Optimizing hierarchical menus: a usage-based approach

Hollink, V.

Citation for published version (APA):
Chapter 1

Introduction

1.1 Background

Nowadays many web sites consist of hundreds, thousands or even hundreds of thousands of pages. Together these pages contain a wealth of information that can be used to answer many different questions. However, at the same time the large number of pages makes it difficult for users to find answers to their information needs, even when they have found a web site that does contain the answers. To assist users in their search, modern web sites offer a range of navigation means, such as in-text links and site search engines. In this thesis we focus on one of the oldest and most frequently used navigation means: hierarchical menus. In particular, we investigate how hierarchical menus can automatically be optimized in such a way that navigation becomes as efficient as possible.

Hierarchical menus are navigation structures consisting of hierarchies of links. Each link has a label that describes the content that can be reached by following the link. Users read the descriptions of the available links to choose which links they will open. When a link is selected, the content of the new page and the subitems of the selected link are shown. Most menus contain content from one web site, but there are exceptions. For instance, web directories such as Yahoo (Yahoo! Inc., 2007) and Dmoz (Dmoz, 2007) can be seen as very large menus linking to content from many different sites.

A great asset of hierarchical menus is that they do not require users to express their information needs in the terminology of the web site. Users often have difficulties specifying their information needs as free text, which makes it impossible to make effective use of site search engines (Alpay et al., 2004). For these users menus are more appropriate because menus make the available options visible. In this way, menus allow users to recognize the relevant links instead of forcing them to recall keywords, which reduces memory load (Molich and Nielsen, 1990; Nielsen, 1994). For the same reason, menus form a good solution when users think of their problem in a different vocabulary than the one used on the web site.

Menus can support various aspects of navigation. Most hierarchical menus are aimed at users with specific information needs. These users visit the site with the goal
Chapter 1: Introduction

to find certain pieces of information. We call the set of pages that together provide the best answer to a user's information needs the user's target pages. The goal of menus aimed at this type of users is to allow the users fast and easy access to their target pages. Ezendam et al. (2005) introduced menus for users with less articulate information needs, who do not know exactly what information they are looking for. The menus guide users step by step through the information on the site showing them in which order they should read the available information. In the following, these menus will be referred to as problem-oriented menus.

The navigational function of menus distinguishes them from taxonomic hierarchies. Both structures comprise a hierarchy of categories. However, the relations in a taxonomy represent important features of the world, while the links in a menu are purely for navigation. This difference can lead to very different hierarchical structures. For example, in the Dmoz directory (Dmoz, 2007) the two internet browsers 'Firefox' and 'Charlotte' can be found in the same directory (Computers: Software: Internet: Clients: WWW: Browsers). This is conceptually logical, but not necessary efficient. For instance, suppose that many users visit Dmoz to find information on 'Firefox' and hardly anyone needs information on 'Charlotte'. Then navigation can be made more efficient by placing 'Firefox' at a higher position in the hierarchy than 'Charlotte', for example, directly under 'Software'.

Menus that are well-suited to the needs of the user population can effectively facilitate the users' navigation processes. However, when the hierarchical structure or the link descriptions of a menu do not match the users' needs, navigation becomes inefficient. The structure of the hierarchy determines how much navigation is needed to reach certain targets. Menus that are not well-structured (for instance, because popular targets are located deep in the hierarchy) require users to make many navigation steps. When the descriptions of the links are not correctly interpreted by the users, users cannot predict which links will lead to their target information. This results in navigation errors, which increases navigation time.

Developing high quality menus is a non-trivial task. Web designers often do not know exactly who their users will be and for what purpose they will visit the site. Moreover, it is not clear how characteristics of the user population should be translated to properties of a menu. On top of that, user populations as well as contents of web sites tend to change, so that over time even initially well-designed menus often become less efficient.

