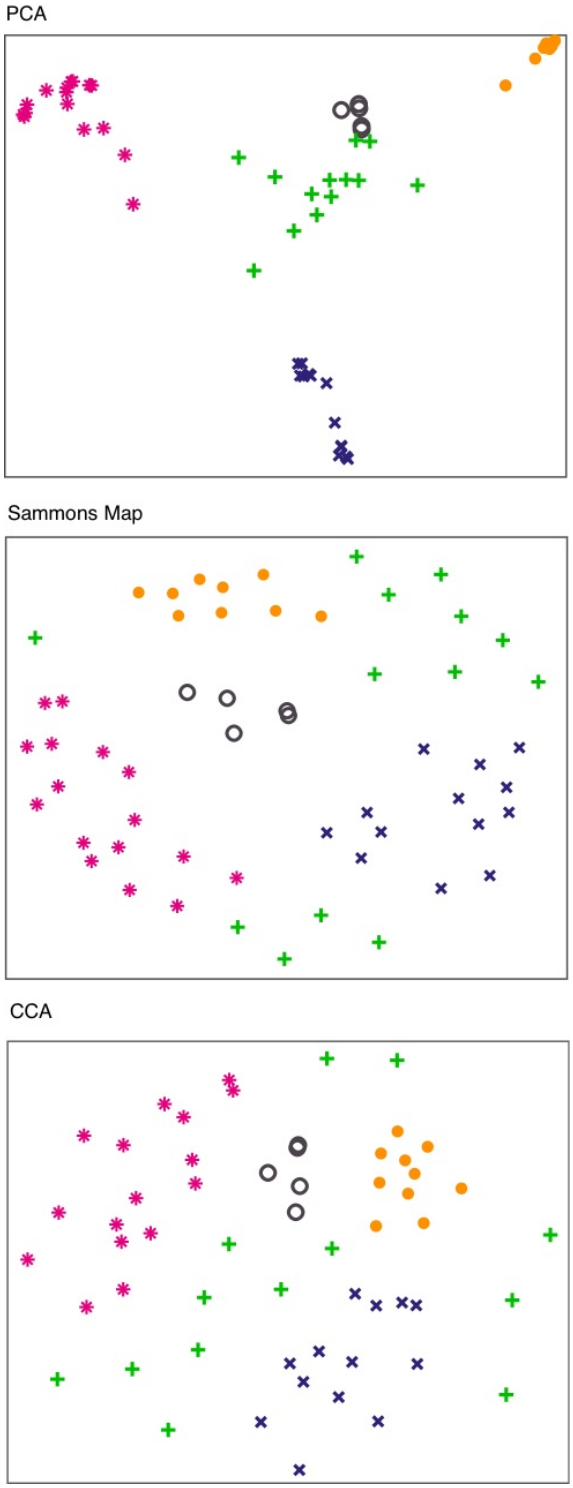


Figure 4.13: Test projections

Several observations can be made:

- In the PCA projection, many points are almost co–located, resulting in compact clusters with much whitespace in between. This loss in local topology results from the emphasis put on large distances and the globalist of the projection. In the following section, we will see that occasionally even very distant documents are be presented close together due to the neglect of a huge part of the available data.

- CCA and Sammon's map result in "blown up" clusters with more evenly distributed documents. From the visualization perspective, the available map space is used more efficiently, since clusters of similar documents will occupy more space than in a linear projection. In the Sammon's map, most cluster boundaries remain intact; in the CCA projection, clusters are not clearly distinguishable. Sammon's map tends to produce round structures with lower average distance towards the edges, while CCA produces a well–distributed map. Care has to be taken in visualization, however, to communicate the non–linear relationship of map distance and similarity. In extreme cases, cluster borders can get so close together that users might read high similarity between actually very dissimilar objects out of the map.

- The cluster marked with the green $\hat{\text{O}}+\tilde{\text{O}}$–symbol is located in the middle of the PCA plot. This indicates average values in both PCA components, which hints at a bad distinguishability from the rest of the documents. Consequently, in both CCA and Sammon's map, the cluster is torn apart and spread across the map. This situation is problematic as it might lead to wrong conclusions. On the other hand, also in the PCA plot it is counterintuitive that the least informative documents are located in the center. Only knowledge about the nature of PCA projection on side of the user can lead to the right interpretation, which cannot be presupposed.

