1-1-2005

Restructuring help systems using formal concept analysis

Peter W. Eklund

University of Wollongong, peklund@uow.edu.au

Bastian Wormuth

Darmstadt University of Technology

Follow this and additional works at: <https://ro.uow.edu.au/commpapers>

Part of the [Business Commons](#), and the [Social and Behavioral Sciences Commons](#)

Recommended Citation

Eklund, Peter W. and Wormuth, Bastian: Restructuring help systems using formal concept analysis 2005, 129-144.

<https://ro.uow.edu.au/commpapers/2185>

Restructuring help systems using formal concept analysis

Abstract

This paper extends standard help system technology to demonstrate the suitability of Formal Concept Analysis in displaying, searching and navigating help content. The paper introduces a method for building suitable scales directly from the help system index by computing a keyword extension set. The keyword extension technique is generalisable in any document collection where a hand-crafted index of terms is available.

Keywords

Restructuring, Help, Systems, Using, Formal, Concept, Analysis

Disciplines

Business | Social and Behavioral Sciences

Publication Details

Eklund, P. W. & Wormuth, B. (2005). Restructuring help systems using formal concept analysis. International Formal Concept Analysis Conference (pp. 129-144). Berlin, Germany: Springer Verlag.

Restructuring Help Systems using Formal Concept Analysis

Peter Eklund¹ and Bastian Wormuth²

¹ School of Information Technology and Computer Science
The University of Wollongong
Northfields Avenue, Wollongong, NSW 2522, Australia
peklund@uow.edu.au

² Darmstadt University of Technology, Department of Mathematics,
Schloßgartenstr. 7, 64289 Darmstadt, Germany;
bastian@wormuth.info

Abstract. This paper extends standard help system technology to demonstrate the suitability of Formal Concept Analysis in displaying, searching and navigating help content. The paper introduces a method for building suitable scales directly from the help system index by computing a keyword extension set. The keyword extension technique is generalisable in any document collection where a hand-crafted index of terms is available.

1 Introduction

Wille [1] writes that methods for knowledge processing presuppose an understanding of what knowledge is, we therefore begin by examining the nature of “knowledge” in the help domain.

Firstly, help systems are multipurpose. To understand their purpose, we examine help in terms of the dimensions of Conceptual Knowledge Processing (CKP) [1]. At the simplest level a help system provides instruction in the style of an on-line tutorial. In this case there is an instructional narrative where tasks are presented in a predefined order. In the same vein, help systems are also used to present more advanced “how to” information to users who have mastered the basics but are following a command sequence for the first time. This corresponds to **exploring** in CKP [1]. More advanced users have completed most command sequences in the software. For these users, the help system is used to **search** for something they can “more or less specify but not localize” [1]. The help materializes knowledge that is either beyond the present recollection or too unimportant to commit to memory.

Identifying the taxonomy of commands – or functionality – is a major problem for many software systems, particularly those based on Windows. The drill down made possible by the combination of dynamic menus, property settings, tabs and dialog boxes often leads to feature discovery through trial and error. Windows-based help systems have a particular style that supports pathfinding of this sort to reduce the “uncertainty” (as it is called in information science [2]) resulting from so many branches in a hierarchy.

Fig. 1. Naive analysis of the help system for the MAIL-SLEUTH program using SIENA. Many of the pages towards the top of the lattice involve initial set-up, configuration or initialization tasks, done once on installation. Help pages that are insinuated across the MAIL-SLEUTH help appear lower in the lattice, the idea of “structures” (presented in the page Structure.html) is (for instance) important because it is referred to from many other pages. The poor structure and presentation are indicative of the unstructured *ad hoc* content of the help system.

Having identified the “knowledge” of help systems w.r.t. elementary methods used in CKP, the hypothesis we test is that the application of FCA to the **analysis** of help system content can be **restructured** to **improve** its presentation for the purpose of supporting inter-subjective human argumentation (**decision** support based on mixed initiative – a dominant theme in the modern practice of HCI [3]). This we believe will enhance the user’s capacity to **investigate** the help content.

