Optimizing hierarchical menus: a usage-based approach

Hollink, V.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
6.1 Main contributions

The aim of this thesis was to investigate how we can optimize hierarchical menus in such a way that navigation becomes as efficient as possible. Previous research in the area of link structure adaptation focused in majority on the optimization process itself. The question whether the criteria to which the optimization process is directed are consistent with the needs of the user population has received much less attention. In human-computer interaction research user needs are a major focus. Many HCI guidelines for developing menus involve analysis of the requirements of the user populations. However, these guidelines are usually not directly applicable for optimization as they are too general to decide which of a number of alternative menus is most efficient. In this work we took an approach that combines the best of both worlds. We created generic models that represented the needs of various types of users and provided methods to fit the model parameters for a specific site and user population. These models enabled us to explicitly represent user needs and at the same time were specific enough to direct optimization.

Until now menu optimization research was restricted to users with specific information needs and mainly covered the structures of menus. This thesis extended this work in several ways. We discussed both the optimization of menu structures and the optimization of the descriptions of menu items. In addition, we distinguished menus aimed at two types of users. The first type of menus supports users with specific information needs, who want to reach their target information as fast as possible. The second type are problem-oriented menus. These menus are aimed at users with less articulate questions, who gradually define their information needs while reading the information on the site.

Chapters 2 and 3 focused on the improvement of menu structures for users with specific navigation needs. We explored methods to determine the efficiency of a given menu structure and to find the most efficient structure for a menu. We showed that to accurately predict navigation time we need a model of the users’ navigation. We provided a method to find the best model for a user population and demonstrated how a navigation model can be incorporated in an algorithm that optimizes menu...
structures.

The creation of menus for users with unspecific information needs was the topic of Chapter 4. Inspiration for this work originated from the SeniorGezond site (SeniorGezond, 2007), which offers a menu that guides users step by step through the pages of the site. Evaluation of this site has shown that this problem-oriented menu facilitates navigation for users who do not know exactly which information they need (Ezendam et al., 2005). Problem-oriented menus have great potential for other sites that are visited by users with unspecific information needs, but manual creation of these menus requires considerable time and effort. To support this process, we presented an algorithm that automatically generates problem-oriented menus on the basis of usage information stored in log files.

Finally, in Chapter 5 we addressed the improvement of descriptions of menu items. Accurate descriptions prevent users from making navigation errors and thus lead to more efficient navigation. We presented a method to automatically detect descriptions that cause confusion. We distinguished various types of inaccurate link descriptions and for each type we provided a number of possible solutions that can help web masters to improve the descriptions.

Below, we go back to the research questions posed in the introduction. We discuss how the various parts of our research contribute to answering the questions. The last section discusses limitations of our work and explores avenues for future research.

6.2 Reflections on the research questions

6.2.1 How can we adapt the structure of hierarchical menus in such a way that they become maximally efficient for their user populations?

Many different algorithms have been developed that aim to make navigation more efficient by dynamically adding a fixed number of links to an existing link structure. In Chapter 2 we showed that the majority of these algorithms follow a greedy approach: they estimate the probability that the user is interested in each page and provide links to the most interesting pages. In each step the greedy approach maximizes the probability of leading a user to a target page directly, but we showed that it does not necessarily minimize the length of the users’ sessions. Simulation experiments and user studies confirmed that in practice the greedy approach indeed sometimes results in navigation times that are longer than necessary.

We presented a method to minimize the length of a session. This method uses principles from information theory to select the most informative sets of links. A maximally informative set of links divides the pages of a site into parts with equal probability of containing the user’s targets. When these links are added to the site, we can determine in a minimal number of navigation steps what the targets of the user are. Three variants of this method were tested in a series of simulation experiments and user studies. In all experiments the method proved more effective than the greedy approach: it significantly reduced the length of the users’ sessions. All three variants
are fully automatic and computationally efficient. Therefore, they are suitable for online use, for example, in a personalized recommender system.

The experiments proved that the page division method is effective when the dynamically added links form the only available navigation means and the number of added links is fixed. However, we found that these type of methods are insufficient in more realistic settings. Moreover, these methods do not provide a way to make a trade-off between the number of links in each menu layer and the number of menu layers.