These points are closely to connected to the quality evaluation of the computed coordinates. However, comparing the presented algorithms and their output is notoriously difficult, as each of the presented techniques highlights different aspects of the data.

One possibility for a coarse graphical inspection is to create a biplot of the original dissimilarities and the projected distances (see figure 4.14). Data points located, e.g. in the lower right corner of the plots indicate distances which are originally large, yet small in the projection, thus leading to a false indication of neighborhood relations. We can observe that there are significantly more of these points in the PCA plot, which hints at a low truthfulness of the projection for the involved documents. The CCA and Sammon's map plots do not exhibit dramatic differences, with a slight advantage for the Sammon's map as more points accumulate in the upper right.

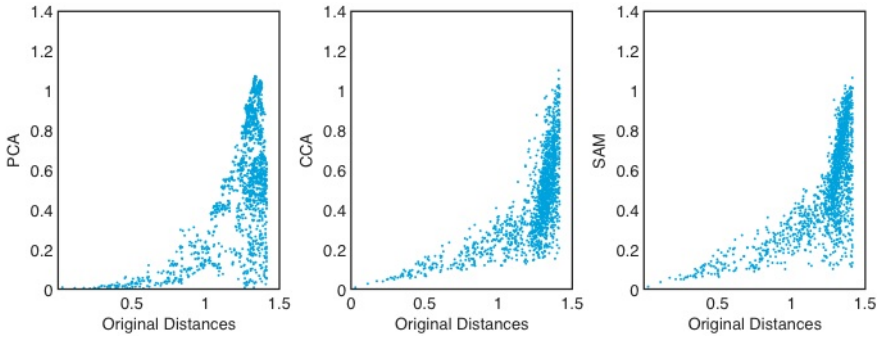


Figure 4.14: Biplots of original vs. projected distances

One option to quantify the mentioned "false neighborhood mistakes" to compare projection techniques is the trustworthiness measure. Essentially, it is computed by comparing the rank orders of projected and original distances in a neighborhood of certain size with respect to each projected item. For details about the measure, see (12).

The test confirmed the theoretical differences between the linear PCA projection and the non-linear MDS techniques Sammon's map and CCA. PCA has its strengths in feature extraction and data pre-processing, but should be used with caution for map displays. Based on the given data, the rather novel CCA technique seems to be slightly superior to the traditional Sammon's mapping due to its better run-time and efficient map space usage. However, further tests would have to be conducted to verify this impression.

4.6 Labeling

Even if the document maps can give us a rough orientation 'interpreting a [...] map proves to be difficult because the features responsible for a specific cluster assignment are not evident from the resulting map representation' (27). For this reason we need some textual guidance to navigate through document space. In our interface the document are presented as little icons on the document map. In a second window below the map the document titles are presented. This arrangement turned out to be not sufficient. According to Becks (1) users want to have a closer connection of document description and document map. Due to a lack of space the amount of textual information that can be provided is rather limited. The insertion of the titles into the map is therefore not a viable alternative. What is therefore needed is a small number of self-explaining terms that can provide some hints about the structure of the map and the distribution of the documents. Not the labeling of a single document is here the main issue but the labeling of document groups to emphasize their connecting elements. Lagus (18) proposes a labeling method for exactly that purpose, identifying the best keywords for a cluster of documents. 'A good descriptor of a cluster characterizes some outstanding property of the cluster in relation to the rest of the collection'. A potential keyword term has therefore to fulfill two properties. It should be prominent in the cluster compared to the other terms in that cluster. It should be prominent in the cluster compared to the occurrence of that term in the whole collection. One instantiation of these properties is the following formula.

$$G^0(w, j) = F_j(w)\frac{F_j(w)}{\sum_i F_i(w)}$$

: The quality of a term as a label is defined as the relative frequency of the term in the cluster times the relative frequency of the term in the cluster, divided by frequency of that term in the rest of the collection. The whole algorithm was implemented in matlab and could therefore be integrated directly into the whole process. The keywords identified by this method leave us with a rough idea what the single clusters are about. As the algorithm only relies on those terms that appear in a given cluster or document set, it has not possibility to abstract over the content of the documents. However the ability to find a common term that

is not part of the document is exactly what is needed to find a label for a set of documents. A good label for us is a term that subsumes the content of all documents. For this reason the method used here only can be a first step. We are currently working on an extension of the Lagus method that is capable to find the unifying concept.

