Navigating the Web with GERHARD

Gerhard Möller[1], Kai-Uwe Carstensen[2], Bernd Diekmann[3], and Han Wätjen[3]

[1] Oldenburger Forschungs- und Entwicklungsinstitut für Informationswerkzeuge und -systeme (OFFIS), D-26121 Oldenburg, Germany, moeller@offis.uni-oldenburg.de
[2] Institut für Semantische Informationsverarbeitung (ISIV), Universität Osnabrück, D-49078 Osnabrück, Germany kai.carstensen@cogsci.uni-osnabrueck.de
[3] Bibliotheks- und Informationssystem (BIS) der Carl von Ossietzky Universität Oldenburg, D-26015 Oldenburg, Germany {diekmann,waetjen}@bis.uni-oldenburg.de

Abstract. GERHARD is a fully automatic indexing and classification system of the German World-Wide Web for integrated searching and browsing. A database-driven robot collects academically relevant documents, which are automatically classified with computer-linguistic and statistical methods using the Universal Decimal Classification. The generated metadata and the index of the documents are held in a relational database (Oracle with Context option). The user-interface is trilingual (German, English, French) and allows the user to look for "similar" documents very easily through its tight integration of searching and browsing mechanisms.

1 Introduction

Search-services in the World-Wide Web are in a state of crisis. Queries to index-based search-engines like Altavista most often result in a high recall but very bad precision. The reasons for this are manifold. For the most part it is due to poorly made queries, another reason is that the quality of the pages vary widely. Although there is effort to find better ranking criteria the general problem remains: the translation problem between the user needs and the verbalized request.

To cope with this problem, Web directory services like Yahoo! and excite were built. However, as the classification is done intellectually, they are hopelessly swamped by the huge amount of documents and their very small half-life period. Only very specialized services like the Engineering Electronic Library, Sweden, have at least a chance to cover a broad part of the relevant documents.

Another problem with most directory services is that they are based on an ad-hoc created classification scheme. But in general the quality of

classification strongly depends on the intellectual effort spent in establishing the scheme itself. If it does not cover the subject area completely, some aspects cannot be categorized. If inconsistencies exist in the system, contradictions in the classification will be the result. This is why legions of highly qualified librarians have spent many decades building and maintaining such classification systems.[1]

Finally, the existing Web directory services fail to integrate their searching and browsing facilities such that a user can directly see all the categories assigned to a document to then find all other documents that have been assigned the same or similar categories.

GERHARD's (German Harvest Automated Retrieval and Directory) approach to the above problems is to

- use a database-driven gatherer (Sect. 4),
- use fast automatic classification (Sect. 5),
- use a professional classification system (Sect. 2), and
- integrate the searching and navigation service (Sect. 7).

The following chapters describe the components of the system and first experiences with the service. Of course, there are other projects related to GERHARD, which will be discussed first.

1.1 Related Projects

Next to GERHARD there exist several other projects related to automatic classification. One of the first was probably the Nordic WAIS/WWW Project[2] [1]. However, this project was quite limited, as it used only 51 UDC entries for classification and classified only approx. 700 very homogeneous descriptions about WAIS databases.

A very extensive project in that area is the OCLC Scorpion[3] [10], which uses the Dewey Decimal Classification (DDC) [12], [2], a purely hierarchical classification system. Scorpion does not offer a Web-catalogue, but is rather a research project exploring the use of automatic classification with various methods. Like GERHARD Scorpion uses linguistic and statistic methods, which makes the two projects most related to each other. (This is true for their problems and shortcomings as well...)

[1] As have many philosophers...
[2] http://www.ub2.lu.se/auto_new/UDC.html
[3] http://orc.rsch.oclc.org:6109/

Web directories that use DDC are for example BUBL Link[4] [3] and CyberDewey[5]. However, the classification is done intellectually, so only very few documents are available and there is no integration between searching and browsing facilities. Actually, none of the above services offer integrated browsing and searching.

There is also the EU-project Desire II, which is in planning stage. One of their goals is a "prototype service providing automatic classification of Engineering resources" [6], the date for finishing is set to February 2000. First contacts to the developers of GERHARD and Scorpion have already been made.

2 The Classification System

The classification system can be seen as a material ontology offering a very valuable store of knowledge. It defines which categories exist and how they are related to each other. In a well structured ontology, the experienced user can find quickly the needed information and the novice user can learn quite a bit about the subjects of interest. However, there are three additional preconditions to be met:

1. As the domain of the expected documents is likely to be very heterogeneous, specialized classification systems as ACM Computing Classification System or the Engineering Information Classification System[6] cannot be used. Instead, a general system has to be used.
2. The target of GERHARD is the German World-Wide Web, so at least German has to be supported. On the other hand, many documents are supposed to be in English language, so English should be supported as well.
3. Finally, the classification system has to be available in electronic form.

