

NavOptim Coding: Web Navigational Construction to Minimise Navigation Effort

Dr Xiaoying Kong [[HREF1](#)], Faculty of Engineering [[HREF2](#)], University of Technology, Sydney, Australia [[HREF3](#)], Email: xiaoying.kong@uts.edu.au

Prof David Lowe [[HREF4](#)], Faculty of Engineering [[HREF2](#)], University of Technology, Sydney, Australia [[HREF3](#)], Email: david.lowe@uts.edu.au

Abstract

Web applications have rapidly become critical to the interaction that organisations have with their external stakeholders. A major factor in the effectiveness of this interaction is the ease with which users can locate information and functionality which they are seeking. Effective design is however complicated by the multiple purposes and users which Web applications typically support. In our earlier work we described a model for evaluating the overall navigation entropy of a Web application which provides a measure of the weighted effort required of users. In this paper we describe a navigational design method aimed at minimizing this navigational entropy. The approach uses a theoretical navigational depth for the various information and service components to moderate a nested hierarchical clustering of the content.

Introduction

Over the last decade the Web has become a key vehicle for accessing organisational applications. These applications often provide crucial business or government services, and hence the quality of the interaction is vital to their success. Given this, effective design of the web application interface is crucial.

This in turn raises the issue of what is actually meant by "effective design"? Effectiveness can be defined in terms of the ability to support the users' goals - yet for Web applications these goals are typically both complex and diverse, with different users having different expectations and objectives. Optimising the design becomes a difficult activity involving trade-offs of multiple constraints. Despite this the design of navigation structures is rarely treated as an optimization problem.

In our earlier work (Lowe et al. 2004) we described a model of the relationship between the intended user tasks, the navigational structure and the resultant overall navigational effort. This model can be used to support this optimization, and was inspired by entropy coding techniques (such as Huffman coding). Our representation was based around an evaluation of the weighted navigational effort required to access the various information and services within a Web application. The navigational effort is dependant both upon the length of the navigational path as well as the effort required at each navigational step. This is, in turn dependant upon the structure of the site and the semantic clustering within the site.

This model provides a mechanism for evaluating the overall quality (or navigational entropy) associated with a particular design. It is therefore possible to compare possible navigational designs to determine which is preferred. Whilst the nature of the model makes the identification of a theoretical optimal design an intractable problem, one possible way of creating improved navigational structures would be to perturb existing designs and evaluate the impact on the overall navigational effort.

In this paper we consider an alternative use of the model. We decompose the model into the key aspects which influence the overall navigation entropy, and propose a design approach based on this decomposition. In the next section we begin by discussing related work, including approaches to the design of Web navigation structures. In section "[Minimising Navigational Effort](#)" we provide a brief overview of NavOptim coding, and then in section "[Navigational Design](#)" we describe our approach to navigational design and how it can be applied. Finally, in section "[Conclusions](#)", we discuss future directions for this research.

Related Work

As discussed above, we wish to be able to design Web navigation structures which optimise the quality of the user experience. A major factor in this user experience is the navigational structure of the Web application, and how well this supports the ability of users to actually locate and access relevant information or services. Most existing design approaches are based on intuition, general heuristics, or experimental refinement. To illustrate this we will look at current navigational design approaches.

Structured Design Approaches

Numerous approaches have been developed for performing the design of Web system navigational structures. Some of the earliest work emerged out of the hypertext community and was based on Entity-Relationship modelling. Relationship Management Methodology (RMM) (Isakowitz et al. 1995) is an example which provides a structured design methodology for hypermedia applications. RMM focuses on modelling the underlying content, the user viewpoints onto this content, and the navigational structures that interlink the content. The design of these navigational structures (step 3 in the RMM method) is a subjective process where the underlying associative relationships between the content are analysed and "items of interest" are grouped together - though little guidance is provided about how to optimise this grouping. Other approaches, such as Object-Oriented Hypermedia Design Model (OOHDM) (Schwabe et al. 1998) are based on object-oriented software modelling approaches, and provide richer information representations. Other similar examples include an object-oriented design method for hypermedia information systems (EORM) (Lange 1994) and a structured navigation design method for Intranet applications by Lee (1997). A user-centred design method for web sites (WSDM) (De Troyer et al. 1997) attempts to take these approaches one step further, by beginning more explicitly from user requirements, but these are only addressed in a very rudimentary fashion. In general, these techniques were either developed explicitly for modelling information in the context of the Web, or have been adapted to this domain. More recently, Web Modelling Language (WebML) (Ceri et al. 2000) has begun to amalgamate these concepts into a rich modelling