Two outcomes result from our study. The first is the recasting of the help system for the MAIL-SLEUTH program as a Conceptual Information System. We demonstrate how the creation of suitable scales can clarify the presentation of help content. The second is a method to extend search terms from the help system index. This method involves seeding the scale with search terms which are then expanded using the hand made index provided by the CHM file format. We describe this technique and evaluate its suitability to help systems.

2 Motivation

Fig. 1 shows the lattice that results from a context made of a set of objects as help pages (HTML documents) with attributes as URLs (`</href= />` tags to other help pages)³. The resulting concept lattice is rather poor. This line diagram is included because we can learn from it. First, we learn that scaling is required to isolate the help contents w.r.t. the purpose and functionality of the software being described. Second, the disarray in this concept lattice reinforces that the help system was not developed in a systematic way. The presentation of the content is *ad hoc* and is therefore likely to require restructuring.

Most help systems for Windows applications are constructed from compiled HTML files (so called CHM files) using a program called Microsoft HTML help. Compiled HTML has a number of advantages. Firstly, the compiled HTML is usually smaller than the original source. Second, compiled HTML is indexed and is therefore searchable. Third, the format for the presentation of the compiled HTML is standard in the Windows operating environment. Finally, the content of the help pages is not easily copied once compiled.

The search functionality for compiled HTML is a keyword search. Search terms can be entered into the search area, index entries relate to help pages and lack relations to other index terms; no context is given on a search term even when the term appears in multiple contexts. Although it is easy to find explanations when the correct search term has been entered, it is much harder to find the right explanation when the searcher is unsure of the appropriate search term to use. The browsing metaphor for compiled HTML is also rather limited. A tree widget is presented that requires browsing hierarchically by chapter, section and page. In short, the HTML help environment lacks the ability to semantically relate terms with pages and the search context.

What attracts our attention to compiled HTML as a content platform is its potential to be improved through the application of the concept lattice for browsing and visualizing help content. In the next generation of Microsoft operating systems, the use of MAML (Microsoft Assistance Markup Language) claims to counter this difficulty⁴. MAML will enable help to be tagged with dynamic properties that allow parts of the help system to be activated depending on the content. However, this paper addresses how FCA can value-add the current generation of help systems and compiled HTML.

Our paper follows the application thread presented by other work, specifically CEM [4] and RENTAL-FCA [5], where the use of the concept lattice has been shown to be well-suited to document browsing in the domains of email and News classifieds. More recent work by Eklund et al.[6] supports these claims through a usability trial for the MAIL-SLEUTH program. A more comprehensive usability study of MAIL-SLEUTH's Web-mail variant, MAILSTRAINER, was also recently completed [7]. In the MAILSTRAINER study, 16 users completed 7 search and

³ A Java program called EXLINK was written for this purpose by Shaun Domingo to automatically build the context.

⁴ <http://www.winwriters.com/articles/longhorn>

Fig. 2. The *Introduction* scale provides the starting point for the MAIL-SLEUTH help. Fig. 3 is the scale that appears when the user double clicks on the centre left vertex *Line Diagrams*. Nested-line diagrams are therefore implemented as a form of progressive drill-down rather than in the usual all-at-once nested line diagram.

document management tasks and completed a psychometric survey based on 10 questions on their experience and impressions with the software. A detailed presentation of MAILSTRAINER will be presented in a subsequent paper as a synthesis of Domingo [7].

Further, Carpineto and Romano – pioneers in the application of FCA to information retrieval [8,9] – have engineered the CREDO search engine⁵. This outcome reinforces the appeal of a lattice-based approach to search. The outcome from our paper is narrower in scope, it demonstrates the suitability of FCA to the search and navigation of help system content and characterizes the mathematical structures that support an understanding of this process.

3 Examination of Help Content using FCA

In the following section we presume a basic knowledge of Formal Concept Analysis. For an introduction to FCA see [10], further mathematical foundations are given in [11]. Practical FCA algorithms and methods are presented in [8].

We will discuss two ways to create a Conceptual Information System (CIS) based around the functionality of help systems. The first is to take an existing help system based on the index file, the content file and the explanation files (a

⁵ <http://credo.fub.it>

Fig. 3. Restructuring of the MAIL-SLEUTH help involved the creation of several static scales: these included a scale for searching, drawing line diagrams, Getting Started, Configuration, Folder Management and Global Settings.