Neither the Dewey Decimal Classification (DDC), the Library of Congress Classification (LCC), the Netherlands Base Classification nor any other classification system of German Libraries meet all three preconditions.

[4] http://bubl.ac.uk/link/

[5] http://ivory.lm.com/~mundie/CyberDewey/CyberDewey.html

[6] Examples of intellectually generated Web directories that utilize the ACM and EI classifications are Ariadne (http://ariadne.inf.fu-berlin.de:8000/) and EELS (http://www.lub.lu.se/eel/).

The only classification system found suitable is a special version of the Universal Decimal Classification (UDC) [4] modified and extended by the ETH Zürich [8], in the following called "UDCZ".

It consists of approx. 60,000 entries (*categories*), which are connected to each other by 15 different relations[7] [9]. Each entry consists of a DC-number (*notation*), the associated descriptions and synonyms in German, English, and French, and, if necessary, explicit references to prior used[8] and related notations.

The general structure of the UDC is a hierarchy resulting from the structure of decimal numbers, i.e. "51" ("mathematics") is a hypernym of "511" ("number theory") as shown in Table 1. The other relations between UDC-entries are denoted by special characters, e.g. "669.215'22" ("gold-silver-alloy") is a combination (') of "669.215" ("gold-base alloy") and "669.22" ("silver"). Multiple relations like "669.15'255'245" ("iron-cobalt-nickel alloys") are possible as well. This results in a directed graph with cycles, where the UDCZ-entries represent the nodes of the graph.

Table 1. Right truncation specifies hypernomy

5:	"mathematics/natural sciences"
51:	"mathematics"
511:	"number theory"
511.5:	"diophantine equations (number theory)"
511.57:	"forms of higher degree"

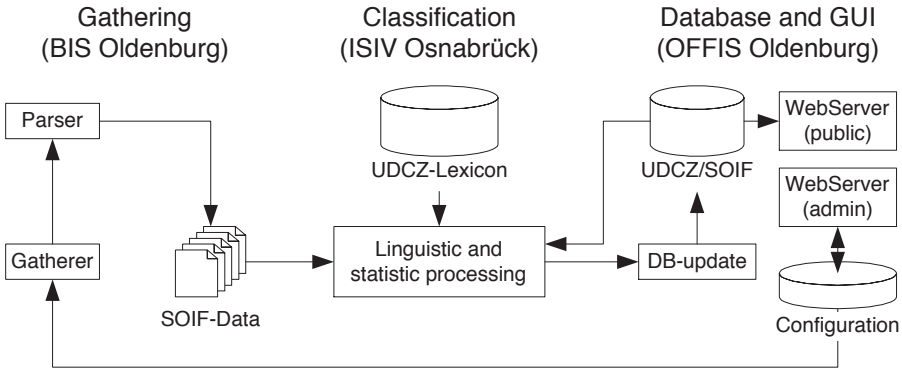
3 System-Architecture

A rough system architecture and allotment of tasks to the project partners is shown in Fig. 1, the next sections describe the components in more detail. All components are distributed and can have multiple instances to balance the load.

[7] Hyper-/hyponym, association, extension, combination, unsolvable connection, relation, supplement of language, supplement of form, supplement of location, supplement of peoples, supplement of time, special supplement, point of view, expansion, and partition.

[8] At several places existing notations had been replaced for more consistency.

Fig. 1. System-architecture and project teams of GERHARD.



4 Gathering

To collect the Web documents for classification and indexing, the Harvest Gatherer [5] is used[9]. It uses starting points and filtering rules to define which documents have to be collected. To be able to define the search-space on the fly, all configuration data is held in a database that can be accessed and modified via a Web-interface.[10]

The current search-space covers the academically relevant sub-space of the German Web, which consists of more than 400 universities, high schools, cultural, political, and scientific institutions, etc.

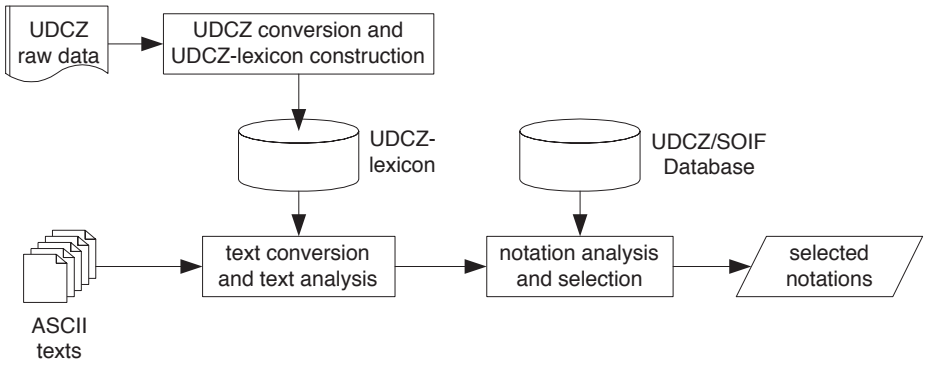
After collecting the documents they have to be analyzed and prepared for further processing. GERHARD utilizes an adapted version of the Harvest Summarizer, which parses the document and stores it in a structured form, called Summary Object Interchange Format (SOIF) [5]. A general problem is that if the document source contains poorly coded HTML, the resulting SOIF can have semantic errors.[11]

