

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.

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)¹. 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.

¹ A Java program called EXLINK was written for this purpose by Shaun Domingo to automatically build the context.