CHM file) and restructure it in Conceptual Knowledge Processing terms. The second is the creation of a help system from the ground up regarding principles of CKP. For both approaches, FCA is used as an analysis technique and the known advantages of line diagrams representing concept lattices are reinforced. The advantages of the concept lattice as a visual metaphor of the information help space include:

- the line diagram simultaneously displays the results of one or more search terms;

Search of the index in HTMLhelp CHM files is conjunctive keyword search, all search terms must be contained in the document if a match is to be found. This encourages one (or few) search terms initially with iterative search proceeding from a single keyword and then introducing additional keywords one at a time with the objective of reducing the size of the result set. Iterative search in the standard help system paradigm therefore proceeds from few-to-many keywords. In the CIS search paradigm, this is not normally the case, search encourages as many search terms as can be thought of in order to produce as rich a lattice structure as possible. This style of search isolates search terms of little value quickly, reducing the dimensionality of the search. Therefore iterative search proceeds in a many-to-few keyword direction initially. The **search** act in CIS, just described, includes **exploration**, **identification** and **investigation** of the information landscape [1].

- *the diagrams display the search results embedded in their conceptual context;*

The result from search in a CIS will display the context of the pages from the search terms. In the case of DOCCO and CEM the documents retrieved are revealed along with their position in the underlying file system. This provides context for the search.

- *the diagrams at one time can offer global, general and detailed views;*

The concept lattice that emerges from search in a CIS shows either the complete boolean lattice of search terms used or alternatively the reduced line diagram showing all points on the concept lattice instantiated by the keyword used in the search. This reveals the distribution of matching documents over the entire document collection. In so doing, it provides both detailed and general views of the collection according to the dimensions of the search.

- *predefined diagrams can guide you through a domain.*

The idea of a library of predefined contextual scales is not new and has been used in the ZIT library system [12] and elsewhere [5]. The creation and re-use of conceptual scales are a fundamental basis for the TOSCANA-workflow. General CIS tools, such as CEM [4] and TOSCANAJ⁶, allow the user to create their own individual search scales to provide reusable search contexts. In MAIL-SLEUTH [6], several useful predefined search scales are pre-loaded with the program and further scales may be added by the user. DOCCO⁷ and CREDO⁸ do not provide any scaling tools but this is consistent with the general nature of the collections they index.

3.1 Reprocessing of CHM-based Help Systems

The CHM files can be decompiled using free decompilers like KEYTOOLS from KEYWORKS Software⁹. The result is a set of HTML files, a file containing the index and a file containing the contents of the help system. The HTML files are the explanation texts that appear in the right hand output window of a standard help system window, Fig. 4 shows such an interface. The system provides three options to find the explanation text that answers a request, the search function, the tree structured content of the help system and the index. The explanation text itself is structured in three elements: the title, the text body and a paragraph that lists related topics: "see also" or "related topics". Further, there may be hyperlinks in the body of the text that connect certain words or phrases in the explanation text to other relevant pages. All hyperlinks from the "see also" section can also address other CHM help files.

⁶ <http://www.toscanaj.sf.net>

⁷ <http://www.tockit.sf.net>

⁸ <http://credo.fub.it>

⁹ <http://www.keyworks.net>

Fig. 4. The standard user interface for a help system in MICROSOFT WINDOWS systems. The user has three means to search the help content provided by tabs on the left hand of the window: the index, the contents and the search function. All these functions produce a list of potential titles, phrases or keywords to the user, selecting one item from this list displays its corresponding explanation file in the right hand frame of the window. Further navigation functionalities similar to web browsing are also provided.

The application of Formal Concept Analysis requires the assignment of objects and attributes to the de-compiled elements. The subjects the user is looking for are the explanation texts and the constraints for the search are the keywords the user enters. So the explanation files become the objects and the keywords entered become the attributes of the Conceptual Information System. With G as the set of explanation files, M being the set of entered keywords and the incidence relation gIm , if an explanation files contains a keyword m , we produce the Formal Context (G, M, I) that is the basis for the concept lattice displayed in the graphical search interface.

Fig. 5 shows a simple search result using the FCA based retrieval tool DOCCO¹⁰ applied on the help content of MAIL-SLEUTH¹¹.