In Chapter 3 we presented an approach to menu optimization that does not only consider the addition of links, but that takes the whole menu structure into account. This approach is based on an explicit model of the users' navigation. With this model we can simulate navigation through a menu and predict the average time users will need to reach their targets. A semi-automatic hill-climbing algorithm is used to gradually improve a menu. At each step the algorithm tries a number of possible adaptations that result in alternative menu structures. The navigation model is used to estimate the efficiency of each of the alternatives. The most promising alternatives are shown to a webmaster who decides which of the alternative structures is implemented. Using a semi-automatic method has two advantages. First, some of the modifications result in new menu items for which a label must be created by hand. Second, a semi-automatic method can make very fundamental changes to the menu, because all changes are checked by a human. However, the semi-automatic character of this method also means that it is not usable for online optimization.

The model-based approach allows us to assess the quality of the menu as a whole. Moreover, it enables us to clearly distinguish between the estimation of navigation time and the method that is used to find the optimal menu. The potential of the model-based method was shown in four case studies. In these studies the method proved able to select effective adaptations. According to the navigation model the adaptations considerably reduced navigation time. On top of that, the case studies showed that with this method one can create coherent categories for which a description can be found easily.

6.2.2 Which characteristics of user populations must be included in a navigation behavior model to predict the efficiency of hierarchical menus?

In Chapter 3 we showed that many systems that claim to optimize efficiency are based on incorrect assumptions about the navigation and goals of the users. The assumptions are often left implicit and almost never validated. This is a serious shortcoming as our experiments prove that the assumptions have a large influence on the optimization process and the resulting menu structures. Consequently, assumptions that are not consistent with the user population can easily lead to suboptimal efficiency.

To get a clear view on the assumptions that are made, a framework was provided that shows the assumptions in an organized way. In a literature study we collected the (implicit) models underlying various link structure optimization methods. This analysis revealed three main topics about which the methods made different assumptions:
properties of the link structures, the goals of the users and the navigation strategies of
the users. The framework further decomposes these topics in, in total, fifteen features
that represent detailed assumptions. For instance, one of the features describes the
relation between the number of menu items that users read and the reading time.

We developed a procedure to test the validity of a set of assumptions and to select
the best set of assumptions for a user population. We introduced the notion of a
navigation behavior model: an instantiation of the fifteen features of the framework.
A navigation behavior model suffices to predict navigation paths through a menu and,
from this, the average time users need to follow these paths. To find the best model for
a site all possible navigation models are constructed and used to predict navigation.
The predicted navigation is compared to the actual navigation of users as recorded in
the log files. The model that makes the most accurate predictions is selected.

In an experiment we applied the model selection method to four web sites. We
found that some feature values were inherently better than others: these values clearly
outperformed the other values at all four sites. We found that it is better to take into
account that users can have multiple targets than to make the simpler assumption
that each user searches for a single target page. Also, the relative frequency of targets
should not be ignored. Navigation time is a function of both the number of menu items
from which a user makes a choice and the number of times a user opens a menu item.
When selecting a model for a new site, these feature values can be used directly. For
other features the optimal choice differed per site. For instance, navigation time can
vary linearly or logarithmically with the number of subitems under each menu item.
As the optimal values of these features depend on the user population, the various
values must be tested anew for each new site. This can be accomplished with the
presented procedure.

By means of the framework, we compared the feature values that were found opti-
mal to the assumptions used by the menu optimization methods. This study revealed
that none of the methods used an optimal model. As stated, this can be detrimental to
the resulting menu structures. Therefore, we believe that menu optimization can be
improved substantially by application of the presented model selection method.