Human-computer interaction (HCI) research has yielded guidelines for developing hierarchical menus. For example, according to the ISO standard for interaction design (ISO, 2002) menu items that have great importance should be placed first in a menu. Several general HCI principles also apply to menus, including the ten design principles of Nielsen (1994). One of these principles recommends, for instance, that link descriptions are stated in terms that are familiar to the users. Such guidelines provide the minimal conditions to which a menu must comply, but they are often too generic to decide which of a number of alternative menus is best. Also, they do not account for specific properties of a user population, such as the time that users need to make a selection.

Instead of aiming to create optimal menus in advance, various authors have pro-
posed algorithms to improve hierarchical menus on the basis of usage data that is collected over time. These algorithms analyze the log files of a site and on the basis of this analysis predict which adaptations to a menu will make it more efficient. One of the earliest of these algorithms was developed by Witten et al. (1984). It optimizes the index of a digital phonebook using the access frequencies of the phonenumbers. Later algorithms for optimization of hierarchical link structures are, for example, presented by Fisher et al. (1990), Smyth and Cotter (2003) and Wang et al. (2006).

Automatic menu optimization is part of the broader research area of adaptive web sites. Adaptive web sites are web sites ‘that automatically improve their organisation and presentation by learning from user access patterns’ (Perkowitz and Etzioni, 1997, p. 16). Menu optimization has the same goal, but concentrates entirely on the sites’ menus. Systems that optimize menus by adding extra links to the menus are related to recommender systems. Recommender systems select a number of items that they believe to be interesting for a user. When these items are presented in the form of lists of links, these links can be seen as dynamically created menus.

We make a distinction between fully and semi-automatic menu optimization methods. Fully automatic methods adapt a menu structure without human interference. They do not require human effort which means that they can be applied frequently. Some methods even adapt menus to the personal needs of a user while the user is navigating the site. This type of adaptation is called personalization or customization (Perkowitz and Etzioni, 2000). A disadvantage of fully automatic methods is that they can damage a menu when they make mistakes. Therefore, in practice, fully automatic algorithms are only allowed to make small non-destructive changes, such as adding shortcut links. Semi-automatic menu adaptation methods compute useful adaptations, but do not implement them autonomously. A webmaster reviews the adaptations and implements the ones he (or she) finds acceptable. Because all adaptations are checked, semi-automatic methods can be used to make drastic changes to a menu or even completely restructure a menu.

Most adaptation methods receive the usage data they need from web servers, which collect these data in the form of log files. Log files contain data about the requests that users made to the server, such as the time of the request, the requested page and the user's IP address. In principle, log data suffice to determine which sequences of pages users have visited on the site, but there are several reasons why these sequences are not always 100% accurate (Cooley et al., 1999; Pierrakos et al., 2003). For instance, due to browser caching, repeated requests for the same page are sometimes not recorded. At the same time, automatic refreshes result in requests that are not initiated by a user. Moreover, web crawlers create navigation traces that are not always distinguishable from human traces. The effect of this is that most web log data is very noisy. Another problem with web log data is that it shows only which pages a user has visited, but not the reasons why he or she visited these pages. This complicates menu optimization, as it is hard to determine how well a menu supports a user's navigation process when we do not know the purpose of the navigation.

To overcome these problems, some menu adaptation methods require additional data besides standard log data. For example, some methods ask users to explicitly specify their goals or to indicate at the end of their search whether they have found
what they needed (e.g. Joachims et al., 1997). The main drawback of these methods is that users are often not willing to share this information (Perkowitz and Etzioni, 2000). Another type of additional data is information about the content of the site. For example, the WUM method (Spiliopoulou and Pohle, 2001) requires that the pages of the site are divided into categories that represent the various functions that pages can fulfill on a site. The algorithm presented in Wang et al. (2006) makes use of product categories. Creating these types of categories requires manual labor. Moreover, the categorization schemes are generally domain specific.

In this thesis we focus on domain-independent methods for improving hierarchical menus. The methods are very generally applicable, because they use only log data that is generally available and do not pose any restrictions on the contents of the pages. We identify a number of shortcomings of existing methods that optimize menu structures and present new strategies to overcome these problems. In addition, we address novel tasks, such as automatically creating problem-oriented menus and improving descriptions of links.