[9] It turned out that the Harvest Gatherer is not efficient enough for our needs and takes too many resources. Therefore, it will soon be replaced by Combine, a web-robot that was developed by NetLab, Lund for the EU-project Desire.

[10] It is planned to make parts of this interface available to the public too check the status of the gatherer and submit places of possible interest.

[11] For example some authors use the `<author>`-tag instead of the `<em>`-tag to print emphasis. The resulting SOIF obviously contains nonsensical values for the attribute “Author”.

Fig. 2. Architecture of the linguistically based classification



5 Automatic Classification

Classification in GERHARD pursues a pragmatic approach, albeit with the use of sophisticated linguistic technologies. It is governed by two general demands: *Maximum quality and Minimum time consumption*.

In order to assure high precision and recall in later navigation and search, the quality of the classification result should be optimal. This rules out simple pattern matching and statistics approaches which in the face of the UDCZ's complexity do not reach the quality level aimed at.

Given the huge amount of gathered text data, however, good quality may not be achieved through extensive processing of the texts to be classified and/or the application of high-level computational linguistic methods. Instead, texts have to be processed with minimum time costs.

We have solved this problem of maneuvering between these antagonistic requirements by a division of labor in the processing tasks. The key idea is to transform the UDCZ into a lexicon mapping natural language expressions on UDCZ-notations, using sophisticated linguistic analysis tools at compile time, and to analyze the texts to be classified using an efficient recognizer built from this lexicon.

The architecture of the linguistically based classification is shown in Fig. 2. There are three main components, UDCZ-conversion and UDCZ-lexicon construction, text conversion and analysis and notation analysis and selection, with only the latter two operating at run time of classification.

5.1 UDCZ-Conversion and UDCZ-Lexicon Construction

The raw data of the UDCZ consists of 27 MByte text data with ca. 500000 lines of text dumped from the database of the ETH Zürich. Each entry (see Table 2 for an example) contains among other things the information about a category marked by a UDCZ notation and natural language descriptions of the category in German, English, and French.

Table 2. Example of a raw data entry of the UDCZ

```
001Z ~03
002DDUEBERSETZUNGEN / TECHNISCHE U. NATURWISSENSCHAFTLICHE
003DETRANSLATIONS / TECHNICAL AND SCIENTIFIC
004DFTRADUCTION / SCIENTIFIQUE ET TECHNIQUE
```

The aim of the UDCZ conversion is to extract/generate natural language expression that may occur in texts. As Table 2 indicates, this requires elaborate processing of the linguistic material available in the UDCZ.

Ignoring aspects of necessary automatic editing, conversion proceeds in three steps:

1. Morphological analysis[12] of each word in the UDC entry, reduction to its stem (if differing, to both singular and plural word stems), and annotation with word class information.
2. Application of rules sensitive to the available linguistic information in order to extract or construct well-formed natural language expressions.
3. Deletion of annotations as well as stopwords[13].

For the German part of the example in Table 2, the result of conversion are presented in Table 3, showing the stems and tagged word forms (first line) and the two natural language keys constructed (second line).

Each word stem is implicitly suffixed with a variable so that a match with specific word forms during text classification is possible (e. g. “technische uebersetzungen”, but also “technischer uebersetzungsvorschriften”).

[12] Lingsoft’s (http://www.lingsoft.fi/) programs GERTWOL and ENGTWOL are used for this.

[13] Gathered from the CELEX database of the Max Planck Institute, Nijmegen (http://www.kun.nl/celex/).

Table 3. Result of UDCZ-conversion

```
uebersetzung~~S/technisch~~A u.~~ABK naturwissenschaftlich~~A
technisch uebersetzung / naturwissenschaftlich uebersetzung
```

Although these truncation variables are useful in general and lead to flexible matches, they overgeneralize in the case of short words and result in false matches. For example, “gene” would match “general”, “generic” etc.

Depending on its length, we therefore generate the list of all possible morphological endings of a word, which leads to the differences in Table 4 (with “-” indicating arbitrary endings, and “xxx” indicating that the stem itself is a word form).