¹⁰ <http://www.tockit.sf.net>

¹¹ <http://www.mail-sleuth.com>

Fig. 5. Screenshot of Docco: the entered search terms are "drill down", "virtual folder" and "diagram". The relevant search result is actually the explanation file attached to the bottom node highlighted. The selected file(s) can directly be opened with the button "Open with default application...".

This result is produced against on the raw set of HTML files from the de-compilation. DOCCO creates an index of the words contained in the explanation files and displays a diagram of a concept lattice where the search terms are the attributes and the explanation files containing any of the search terms as objects. The inherent structures of the explanation files given by the links to related topics are not considered in this approach.

The opposite approach, where the search terms are the objects and the explanation files the attributes, yields a diagram that better supports users without familiarity with FCA. This structure provides a kind of ranking with the best results, the most specific explanation files, in the top section of the diagram and more general matches below. This hierarchy is also given in the approach pursued in this paper: the best results are expected in the extent of the bottom concept of the lattice.

Fig. 6. Diagram representing the concept lattice of a search for the explanation for "Changing the name of a folder": the user enters "change" and "name" as search terms.

3.2 Search Improvements and Automatic Search Extension

A user searching a large data set for a special question would like to achieve best results with minimum effort. The process of searching has been effective and successful when the set of relevant search results resp. explanation files is small. So the objective is to minimize the number of objects in the extent of the bottom concept of the line diagram. The more search terms provided, the smaller the extent and the more specific the concept. The problem here is that users do not generally enter many search terms, so some alternative support is required to improve the search functionality: through query expansion. Fig. 6 shows the result of a search for the simple question "Changing the name of a folder" in the standard help file of MICROSOFT WINDOWS. The user entered "change" and "name" as search terms, the size of extent of the bottom concept is 9, so 9 different explanation files are offered as result.

Most help systems do not support query expansion to find the relevant explanation file, so the user often has few clues on what to search for. Query expansion by the automatic extension of the search terms is one solution. The use case requires one or two keywords to act as a seed for generating extension terms. In Windows help, the search terms "change", "name" might be expanded to "change", "name" "file" and "folder" for instance.

Additional search terms can be derived from the index of the given help system. In the following we propose a simple method which leverages the human

effort formalized in the hand-made index contained within the CHM file. By understanding this hand-made index as a formal context $\mathbb{K}_{CHM}$, as described in Section 3.1, we can use it to derive additional search terms using the prime operator.

Let $\mathbb{K}_{index}$ be the formal context used by DOCCO, with $G_{index} = G_{CHM}$ being the set of explanation files of the help system, M_{index} being the automatically indexed words and gIm , if $m \in M_{index}$ appears in $g \in G_{index}$. As the index used by DOCCO is not manually created, but collects all words (except those in a stop words list) that appear in the explanation files, it is (in practice) much larger than the hand-made index. By larger we mean that $M_{CHM} \subseteq M_{index}$, the object sets are identical. It is possible that the author of the help system has added entries in the hand-made CHM index that do not occur in the explanation file, but this is a rather pathological case. Furthermore there are differences in the incidence relations I_{CHM} and I_{index} : not every incidence relation in between a term $m \in M_{index}$ and an explanation file $g \in G_{index}$ is realised in the context $\mathbb{K}_{CHM}$ and symmetrically there may be incidence relations in $\mathbb{K}_{CHM}$ that are not realised in $\mathbb{K}_{index}$.

The advantage of the hand-made index is the fact that the words or phrases are assigned by a human to a certain explanation file and therefore the explanation file being "about" the phrase or keyword assigned is assured.

Let S be the set of seed search terms entered by the user, S_{ext} be the extended set of search terms and G_{CHM} the set of explanation files. S_{ext} is derived from S and G_{CHM} by the following:

1. Derive the set $S' \in G_{CHM}$ of explanation files where all instances of S occur in every instance of S' .
2. $S_{ext} = S'', S'' \in M_{CHM}$ is the set of words that all instances of S' have in common.