language for describing Web applications. However, despite its aim to support comprehensive descriptions, as with the above techniques the focus of WebML is very much on content modelling rather than describing the functionality that is a key element of most current commercial Web systems. One of the few approaches that attempts to integrate content representation with functionality is (Takahashi et al. 1997).

These approaches have typically undertaken the navigation design based on a subjective view of the designer with regard to how users are likely to want to interact with the information. In most cases this is not well informed by the underlying requirements that drive this architecture - and especially user objectives. An exception to this is some of the work extending OOHDM, to include tools such as user scenarios and use cases which look at how a system is likely to be used (Guell et al. 2000). They still do not, however, provide any formal way to ensure that the navigation structures are theoretically optimal. One interesting alternative is work on approaches to formally representing the navigation structures within Web system. For example, Hadez (German et al. 1999, German et al. 2000 looks at the use of formal methods (using the Z notation) to specify conceptual, structural and perspective schemas. Whilst a formal representation is potentially amenable to optimisation, this has yet to be considered by the authors of Hadez.

Usage Analysis Approaches

There has been substantial research investigating design of Web system, including navigation design, based on either likely usage patterns or an analysis of actual usage. The most common example of this is user-centred design (Norman et al. 1986, Noyes et al. 1999). Essentially this involves a strong focus on the user, including evaluation of early design prototypes. As with the structured design approaches this leads to refinement of the navigation structures, but does not guarantee a theoretical optimisation. In usage-centred design (Constantine et al. 1999) the focus is on usage of the system, rather than users of the system - though again the optimisation is subjective.

Optimisation

In each of the above cases the approach is focusing on subjective analysis and refinement of the navigational structure. We are arguing that this is inappropriate. In effect, the subjectivity lies not in the navigational structure, but rather in the significance of the various information and services that are provided to users and the design choices about the usage patterns which we wish the system to support. Once this is known (albeit through a subjective interpretation) we ought to be able to design the navigational structure which provides the theoretical minimum average navigational effort required by users in accessing these information sources and services - assuming we have a reliable measure of navigational effort.

In our earlier work (Lowe et al. 2004) we described a model of the relationship between the intended user tasks, the navigational structure and the resultant overall navigational effort. This model can be used to support the identification of the optimal navigational structures - i.e. those which minimize the overall navigational effort for a set of designated user tasks.

As we mentioned previously, this approach was inspired by entropy coding techniques such as Huffman coding (Huffman 1952). With Huffman coding a measure of the entropy associated with a particular encoding of a character stream is derived, and then an algorithm is provided which selects the theoretically optimal character coding to minimize this entropy. Unfortunately, the measure of the navigational entropy which we have developed previously is not tractable to an algorithmic identification of the theoretic optimum. This is primarily due to both the nature of the navigational tree not having a single n-ary nature, as well as the coupling between the nodes of the navigational tree (i.e. whereas in a Huffman code, the nodes are independent, in our navigational tree, moving a node not only affects the navigational path, but also the related coupling with other nodes and therefore the overall semantic cohesion and hence navigational effort required).

In the following sections we describe an approach to using our model to empirically optimize the navigational structures. We begin by considering the model itself, and then go on to describe our technique.

Minimising Navigational Effort

Modelling of Navigational Effort

The following is a brief summary of the approach which was introduced in (Lowe et al. 2004).

Given a particular navigational structure, and a particular user task (i.e. locating a specific item of information or a specific functional service) we can determine the navigational path probabilities, as shown in Figure 1.