6.2.3 How can we automatically create problem-oriented menus?

In Chapter 4 we presented a novel way to represent the preferred reading order of a
set of web pages. So called stage models consist of a number of navigation stages.
Each stage represents a cluster of pages that play similar roles in the users’ navigation.
The ordering of the stages is such that users tend to start their sessions by viewing
a number of pages from the first stage, then visit some pages from the second stage,
etc. There is no preferred reading order for two pages from the same stage. Stage
models can be combined with traditional topic-based structures to construct problem-
oriented menus. The stage models form the layers of the menu hierarchies. When
users navigate from the root of the hierarchy to the deeper layers, the pages they
encounter become more and more specific. The topic-based structures determine the
position of the pages within each layer. Users use the topics to select pages that are
relevant to their information needs.
We presented an algorithm to automatically find a specific stage model for a user population on the basis of usage data. The algorithm searches for regularities in the order in which users visited the pages of a site. It computes for each page the average relative position of the page in the users’ sessions. Expectation maximization is employed to divide the pages in a number of stages on the basis of their relative positions. Stage models with various amounts of stages are tried and the best fitting model is selected. Finally, the stage model is optimized through bootstrapping. The initial stage model is compared to the page order in the individual sessions and improved where necessary.

We evaluated the stage discovery algorithm on the SeniorGezond site (SeniorGezond, 2007), for which a problem-oriented menu was created by experts. In an offline version of the site we replaced the problem-oriented menu with a basic menu that did not impose a reading order. Participants were asked to perform search tasks and their actions were recorded. The algorithm proved able to reconstruct the correct stage structure on the basis of these log files. In a second experiment the algorithm was applied to a site that did not yet offer a problem-oriented menu. Also in this domain the algorithm was able to generate an adequate stage model that was consistent with the natural reading order of the sites’ pages. In addition, we demonstrated how this stage model can be used to automatically construct a problem-oriented menu for the site. Simulation data allowed us to test the sensitivity of the algorithm to properties of the site and the log files. We found that the algorithm could find stage models for large sites, for sites with many stages, and for sites with noisy log files, provided that enough log data was available. These results show that stage models are a suitable means for creating problem-oriented menus and that the stage discovery algorithm can effectively learn stage models from log data.

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

Experiments of Miller and Remington (2004) and ourselves (Chapter 2) show that clicks on menu items that do not lead to a user's targets strongly increase navigation time. Therefore, reducing navigation errors can considerably contribute to the efficiency of hierarchical menus. In the introduction we formulated the hypothesis that navigation errors are often caused by inaccurate link descriptions. Inaccurate descriptions give users incorrect ideas of the content that can be reached by following the links, so that users cannot predict which links lead to their target information.

In Chapter 5 we proposed a model that describes how users navigate in the presence of accurate link descriptions as well as various types of inaccurate descriptions. Accurate descriptions are characterized by a usage pattern in which most users select the links that lead to their target pages. Other usage patterns indicate that descriptions are not correctly understood or that descriptions of links are confused.

We provided a method that determines which descriptions in a menu are inaccurate by comparing the model to the navigation patterns of the sites’ users. For all sessions in the log files the method estimates which target pages the users were looking for. It compares the menu items that the users should have selected to reach their
targets most efficiently to the items that were actually selected. The method determines the types of the mistakes that are made at the various locations in the menu and makes recommendations for how the descriptions can be improved.

We evaluated our approach on three web sites. The method was used to analyze the logs of the sites and its findings were shown to the webmasters of the sites. All webmasters found the analyses very useful for improving the menus. 64% of the problems that were identified by our method were assessed as relevant. A user experiment showed that on all sites the adaptations to the menus that the webmasters chose on the basis of the analyses significantly reduced the number of navigation mistakes. This confirms our hypothesis that navigation mistakes can be brought down by improving inaccurate link descriptions. Moreover, it indicates that our method can effectively help webmasters to improve hierarchical menus and, consequently, to reduce navigation time.

6.2.5 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?

Above we answered the four specific research questions. We will now return to our main research question and draw general conclusions about the optimization of hierarchical menus.

Our first observation is that one can optimize various aspects of menus and various types of menus. We optimized menus for users with two types of information needs: specific information needs and unspecific information needs. These two types of users have different goals during navigation and thus pose different requirements on a menu. Furthermore, menus consist of a hierarchical link structure and descriptions for the links. Previous research on menus focused primarily on link structures. We showed that inaccurate link descriptions have a large influence on navigation time, which means that efficiency can be further enhanced by improving link descriptions.