Table 4. Sample UDCZ-lexicon entries

```
technisch uebersetzung:-:~03
gene:xxx s:575.113.1
```

Table 4 is an example of entries in the UDCZ-lexicon, which maps natural language expressions to notations of the UDCZ. This lexicon is compiled into a recognizer, that is, a finite state automaton which accepts instances of the regular expressions implicit in the lexicon and outputs the corresponding notations.

5.2 Text Conversion and Analysis

The texts to be classified have first to be adjusted to the standard set by the UDCZ-lexicon (regarding umlauts, removal of stopwords etc.). After that, the recognizer is applied iteratively to a given text, cutting off prefixes accordingly. Text analysis thus yields a bag of notations as a result, basically found by matching strings in the text with entries of the UDCZ-lexicon. It should be noted that the recognition of multi-word strings (corresponding to very specific and unambiguous notations) combined with the flexibility and specificity of textual matches is a special feature of this component and represents an important advantage of GERHARDs classification, as compared to approaches using simple stemming algorithms and single-word database look-ups.

5.3 Notation Analysis and Selection

Selection of the relevant notations from the bag of all found is done in two steps:

1. Exploitation of the information given in a notation, its frequency of occurrence, and its textual match to find salient clusters of notations.
2. Further statistic and heuristic processing using lookups in the UDCZ-database to further reduce and weight the found notations.

Exploitation of Notations. Notation analysis involves the following aspects. First, the hierarchical information coded in UDCZ notations is exploited. As Table 1 in Sect. 2 shows, information coding in general follows the principle of right truncation for specifying the superclass relation.

According to that, the longer a notation is, the more specific is the category it codes.[14] Although this principle is not used consistently in the UDCZ [8], the available hierarchical information is sufficient for the purpose of classification in GERHARD. So each occurring prefix of a found notation is inspected and its relative importance, given its absolute frequency, is computed. This is implemented by sorting the bag of found notations into a tree of characters, where each node in the tree implicitly codes the information about a prefix.

Second, the length of a textual match is considered with the assumption that the longer the match according to the UDCZ-lexicon (Table 4) of a category is, the more specific it will be inside its sub-tree (e. g. because the probability of ambiguity is reduced).

Both of these aspects are extensively used in the algorithm for selecting a notation. Traversing the character tree down to a depth d, a notation is considered as relevant if the sum of its prefixes' maximum "match-lengths" weighted by the depth of the prefix exceeds a relative threshold t (1). This identifies notations of salient clusters with maximum specificity.

$$\sum_{i=1}^{d} \frac{i \cdot \mathrm{N}_{\mathrm{matches}}(\mathtt{not}[i]) \cdot \mathrm{l}_{\max}(\mathtt{not}[i])}{\mathrm{N}_{\mathrm{notations}}} > t \qquad (1)$$

[14] This is not necessarily true for all relations in the UDCZ especially for the expansion, e. g. "321.1/.8" ("forms of government") is more general than "321.15" ("democracy, theocracy, aristocracy, oligarchy, patricianism in antiquity"). Fortunately, those exceptions are rare.

where		
	d	depth of analysis (set to average notation length)
	`not[`i`]`	character of notation at position i, also node in the notation character tree
	$\mathrm{N}_{\mathrm{matches}}(\texttt{not}[i])$	number of matches with a prefix up to `not[i]`
	$\mathrm{l}_{\mathrm{max}}(\texttt{not}[i])$	length of longest match found for notations with prefix up to `not[i]` / average match length
	t	(threshold) average match length / average notation length
	$\mathrm{N}_{\mathrm{notations}}$	number of found notations.

Statistical Post-Processing. The above classification process results averaged in 14 notations per document. In a last step they are weighted, and only the best six to eight of them are assigned to a document. This statistic and heuristic post-processing consists of the following steps:

1. The documents are checked for being a doublet by comparing the MD5-checksum and the title of the document with the entries in the database.
2. The title of the document is checked against an exclusion list. Certain documents, e. g. Web-server statistics are excluded from classification and indexing.
3. The title, headings and body of the document are classified separately. A total sum of max. 499 notations including max. 9 precedented notations from the title are held in a bag of unique notations $\{\mathrm{not}\}_i$.
4. A primary quality $q(not_i)$is calculated:

$$q(not_i) = 1 + 10 \cdot \left(\frac{\mathrm{N}_{\mathrm{matches}}(not_i)^{\alpha_1}}{\mathrm{N}_{\mathrm{words}}} \right)^{\alpha_2} \tag{2}$$

   where α_1 and α_2 are constant values, $\mathrm{N}_{\mathrm{matches}}$ the number of matches of notation not_i, $\mathrm{N}_{\mathrm{words}}$ is the number of words in the document.
5. The quality $q(not_i)$ is corrected by

$$q(not_i) = q(not_i) + \sum_{l=1,L}^{l} f_l(not_i) + \sum_{l=1,R}^{l} h_l(not_i) \tag{3}$$

   where $f_i(not_i)$ are corrections depending on the notation itself, e. g. the quality geographic subjects are decreased, and $h_i(not_i)$ are corrections which respect the hierarchical context of the bag of notations:

$$h_l(not_i) = \gamma_l(\mathrm{R}, \mathrm{H}\{\mathrm{not}\}_{j \neq i}) \tag{4}$$