In the language of FCA this is the derivation of the attribute concept of the seed search terms in the context $\mathbb{K}_{CHM}$:

$$S_{ext} := \{S \cap M_{CHM}\}''.$$

If the CHM index on the explanation files does not remove stop words S_{ext} will contain many redundant words. This stop word filter should be applied *after* the derivation of S_{ext} , because the removal of stop words can result in substantial information loss, e.g. the words "on" and "off" in "log on" and "log off".

The resulting set S_{ext} can be offered to the user to specialize.

Applying this method on the search displayed in Fig. 6 with $S = \{\text{change, name}\}$ returns,

$$S_{ext} = \{\text{Change, file, folder, changing, forbidden, characters, names, renaming, naming, overview}\}.$$

Adding the search terms "file" and "folder" to the search yields the diagram given in Fig. 7. Due to the more specific search this line diagram has just three

Fig. 7. A line diagram representing the concept lattice of a search for the explanation for "Changing the name of a folder" as in Fig. 6. Additional to the initially entered terms "change" and "name", the user also entered "file" and "folder": derived by automatic search term extension. The most specific results are the most relevant and represented in the three explanation files objects in the extent of the bottom concept of the line diagram.

explanation files in the extent of the bottom concept and among those the most relevant file to the intended query, "Change the name of a file or folder" is included.

The complexity of this operation is $O(|G||M|)$, $|G|$ being the number of objects and $|M|$ the number of attributes. This operation computes in an acceptable practical time frame for instantaneous keyword expansion for contexts based on indexed decompiled CHM Help files.

The method to derive an extended set of search terms relates to [13] and [14]. The following presents a short introduction to multicontexts:

Multicontexts can be understood as a network of formal contexts describing incidence relations of a domain, each context representing a different view of the "situation". Different contexts may have different object and attribute sets, they may also share elements in common, although they do not have to. It is also possible that the object set of a context shares elements with the attribute

set of another context, which causes interesting relations in between the lattices of two different contexts of a multicontext.

Following the definition of multicontexts, quoted from [13]:

Definition 1. A multicontext of signature $\sigma : P \rightarrow I^2$, where I and P are nonempty sets, is defined as a pair (S_I, R_P) consisting of a family $S_I := (S_i)_{i \in I}$ of sets and a family $R_P := (R_p)_{p \in P}$ of binary relations with $R_p \subseteq S_i \times S_j$ if $\sigma p = (i, j)$. A multicontext (S_I, R_P) of signature $\sigma : P \rightarrow I^2$ can be understood as a network of formal contexts $\mathbb{K}_p := (S_i, S_j, R_p)$ with $\sigma p = (i, j)$.

Multicontexts provide so called coherence mappings between two contexts:

Definition 2. Let (S_I, R_P) be a multicontext, let $\mathbb{K}_p := (S_i, S_j, R_p)$ and $\mathbb{K}_q := (S_k, S_l, R_q)$ be formal contexts of (S_I, R_P) , and let (A, B) be a formal concept of $\mathbb{K}_p$. Then there are four coherence mappings from $\mathfrak{B}(\mathbb{K}_p)$ to $\mathfrak{B}(\mathbb{K}_q)$ defined by:

$$\begin{aligned}\lambda_{pq}(A, B) &:= ((A \cap S_k)^{qq}, (A \cap S_k)^q) \\ \varrho_{pq}(A, B) &:= ((B \cap S_l)^q, (B \cap S_l)^{qq}) \\ \varphi_{pq}(A, B) &:= ((B \cap S_k)^{qq}, (B \cap S_k)^q) \\ \psi_{pq}(A, B) &:= ((A \cap S_l)^q, (A \cap S_l)^{qq})\end{aligned}$$

The concept lattices of the formal contexts of (S_I, R_P) together with all coherence mappings form a coherence network of concept lattices which shall be denoted by $\mathfrak{R}(S_I, R_P)$. See also Fig. 8.

Fig. 8. Coherence mappings

Coherence mappings, shown in Fig. 8, allow the user to select a concept as a formal context of a multicontext and to map this concept to another context – simply put, the user has the option to select his objects of interest in a certain view and to compare this selection to the corresponding results in a different view.

The coherence mappings establish a means to examine the relations between two contexts – exactly the way we derived the extended set S_{ext} of search terms.