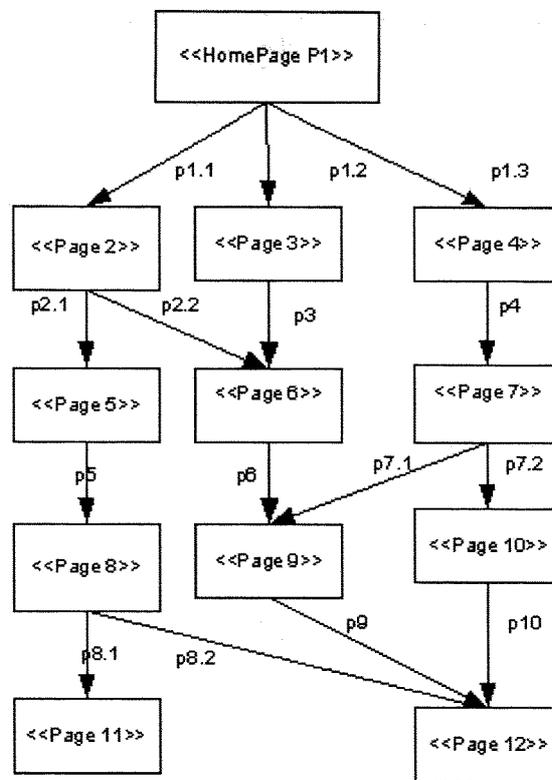


Figure 1: Architecture with probabilities for single task case.

Each possible navigational step has a particular probability associated with it. For example, consider the case where a user task involves accessing information on <<Page-12>>. If the user is currently accessing <<Page-2>> then the options available to them are to follow links to either <<Page 5>> or <<Page 6>>. The probability of each of these occurring is $p_{2.1}$ and $p_{2.2}$ respectively. These probabilities represent the likelihood of the user choosing to follow each link, and are dependent upon the users' perception of the semantic cohesion between their navigational objective and the content within that navigational branch (as typically represented through the link source anchor). This is formalized as:

$$p_{n,i} = k * SmtCoh(V_{n,i}, V_{TaskCase}), \quad i = 1 : m \quad (1)$$

$$\sum_{i=1}^m p_{n,i} = 1 \quad (2)$$

where $V_{n,i}$ is the information vector available to the user regarding navigating from link i when on page n .

This then gives:

$$p_{n,i} = \frac{SmtCoh(V_{n,i}, V_{TaskCase})}{\sum_{j=1}^m SmtCoh(V_{n,j}, V_{TaskCase})}, i = 1 : m \quad (3)$$

This can be readily interpreted as capturing the concept that the likelihood of a user making a correct navigational choice increases (and the effort decreases) as the relationship of their goal to a particular path is strengthened. For example, given two navigational choices, a correct one with a semantic cohesion of 0.8 and an incorrect one with a semantic cohesion of 0.7, the probability of a correct choice will only be 0.53. Conversely, if the cohesions were 0.9 and 0.2 respectively then the probability of a correct choice becomes 0.82.

Using the entropy from the probability distribution, the user navigation effort for a task case $TaskCase_j$ can then be represented as:

$$H(j) = - \sum_{i=1}^m p_i \log p_i \quad (4)$$

where j is the task case number, and i varies from the navigational entry point (normally the site homepage) to the navigational exit point (normally the page which satisfies the users' information or service objective associated with the given task case).

Suppose there are n task cases which capture the overall design objectives for the navigational structure. For each task case $TaskCase_j$ ($j=1\dots n$), we can define a significance factor $Sgnf(j)$ which captures the weight of that particular task case (i.e. it will be more important for the design to support particular users or particular tasks more effectively). The navigation effort for $TaskCase_j$ is represented as $H(j)$. The weighted navigation effort (or navigational entropy) for the entire system H_{sys} is the weighted sum of the efforts of individual task cases. That is:

$$\begin{aligned}
 H_{sys} &= \sum_{j=1}^n [Sgnf(j) * H(j)] \\
 &= - \sum_{j=1}^n [Sgnf(j) * (\sum_{i=1}^m p_i \log p_i)]
 \end{aligned}
 \tag{5}$$

This formula provides a basis for evaluating particular designs. We have begun building tools which allow us to take a particular navigational design, specify a given set of task cases and have the tool calculate the overall navigational entropy.