If not_i has the relation H to another element of $\{\text{not}\}$, h_l is a definable value. I. e. if we have the notations "physics" and "quantum physics" in $\{\text{not}\}$, the quality of "quantum physics" is increased and the quality of "physics" is decreased.

6. The quality of the notation not_i is weighted depending on the appearance of the matching terms.

$$q(not_i) = q(not_i) \cdot \sigma(not_i) \tag{5}$$

where $\sigma(not_i)$ depends on the part (title, headings, body) of the document where not_i matched.

7. A threshold is applied to reduce the number of notations to an average of six to eight.

6 Database

As seen in Sect. 2 the data-structure describing the UDCZ is a directed graph with cycles and labels on its nodes (categories) and edges (relations). Each node is assigned an arbitrary number of SOIF-data records, consisting of structured (e. g. URL, date) and unstructured data (e. g. authors, full-text).

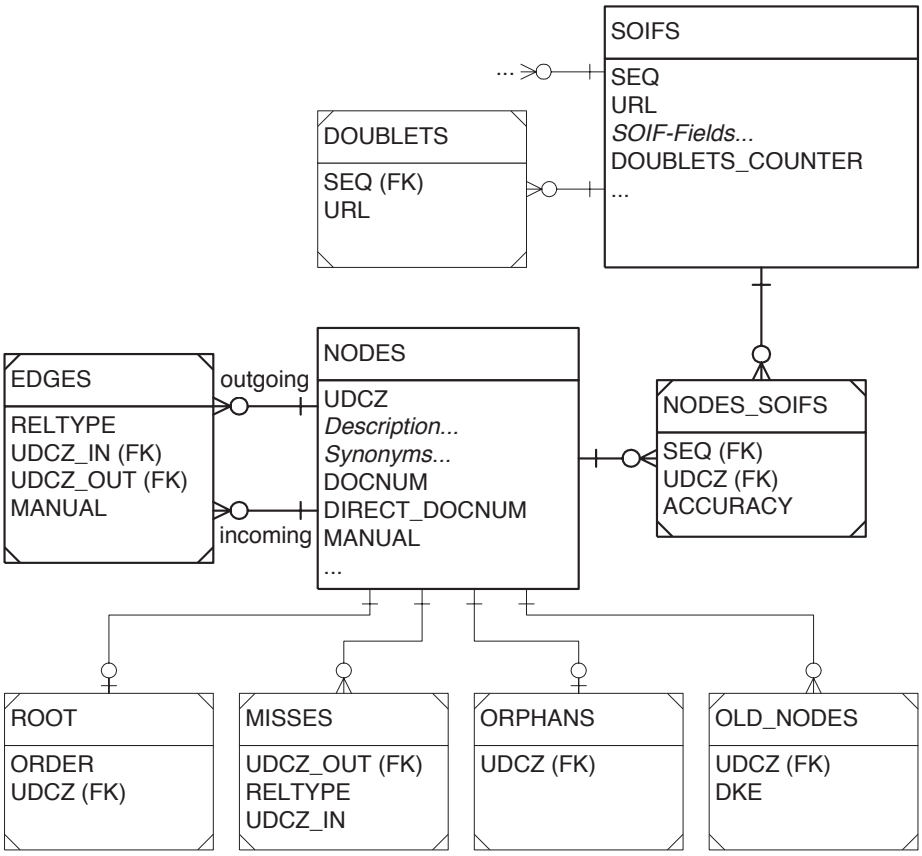
For the structured data a relational database system is suited very well, however, for unstructured data an IR-system or a RDBMS with IR-functionality has to be used. (Alternatively, an IR-System that can handle structured data could be used as well.) Several systems were evaluated, namely free-WAIS-sf, mSQL, Postgres, Fulcrum, and Oracle. The first prototype was built with Postgres95 in conjunction with Perl5. Unfortunately this solution proved very soon to be far too inefficient. Finally, Oracle7 with the ConText option[15] and the WebServer 2.1 was chosen. Reasons are good integration of structured and unstructured data, scalability, performance, and Web-integration.