In our example we took the concept of interest and mapped it to the corresponding one in the lattice of $\mathbb{K}_{CHM}$, then we transferred the intent of the mappings image back to DOCCO and created the sub lattice of $\mathbb{K}_{index}$ with the new set of search terms to specify the search.

Let (A, B) be the concept of $\mathbb{K}_{index}$ that seems interesting from the users point of view. Then the corresponding concept of $\mathbb{K}_{CHM}$ is derived using the mapping φ_{pq} with $p = index$ and $q = CHM$ that maps from $\mathbb{K}_{index}$ to $\mathbb{K}_{CHM}$:

$$\varphi_{pq}(A, B) = ((B \cap M_{CHM})^{qq}, (B \cap M_{CHM})^q)$$

The intent of $\varphi_{pq}(A, B)$ is the extension set S_{ext} that we used to specify the search, and because of $M_{CHM} \subseteq M_{index}$, the intent of the derived concept is still part of M_{index} and produces a promising extension of search terms.

3.3 Processing the Structure of the Help Content

Any CHM based help system is a structured document with three key elements. The first is the HTML of the actual explanation file. Sub-parts of the layout as described in Section 3.1 can easily be found in the HTML source code: the title and the body tags of the explanation file for example. The identification of the "related topic" paragraph is more difficult, because HTML provides various means to tag this paragraph. The second key element is the table of contents of the help system. Although a tree structure, the table of contents groups explanation files by their content. The third key element is given by the links from words in the body of an explanation file to other explanation files. The fact that external explanation files of other CHM files can be incorporated allows the inclusion of related files.

These elements can be used to derive sets of explanation files that are of higher importance to each explanation file: for every explanation file we define a set of related explanation files, these are those that appear in the same sub-directory of the table of contents, those listed in the "related topics" paragraph and those that are link targets from the body of the explanation file. This set, we call the *scope set* of an explanation file, because it forms a scope for every explanation file and can be used in Formal Concept Analysis to create predefined conceptual scales for the CIS. Such derived sets will not normally be disjoint, except in the case where pages indexed by the table of contents are disjoint.

The elements: title, body and "related topics" can also be used to apply constraints on the search. These provide three "search modalities": restricting search to titles, bodies, titles and bodies, and so on. Only title search can be achieved with current version of DOCCO. Fig. 9 shows the result of a title element search using the same terms "change" and "name" as in Figure 6 and 7, but restricted to the titles. The only object in the extent of the bottom concept is the relevant explanation file.

Further possibilities to use the resolved structure of the HTML help files to improve the relevant results will be proposed in Section 4.

Fig. 9. A line diagram representing the concept lattice of a search for the explanation for "how to change name of a folder": the user entered "change" and "name" as search terms and the search is restricted to the title element tags of the explanation files.

4 Further Research

The structure provided by the HTML files and the human effort coded in the content and index files that form part of the CHM files have not been used before to engineer a Conceptual Information System. In addition, the following paragraph proposes a means to capture the inherited structure of the explanation and the content files using multicontexts.

In addition to multicontexts, triadic contexts as introduced by Wille and Zickwolff in [15] need to be addressed also. The core difference between triadic contexts and formal contexts of a multicontext is that every incidence relation in a triadic context is valid in a certain modality. This suits our requirements, with different "search modalities" conforming to help system key elements. Triadic contexts and multicontexts are related, because every triadic context gives rise to a multicontext [13]. The problem of characterising triadic contexts to improve the exploration of help systems is that the understanding of line diagrams representing triadic contexts requires considerable experience in Formal Concept Analysis, so we focus on multicontexts.

The idea is to create a multicontext consisting of contexts based on the indexed words of the titles of the explanation texts, e.g., the bodies and the "related topics" paragraph in order to have a mathematical foundation for the search modalities as described in Section 3.3.

Additional contexts can be derived from each chapter of the table of contents. An implementation of the multicontexts would allow the user to search the data in different modalities, e.g. to restrict the search to titles and to incorporate the bodies of the explanation files after selecting a certain concept. The introduction of other content tags can enhance the search additionally by providing finer segmentation on the document. Imagine a user having problems with his network connection. A search on the standard MS Windows help for "network connection" gives about 100 hits. The Windows help has a chapter "Networking" and another "Mobile Users", a search using the "mobile" or "standard" context would focus on the related explanation files. If the user has found an explanation file that helps, but does not solve his problem, he can then compare search results to the corresponding results from another search modality.