We have shown that the optimization of both types of menus as well as both aspects of menus can be accomplished by a model-based approach. The models that we used consist of two layers. The first layer comprises a generic model, that represents the requirements of the task. This thesis provided three examples of generic models: navigation behavior models for users with specific information needs, stage models for users with unspecific information needs and mistake models that described navigation patterns of accurate and inaccurate link descriptions. For each of these generic models we provided a method to create a specific model that represents a particular site and user population. When a specific model is created, the next step is to find a menu that is optimal according the model. Sometimes this is very straightforward, for instance, transforming a specific stage structure into a problem-oriented menu. In other applications optimization requires a considerable amount of computation, as is the case for navigation behavior models. Some optimization tasks even have to be performed by humans, such as optimizing inaccurate descriptions.

The model-based approach allows us to make a sharp distinction between modeling the needs of users and finding menus that optimally support these needs. This
6.3. Discussion and future research

In this section we discuss advantages and limitations of our work. In particular, we address the applicability of the methods that we have presented. In the last part of the section we identify unsolved issues and directions for future research.

Hierarchical menus can be implemented in a variety of ways. Some menus are simply hierarchies of HTML hyperlinks. Others are formed by, for instance, javascript or cgi programs. Although in this thesis we often spoke about menu items as 'links', the presented methods are not limited to menus implemented as HTML hyperlinks. The only restriction that our methods pose on a menu is that the opening of menu items happens on the server side rather than the client side, so that menu openings can be recorded in log files. For our experiments, the menus of the various sites were transformed to an internal format by custom made scripts. For each new site this required several hours of manual labor. A standard representation of menus would significantly facilitate this process. However, at present no such standard exists.

In the previous section we named several advantages of usage-based methods. However, the use of usage data also poses constraints on the applicability of the methods. Most importantly, usage-based methods suffer from the cold-start problem. When new content is added to a site or when a site has just gone online, no usage information is available. Consequently, optimization has to be delayed until a sufficient number of users have visited the new site or the new part of the site.

The storage of usage data that is required for our methods may raise questions
about privacy. We believe, however, that our methods can be applied while privacy remains reasonably well-protected. The data that we use is limited to sessions on a single site. This means that per user only a few requests are stored, so that relatively little can be inferred about individual users. This stands in contrast to data collected in proxies which includes all request that a user makes to any site. Moreover, our methods only use click stream data and do not require users to register or otherwise disclose personal information. For domains where privacy is so crucial that even regular log data cannot be stored for long periods of time, our methods must be made incremental. Instead of using large amounts of log data at once, incremental methods digest data directly when it comes in.

We put much emphasis on the evaluation of our methods. We evaluated each of our methods on the menus of real web sites. In all cases, these studies showed that the methods were able to substantially improve the menus. However, the evaluations were limited to case studies and user experiments. To test the quality of the optimized menus with real users the optimized menus must be placed on the sites. Comparison between navigation through the original and the optimized menus will allow an unbiased view on the effects of menu optimization on navigation of real users.

In this thesis various aspects of menus were addressed separately. Future research should explore how these aspects can be integrated in one system that helps a web master to optimize all aspects of a site’s menu. A particularly promising research direction involves the combination of menus for users with specific and unspecific information needs. Combined menus are urgently needed as most sites accommodate both types of users. Another issue that awaits further exploration is the integration of structure optimization with link description optimization. The current versions of the methods cannot be combined directly. The description optimization method enables us to assess the quality of descriptions in the current menu structure, but does not provide a way to predict the quality of descriptions in alternative structures. Finally, in this thesis we focussed mainly on the minimization of navigation time. Future work could investigate usage-based approaches to the optimization of other criteria, such as the visual design of menus or the consistency between the terminology of a menu and the rest of a site.

In this work we showed how explicit navigation models constructed from usage-data can be applied for menu optimization. Our experiments clearly demonstrate the promise of the model-based approach in this domain. However, the benefits of models are not inherently limited to hierarchical menus. We believe, therefore, that this type of methods will prove useful for a much wider range of applications. Future research should explore models that describe how users interact with other types of interfaces. These models will enable easy and unbiased optimization of these interfaces.