Chapter 5

Summary and Conclusions

There are two types of tasks that are tackled in the ASADO project. On the one hand, two requirement studies have been carried out: The first one addresses among other things the actual program interface for the reading team members of the archive service who feed printed engineering documents into an electronic archive; the second one addresses the (visual) search interface which will be used by product safety members and engineers when looking for archived engineering documents. On the other hand, different program modules have been combined in order to build up a rather complex prototype for document management tasks: terminology extraction, clustering, cluster labeling and navigation in document collections. The aim of this prototype is to give a proof of concept that illustrates how, for an aviation manufacturer as Airbus, archiving and knowledge management could be optimized in the medium term.

The requirement studies mentioned above have proven to be of great importance for such a complex task as the one tackled in the ASADO project. When focusing all the technical problems that have to be solved in order to implement and integrate a great number of program modules, developers easily tend to ignore the original aim, namely to support the people who feed and use the engineering archive. Taking into account the results of requirement studies improves the acceptance of software by its end users. Furthermore, these studies seemed to be especially important for the ASADO project members. They knew very little about manufacturing airplanes. In general, such studies are expected to take place before designing a software system. But in the ASADO project, due

to organizational problems, these studies could only be prepared during the first half of the project and carried out at the beginning of the second half. At that time, a number of core components were already specified or even implemented in a first version. Therefore, the outcome of these studies had only a limited impact on the developed system. But, when we tried to identify the requirements of the archive end user, presenting a first version of the visual navigation tool turned out to be a helpful guidance for the discussion with representatives from engineering, product safety and archive service. It successfully triggered their motivation for cooperation. For this reason, detailed requirements with respect to different use cases could be detected.

The specified prototype turned out to be a rather complex system integrating a reasonable number of individual modules. Most of these modules are well established and easily available: OCR, tokenization, pos-tagging, lemmatization, clustering and a number of techniques for mapping vector spaces. The main task with respect to their integration consisted in putting into action their interfaces. Sometimes this turned out to be quite demanding and should not be underestimated with respect to the effort needed. Some other modules are designed, parameterized or even implemented by ASADO members themselves: language identification, chunk parsing, terminology extraction, different types of vector representations (e.g. one making use of document structure, another considering multi word terminology), cluster labeling and visual presentation of and navigation through document collections.

Some comments should be made on the quality of the engineering documents that were given to the project for analysis since these characteristics had an important impact on the research results.

First of all, the topic of the document set was restricted to airplane seats. Therefore, there was rather little content variety within the collection. That is, with respect to content, the set was quite homogeneous and content indicating clusters were not really obvious from the very beginning of the study. The great similarity of the clusters to be identified and displayed made huge demands on the clustering, mapping and presentation techniques applied. This problem, however, should loose its importance if larger aviation domains are dealt with.

Second, as expected by the project members, the vocabulary of the documents was very specific. On the one hand, such documents cannot be satisfactorily processed by existing programs that are tuned to standard vocabulary. But, in the medium term, this problem might be overcome by expanding the internal lexicons of these programs. On the other hand, some of the specific vocabulary is given by word composition and multi term units. And this peculiarity simplified terminology extraction.

Third, sometimes there were language switches within a document. This makes an additional module for language identification necessary.

Finally, most of the documents were quite short and their texts did not correspond to the expectations of the project members who wanted to make use of corpus linguistics for document analysis. Corpus linguistics typically deals with continuous texts but the documents to be analyzed contained significantly less continuous text than originally expected by the project members. There were many tables and sketches, and the documents often contained cover letters with distribution lists that were not really interesting with respect to the task to be done. Furthermore, those parts that were hand written could not be taken into consideration.

All these peculiarities led to the decision to fall back on additional texts not belonging to the aviation domain in order to evaluate the prototype during the project in a simple but convincing manner.

Despite the success and the impressive advantages of the ASADO prototype, it seems to be a non-trivial further step to develop a corresponding system that can be used professionally in a company. Among many other questions the following ones need to be addressed first: Are the developed techniques applicable to very large document sets? Are the developed techniques applicable to larger vector space dimensions? Can a system be maintained which is built-up from such heterogeneous modules?