5 Conclusion

MAIL-SLEUTH is a program that itself uses the concept lattice to dynamically organize the contents of an email collection into thematically coherent views based on the application of Formal Concept Analysis. It therefore seems fitting to experiment with the very same idea for the MAIL-SLEUTH help system. This desire for design coherence was the starting point for our study.

Our research reveals that help systems (in general) are well-suited to a content-management approach based on the concept lattice for navigation. There are several reasons for this, the first is that help systems are relatively small so computing extension keywords is realistic, the second is that help systems are a semi-structured document source with strongly inter-related content and finally help systems come equipped with a handy precomputed, hand-crafted index.

In our paper Wille's **restructuring** idea is applied by examining the content of the MAIL-SLEUTH help. The results were the creation of six predefined scales that thematically covered all functional aspects of the software. The new help contents were then re-written w.r.t. the scales with self-evident improvements to the presentation and structure of the help system. Hand crafting scales in this way is ideal but there is also scope to automatically generate scales by using seed search terms and computing the extension set. We show how this can be achieved and present examples that illustrate keyword extension.

References

1. Wille, R.: Conceptual landscapes of knowledge: A pragmatic paradigm for knowledge processing. In Gaul, W., Locarek-Junge, H., eds.: *Classification in the Information Age*, Springer (1999) 344–356
2. Colomb, R.: *Information Spaces: the Architecture of Cyberspace*. Springer (2002)
3. Horvotz, E., Kadie, C., Paek, T., Hovel, D.: Models of attention in computing and communication: From principles to applications. *CACM* **46** (2003) 52–59
4. Cole, R., Eklund, P., Stumme, G.: Document retrieval for email search and discovery using formal concept analysis. *Journal of Experimental and Theoretical Artificial Intelligence* **17** (2003) 257–280

5. Cole, R., Eklund, P.: Browsing semi-structured web texts using formal concept analysis. In: Proceedings 9th International Conference on Conceptual Structures. LNAI (2001) 319–332
6. Eklund, P., Ducrou, J., Brawn, P.: Concept lattices for information visualization: Can novices read line diagrams. In Eklund, P., ed.: Proceedings of the 2nd International Conference on Formal Concept Analysis - ICFCA'04, Springer (2004)
7. Domingo, S.: Engineering and Interactive Searching and Browsing Environment for Web-Based Email. PhD thesis, School of IT and Computer Science (2004)
8. Carpineto, C., Romano, G.: Concept Data Processing: Theory and Practice. Wiley (2004)
9. Carpineto, C., Romano, G.: Order-theoretical ranking. *Journal of the American Society for Information Sciences (JASIS)* 7 (2000) 587–601
10. Wille, R.: Introduction to Formal Concept Analysis. In Negrini, G., ed.: *Modelli e modellazione, Models and Modelling*, Roma, Consiglio Nazionale delle Ricerche, Istituto di Studi sulle Ricerche e Documentazione Scientifica (1997)
11. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin (1999)
12. Rock, T., Wille, R.: Ein TOSCANA—Erkundungssystem zur Literatursuche. In Stumme, G., Wille, R., eds.: *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*, Berlin-Heidelberg, Springer (2000) 239–253
13. Dörflin, S.K., Wille, R.: *Coherence Networks of Concept Lattices: The Basic Theorem* (2004)
14. Wille, R.: Conceptual Structures of Multicontexts. In Eklund, P.W., Ellis, G., Mann, G., eds.: *Conceptual Structures: Knowledge Representation as Interlingua*. Number 1115 in LNAI, Heidelberg - Berlin - New York, Springer (1996) 23–29
15. Wille, R., Zickwolff, M.: Grundlagen einer Triadischen Begriffsanalyse. In Stumme, G., Wille, R., eds.: *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*, Heidelberg - Berlin - New York, Springer (2000) 125–150
16. Dieberger, A., Dourish, P., Höök, K., Resnick, P., Wexelblat, A.: Social navigation: Techniques for building more usable systems. *ACM Transactions on Human-Computer Interaction* 7 (2004) 26–58