In Fig. 3 the general ERD[16] of GERHARD is shown. `EDGES` and `NODES` represent the UDCZ-graph, to which the `SOIFS` are connected through `NODES_SOIFS`, including the `ACCURACIES` as calculated by the classification component (Sect. 5).

[15] ConText offers IR-Functionality on columns.

[16] Relations for maintenance, statistical analysis, gathering processes, etc. are omitted.

Fig. 3. Partial ER-Diagram of the UDKZ and SOIFS (simplified)



The attributes in the relation `SOIFS` are essentially the according fields of the SOIF-definition.[17] As approx. 30% of all Web-documents are duplicates [11], doublet-checking is essential for efficiency. Therefore only the URLs of duplicate documents are stored in `DOUBLETS`, the counter in `SOIFS` speeds up the browsing and the deletion of SOIFs.

The `NODES` include in addition to their description and synonyms in three languages among other things a cross-reference to prior used notations (`OLD_NODES`), if applicable. The starting point entries for the navigation are held in `ROOT` to be able to change them on the fly.

[17] As ConText allows only one searchable field for each table, the relation had to be splitted into several tables not shown here.

As said in Sect. 2, the relations between UDC-entries are only implicit. This implicit information was extracted and written into `EDGES` with a parser, written in PL/SQL. Although there are rules for the allowed ways of "generating" UDC-entries, not all librarians who created UDCZ-entries seem to have followed them, so the parser uses some heuristics to extract as many edges as possible. Still, only approx. 90% of the edges have been found up to now.

Another problem is that many UDCZ-entries are build from other UDCZ-entries that do not exist in our data. This would result in a disjointed graph, leaving some specialized categories unreachable by navigation. To avoid this, those cases are documented in `MISSES` and in `EDGES` the next available entry found with recursion is stored. Still, some categories[18] and relations had to be entered manually, which is indicated in the attributes `MANUAL` in `NODES` and `EDGES` respectively.[19]

SOIFs and classifications are sent from the statistical post processing via a named pipe to the Oracle SQL*Loader, which stores them in several tables, depending on the information whether it is a duplicate, a new entry, an update or an entry to delete. A PL/SQL program then inserts, updates or deletes records, regard being had to the duplicates.[20]

7 User Interface

The development of the user interface[21] was driven by simplicity and performance. The main functions "browsing the directory", "searching the directory", "searching the documents", and a context-sensitive online help are always available on the left frame[22].

In navigation the description of the active category as well as the descriptions of its super- and subcategories[23] are shown in an indented ta-

[18] Taken from [4].

[19] To be able to update the UDCZ data.

[20] This is the fastest way to fill a database, unfortunately, the SQL*Loader can neither update or delete rows directly.

[21] For a more detailed description of the user interface including the administrative issues with screenshots refer to [13] and [14] or http://www.gerhard.de.

[22] We are aware of the problematical nature using frames and Javascript (which is being used for the context-sensitivity of the online help and to indicate what function is active), however this decision was made to increase efficiency. After all, Javascript can be switched of safely, loosing context-sensitivity.

[23] A "subcategory" is a category that can be reached directly by the foreign key `UDCZ_OUT` of a `EDGE` connected to the category by `UDCZ_IN`. Vice-versa for "super-categories". In gen-

ble. Next to each category-description it is shown how many documents are assigned to the transitive envelope of all subcategories and how many documents are assigned to the category itself.[24] To keep the browsing as simple as possible, only those categories are shown that actually lead to documents either directly or indirectly.[25] Of course, searching for categories is possible.

Fig. 4. Navigation in the UDCZ-categories

Clicking a description of a category browses the directory, clicking a document-symbol next to a category-description enters an overview of all documents assigned to the category, ranked by the accuracy[26] of the classification. As can be seen in Fig. 5, both German and English documents can be classified correctly.[27]

The result of a full-text search in the documents is a similar overview as in Fig. 5. Clicking on the title of a document opens it, clicking on a document-symbol in the overview shows a detailed view (Fig. 6) of the document.

eral, a super-category defines a "more abstract" category and a subcategory defines a "more specialized" category. The "direction" of the relations of the UDCZ can be found in [9].

[24] `DOCNUM` and `DIRECT_DOCNUM` in `NODES` (Fig. 3).

[25] As can bee seen in Fig. 4 it can happen that a category shows up that has no document assigned to itself or the transitive envelope of its subcategories, but still is selectable ("chemorheology").