1.2 Research questions

The main focus of this thesis are methods to automatically improve hierarchical menus. The general research question that we will address is:

_How can we automatically or semi-automatically adapt hierarchical menus of web sites in such a way that the users of the sites can fulfill their information needs more efficiently?_

We refine this general question in four more specific questions. As explained before, the goal of most hierarchical menus is to allow users to reach their target information efficiently. The first question addresses the structure of this type of menus:

1. _How can we adapt the structure of hierarchical menus in such a way that they become maximally efficient for their user populations?_

The assumptions that are made about the way users navigate the site have a large influence on the outcome of optimization algorithms. Thus, before we can optimize the efficiency of a menu structure, we need to know which assumptions are valid. In other words, we need to understand how efficiency is determined by the characteristics of a menu and its users. We call a model that describes these relations a _navigation behavior model_. Finding the most accurate navigation behavior model for a user population is the topic of the second research question:

2. _Which characteristics of user populations must be included in a navigation behavior model to predict the efficiency of hierarchical menus?_

As discussed, problem-oriented menus help users to read pages in the right order. Until now these menus were created manually by experts. We ask ourselves how this process can be automated:
3. How can we automatically create problem-oriented menus?

When users follow incorrect paths through a menu, they have to make extra navigation steps to reach their goals. This increases navigation time and can lead to frustration with the site. Therefore, the last question that we will answer is:

4. How can we reduce the number of navigation mistakes in hierarchical menus?

1.3 Approach and outline

In Chapters 2 and 3 we research the optimization of a menu’s efficiency (research question 1). In Chapter 2 we focus on the optimization of one important aspect of efficiency: the number of navigation steps that users need to make to reach their target information. We identify a fundamental shortcoming of frequently used optimization methods that prevents them from minimizing the number of navigation steps. We explore several methods to overcome this problem. Simulation experiments and user studies are used to assess the effects of the presented methods.

One finding of the studies in Chapter 2 is that the presented methods are only adequate in very limited settings. For example, they are insufficient when besides the optimized menu structure also other navigation means are available. Moreover, other factors besides the number of navigation steps may play a role in navigation efficiency, such as the number of items in a menu. Therefore, in Chapter 3, we move to a more profound approach based on a complete model of user navigation in hierarchical menus. To answer research question 2 we perform a literature study and collect the factors that are explicitly or implicitly used to predict efficiency. The factors are placed in a framework that shows the relations between the various factors in a structured way. In addition, we provide a procedure to measure the influence of each of the factors on the efficiency of a given menu. In the second part of Chapter 3 we return to research question 1. We present a method to find a menu that optimizes the various factors. The outcomes of this method are evaluated by means of case studies.

In Chapter 4 we answer research question 3. We propose a method to determine the preferred reading order from log data. The output of this method is used to automatically construct problem-oriented navigation menus. The method is applied to the SeniorGezond site (SeniorGezond, 2007) which provides a problem-oriented menu created by experts (Ezendam et al., 2005). Evaluation is done by comparing the structure created by our method to the actual organization of the site. Additionally, the method is applied to a site that does not yet offer a problem-oriented navigation menu.

Chapters 2 to 4 all deal with structural properties of menu hierarchies. In Chapter 5 we will treat the optimization of link descriptions. In this chapter we address research question 4: reducing navigation errors. We hypothesize that users choose links on the basis of the descriptions of the available links. If this hypothesis holds, navigation errors can be attacked by improving descriptions that cause confusion. We present a method that analyzes log files and determines the locations in a menu where
users frequently make mistakes. For each location it determines the main type of the mistakes and provides a number of possible solutions. To evaluate the method, we ask experts to judge the value of the analyses. The effects of the improvements on the number of navigation errors are demonstrated in a user experiment.

In the last chapter we review our main conclusions and look back at the four research questions. In addition, we discuss limitations and advantages of our approach and explore directions for future research.