A promising idea that might be pursued could be an incremental integration of (consolidated) ASADO modules into a professional system that is actually used by Airbus. Within this context vector representations might play a major role as interface.

One module that seems to have a very positive impact on a knowledge management platform and that might be integrated is the terminology extraction. Not only the exemplary terminology extracted so far is of high quality. The semi automatic techniques applied for obtaining this terminology proofed to be very efficient as well. It saves time and reduces the needed (expert) staff. Integrating terminology presupposes linguistic pre-processing of the archived documents.

Last but not least, it would be worthwhile to analyze in more detail whether the visual user interface might be integrated into an actually used knowledge platform as, for example, NetWeaver. Which steps have to be taken? How could an integration strategy look like? The ASADO visual and interactive interface presupposes vector representations of documents as well as clustering and labeling.

In order to answer these questions and to come closer to a system that is usable in a real life application environment a lot more work must be done. Nevertheless, the project delineates a first framework for the possibilities that might be considered.

References

[1] Becks, Andreas, Christian Seeling, and Ralf Minkenberg. 2002. Benefits of document maps for text access in knowledge management: a comparative study. In *Sac '02: Proceedings of the 2002 acm symposium on applied computing*, 621–626. New York, NY, USA: ACM Press.

[2] Canvyr, W. B., and J. M. Trenkle. 1994. N-gram based text categorization. In *3rd annual symposium on document analysis and information retrieval*, 161–175. Las Vegas, NV.

[3] Cécile, Frérot, Rigou Géraldine, and Lacombe Annik. 2001. Phraseological approach to automatic terminology extraction from a bilingual aligned scientific corpus. In *Conférence corpus linguistics*, 204–210. Lancaster.

[4] Dick, M., and T. Wehner. 2002. Wissensmanagement bei Airbus: Werkzeugentwicklung und die Kultivierung des Umgangs mit Wissen. project report.

[5] Indexierung zur inhaltlichen Erschliessung von Dokumenten. DIN 31 623.

[6] Flick, U. 1998. *Qualitative forschung.* Hamburg: Rowohlt.

[7] Friebertshäuser, B. 1997. *Handbuch qualitative forschungsmethoden in der erziehungswissenschaft*, chap. Interviewtechniken - ein Überblick, 371–395. Weinheim und München: Juventa.

[8] Geisler, G. 1998. Making information more accessible: A survey of information visualization applications and techniques .

[9] Hacker, W. 1987. *Software-ergonomie*, chap. Software-Gestaltung als Arbeitsgestaltung. Oldenbourg, München u.a.

[10] Hearst, A. Marti, and J. O. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*. Zurich.

[11] Herczeg, M. 2005. *Software-Ergonomie Grundlagen der Mensch-Computer-Kommunikation*. 2nd ed. Oldenbourg, München u.a: Wissenschaftsverlag GmbH.

[12] Himberg, J. 2004. From insights to innovations: data mining, visualization, and user interfaces .

[13] Hopf, C. 2004. *A companion to qualitative research*, chap. Qualitative Interviews: An Overview, 203–208. London: Sage.

[14] Jurafsky, Daniel, and James H. Martin. 2000. *Speech and language processing. an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall Series in Artificial Intelligence, Upper Saddle River, NJ: Prentice Hall.

[15] Karypis, G. 2002. Cluto a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota.

[16] König, B. 2002. *Qualitative sozialforschung - eine einführung*. Rowohlt.

[17] Lagus, K. 2002. Text Retrieval Using Self-Organized Document Maps. *Neural Process. Lett.* 15(1):21–29.

[18] Lagus, K., and S. Kaski. 1999. Keyword selection method for characterizing text document maps. In *Proceedings of icann99, ninth international conference of artificial neural networks volume 1*, 371–376. London: IEE.

[19] Lamnek, S. 1998. *Gruppendiskussion.* Weinheim: Beltz.

[20] L'Homme, Marie-Claude. 2002. What can verbs and adjectives tell us about terms? In *Terminology and knowledge engineering, tke 2002*, 65–70. Nancy (France).