Fig. 5. Overview of found assigned documents

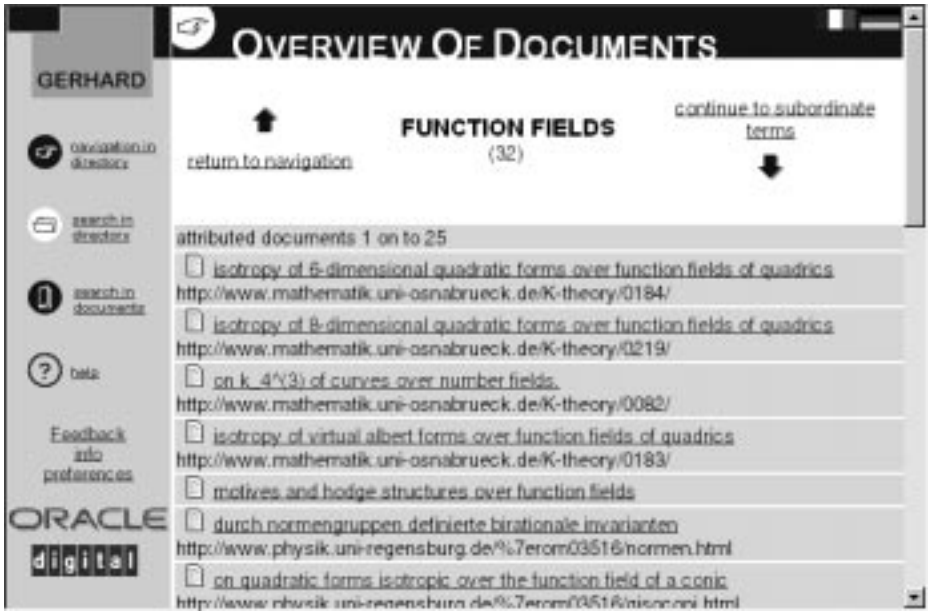
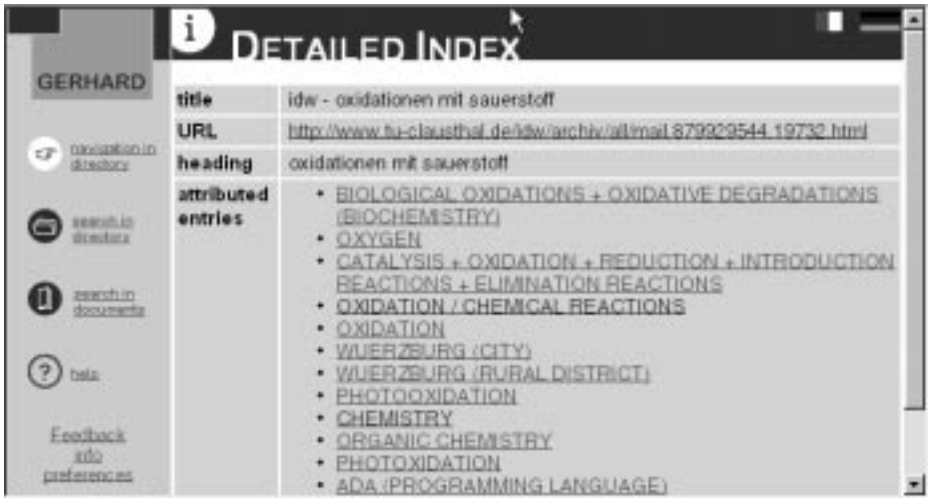


Fig. 6. Detailed view of a document



In the detailed view among other things all assigned categories are listed. Clicking one of those entries directly jumps back into navigation at the position of the clicked category. This enables a similarity-search on concepts (categories): If the user finds an interesting document, she checks the assigned categories. Often categories describing her interests exist that she was not aware of, hence making documents available, being described by former unknown categories![28]

Finally, it should be mentioned that the administration and configuration of GERHARD is possible via a Web-interface, and that all pages are generated on the fly through PL/SQL in the database system itself.

8 Conclusion

GERHARD has been developed in only 24 man-months and is operating on a DEC Alpha Server 1000 (192 MByte RAM, 18 GByte HDD). Since 4/1/98 the service is public and can be accessed via http://www.gerhard.de. All objectives have been met, some results are so promising that a further development of GERHARD seems to be very suggestive.

> [GERHARD] is the most comprehensive and most deeply exploited catalogue that can be due to its automatic creation very up-to-date. Bigger, substantially more current, more systematically exploited, and with monumental less effort built than for example Yahoo! ...[GERHARD] shows high potential and promising approaches, ...[but has to be] improved and further developed in additional research. The development of automatic classification methods in realistic applications in the Internet has just begun. [7]

At the moment there exists a cooperation with the Digital Library Project of Nordrhein-Westfalen, in which an interface to the classification service of GERHARD is used. To make GERHARD available for other search-engines in Europe[29], a Z39.50 gateway is being implemented at

---

This is because the category has an additional super-category with other documents assigned to it. (Here: "chemistry".)

[26] `ACCURACY` in `NODES_SOIF` (Fig. 3).

[27] The linguistic processing for French could not be achieved in the short development period, but could be added.

[28] It is planned to select multiple categories that the documents should share.

[29] E. g. Europa-Gate, Nordic-Web-Index, MeDoc, and the DBV- OSI/Z39.50-Projekt.

the moment. Also work to replace the Harvest gatherer by Combine[30] is ongoing.

GERHARD is a proof of concept and is suited very well for further research in many fields, e. g.:

- improvement of the classification algorithms
- evaluation of several classification methods
- evaluating the behavior of users when looking for new information
- profile services, based on interests and concepts
- automatic extension of the relations between concepts
- improvement of navigation using VRML
- improvement similarity search using more than one shared category
- service for automatic generation of metadata for authors
- integration of subject-specific classification-systems like CRCS

Talks with developers of the EU-project Desire have shown that many synergies can be obtained, this should be pursued actively.

References

[1] Anders Ardö, Franck Falcoz, Traugott Koch, Morten Nielsen, and Mogens Sandfær. Improving resource discovery and retrieval on the Internet: The Nordic WAIS/World Wide Web project – summary report. *Nordinfo-Nytt*, (4):13–28, 1994. http://www.nordinfo.helsinki.fi/nordnytt/nnytt4_94/sandfaer.htm.

[2] Lois Mai Chan, John Phillip Comaromi, and Mohinder Partap Satija. *Dewey Decimal Classification: a practical guide*. Forest Press, Albany, New York, 1994. http://www.oclc.org/oclc/man/9353pg/9353toc.htm.

[3] Alan Dawson. BUBL bursts out of bath. *The Serials Librarian*, 31(4):15–22, 1997. http://bubl.ac.uk/journals/lis/oz/serlib/v31n0497/dawson.htm.

[4] Deutscher Normenausschuß (DNA), editor. *DK Dezimalklassifikation: Deutsche Kurzausgabe*. Beuth-Vertrieb GmbH, Berlin, Köln, Berlin, 4 edition, 1973.

[5] Darren R. Hardy, Michael F. Schwartz, and Duane Wessels. Harvest user's manual. Documentation, U. Colorado, 31. 1. 1996 1996. http://www.tardis.ed.ac.uk/harvest/docs/old-manual/.

[6] Tracey Hooper. Desire II – development of a european service for information on research and education II. Technical Report RE 4004 (RE) - D1.1, Institute for Learning and Research Technology, University of Bristol, 1998. http://www.desire.org/html/research/deliverables/D1_1/D1_1.html.

[7] Traugott Koch. Nutzung von Klassifikationssystemen zur verbesserten Beschreibung, Organisation und Suche von Internetressourcen. *Buch und Bibliothek*, (5):326–335, 1998.

[8] Klaus Loth. Wissensorganisation durch ein neues Notationssystem – eine konstruktive Kritik der UDK. *ABI-Technik*, 16(1):17–28, 1996.

[30] A by NetLab, Lund for the EU-project Desire developed gatherer.

[9] Gerhard Möller. Relationen in der Universellen Dezimalklassifikation, June 1997 1997. http://www.gerhard.de/info/Dokumente/UDKZ/relationen_udkz.pdf.

[10] Keith Shafer. Scorpion helps catalog the web. *Bulletin of the American Society for Information Science*, 24(1):28–29, 1997. http://orc.rsch.oclc.org:6109/b-asis.html.

[11] Narayanan Shivakumar and Hector Garcia-Molina. Finding near-replicas of documents on the web. In *Proceedings of Workshop on Web Databases (WebDB'98)*, Valencia, Spain, 1998. http://www-db.stanford.edu/ shiva/Pubs/web.ps.

[12] Roger Thompson, Keith Shafer, and Diane Vizine-Goetz. Evaluating Dewey concepts as a knowledge base for automatic subject assignment. In *2nd ACM international conference on Digital libraries*, pages 37–46, 1997. http://purl.oclc.org/scorpion/eval_dc.html.

[13] Hans-Joachim Wätjen. GERHARD – Automatisches Sammeln, Klassifizieren und Indexieren von wissenschaftlich relevanten Informationsressourcen im deutschen World Wide Web. *B.I.T. online: Zeitschrift für Bibliothek, Information und Technologie*, (4):279–290, 1998. http://www.gerhard.de/info/index_de.shtml.

[14] Hans-Joachim Wätjen, Bernd Diekmann, Gerhard Möller, and Kai-Uwe Carstensen. Bericht zum DFG-Projekt: GERHARD. Technical report, BIS, 16. 6. 1998 1998. http://www.gerhard.de/info/index_de.shtml.