[21] Ludewig, Petra. 2005. *Korpusbasiertes Kollokationslernen. Computer-assisted language learning als prototypisches Anwendungsszenario der Computerlinguistik*, vol. 9 of *Computer Studies in Language and Speech.* Frankfurt a.M., Berlin, Bern, Bruxelles, New York, Oxford, Wien: Peter Lang.

[22] Maedche, A., and S. Staab. 2001. Learning Ontologies for the Semantic Web. In *Semantic web workshop.* Hongkong.

[23] Montello, D. R., S. I. Fabrikant, M. Ruocco, and R. S. Middleton. 2003. Testing the First Law of Cognitive Geography on Point-Display Spatializations. *Spatial Information Theory. Foundations of Geographic Information Science, International Conference, COSIT 2003, Ittingen, Switzerland, September 24-28, 2003,Proceedings* 2825:316–331.

[24] Nielsen, J. 1993. *Usability engineering.* Boston: AP Professional.

[25] Pirolli, P., and S. Card. 1998. Information foraging models of browsers for very large document spaces. *Proceedings of the Advanced Visual Interfaces Workshop, AVI '98* pp. 83–93.

[26] Terminology work principles, and methods. ISO 704 International Standard.

[27] Rauber, Andreas, and Dieter Merkl. 1999. Automatic Labeling of Self-Organizing Maps: Making a Treasure-Map Reveal Its Secrets. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 228–237.

[28] Reuer, V., P. Ludewig, C. Rollinger, and K. Krüger-Thielmann. 2003. Studienprojekte in den Bereichen Computerlinguistik und Cognitive Science. *SDV - Sprache und Datenverarbeitung - International Journal for Language Data Processing Band* 27(1/2):185–202.

[29] Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *International conference on new methods in language processing*. Manchester, UK.

[30] Schuler, H., ed. 2004. *Lehrbuch Organisationspsychologie*. Dritte, vollständig überarbeitete und erweiterte auflage ed. Bern: Verlag Hans Huber.

[31] Shneiderman, B. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *IEEE Visual Languages (UMCP-CSD CS-TR-3665)* 336–343.

[32] ———. 2001. *User Interface Design*. Bonn: mitp-Verlag.

[33] Skupin, A. 2000. From Metaphor to Method: Cartographic Perspectives on Information Visualization. *Proceedings of InfoVis 2000 (Salt Lake City UT)* 91–98.

[34] Steinbach, M., G. Karypis, and V. Kumar. 2000. A comparison of document clustering techniques. In *Kdd workshop on text mining*.

[35] Thaller, G. E. 2002. *Interface Design- Die Mensch-Maschine-Schnittstelle gestalten*. Frankfurt: Software & Support Verlag GmbH.

[36] Terminology work Vocabulary. ISO 1087-1 International Standard.

[37] Zhao, Y., and G. Karypis. 2001. Criterion functions for document clustering: Experiments and analysis. Technical Report 01–40, Dept. of Computer Science, University of Minnesota.

Online Resources

[2] W. B. Canvyr and J. M. Trenkle. N-gram based text categorization. In *3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, NV, 1994. 32

[@IM] Institut für maschinelle sprachverarbeitung [online, cited September 2005]. Available from: http://www.ims.uni-stuttgart.de/.

[@IS] Information technology – 8-bit single-byte coded graphic character sets – part 1: Latin alphabet no. 1 [online, cited September 2005]. Available from: http://www.iso.org. see also: http://en.wikipedia.org/wiki/ISO_8859-1.

[@MS] Microsoft Office online [online, cited September 2005]. Available from: http://office.microsoft.com.

[@NL] Natural language toolkit [online, cited September 2005]. Available from: http://nltk.sourceforge.net.

[@TC] Textcat: An implementation of the text categorization algorithm presented in (2) [online]. Available from: http://odur.let.rug.nl/~vannoord/TextCat/.

[@TI] Tagged Image File Format (TIFF) resources [online, cited September

2005]. Available from: http://partners.adobe.com/public/developer/tiff/index.html.

[@TR] TreeTagger: A language independent part-of-speech tagger [online, cited September 2005]. Available from: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html.

[@UN] Unicode home page [online, cited September 2005]. Available from: http://www.unicode.org.

[@XM] Extensible markup language (XML) 1.0 (third edition) [online, cited September 2005]. Available from: http://www.w3.org/TR/REC-xml/.