Optimisation of Navigation

Whilst the above approach means that we are able to compare different navigational designs, it still does not allow us to determine the theoretical optimum structure. We can compare this to data compression algorithms in information theory. The message "Msg" to be transmitted consists of M symbols. Each symbol S_i ($i=1\dots M$) has a frequency or a probability P_i ($i=1\dots M$) to appear in the message "Msg". Each symbol is represented and transmitted by a "codeword" (or a number of bits) from a code source of size N_{code} . Each codeword has a length which is the number of bits in the codeword (Applebaum 1996). We can represent the character coding as a binary tree. The effort of transmission of this message is proportional to the length of the message. That is the total amounts of bits in all symbols in this message. According to Shannon (1948):

$$H = - \sum_{i=1}^M P_i \log P_i
 \tag{6}$$

where H is the entropy of an information source, where P_i is the probability of each possible character occurring in the string. This gives the theoretical minimum for the average number of bits to code each character in the string. This is analogous to the theoretical minimum for the navigation effort required for a given task case, as shown in [equation 5](#). Huffman coding (Huffman 1952) provides an algorithm that determines, for a given set of characters with known probabilities of occurring, a set of character codes such that this entropy (i.e. transmission effort) will be the theoretically minimum possible. We would like to be able to identify an equivalent algorithm for determining the navigational structure which achieves the theoretical minimum navigational effort. We have however found this problem to be intractable. This is for several reasons. The first is that whilst the Huffman coding tree is a binary tree, the NavOptim navigation tree is an n -ary tree where n varies in each branch of the tree. Even more problematic is that whilst the characters in the Huffman coding are independent, the same is not true of the nodes in the NavOptim tree. This is because the individual entropies of each navigational path are related to the location of the all nodes within the tree. In other words, we could swap the code representation for any two characters which have the same number of bits and the entropy would not be affected. If however we swap two nodes within a navigational tree (even if they exist at the same navigational depth) then the overall semantic cohesion relationships will be affected and hence the navigational effort will be modified. This then leads to the fundamental question being considered in this paper - is there an algorithm which provides a (sub-optimal) approximation to the theoretically optimal navigation tree? It is this question which we address in the next section.

Navigational Design

Given the intractability of finding a navigational design algorithm which can be mathematical proven to result in the minimization of [equation 5](#), we can turn our attention to approaches which approximate an optimal design, but which are strongly guided by the underlying theory. We begin by analyzing the mathematical models captured by [equations 3, 4 and 5](#). From these we can make several key observations (both rather intuitive, but with useful implications about how they are handled). The first is that the entropy is strongly coupled to the overall navigational depth of specific content. As content is located deeper within the navigational tree, the navigational effort to locate it will typically increase (all other factors being equal). The second is that the entropy is strongly related to the semantic cohesion. As we create stronger cohesion within branches of the tree, we enable clearer navigation choices and hence reduced entropy and effort. We will consider how each of these aspects can be utilized, and then look at integrating them into an overall algorithm.

Optimal Navigation Depth for Content

Binary Tree vs N-ary Tree in web structure

As discussed above, the typical examples of information-theoretic optimal coding structures (such as Huffman trees) are binary trees. Web structures however tend to be N -ary trees. This raises the issue of the optimal selection of N at each branch - i.e. the number of links available from a given node. Considering [equation 4](#), as N increases, we can achieve fewer steps in the navigational path (m decreases) and so the entropy will trend downwards. However, considering [equation 3](#), as N increases, the number of choices at each navigational step increases, and the probability of making the correct choice drops, and hence the entropy increases.

There is considerable existing literature which discusses the issue of the choice of information architecture depth and breadth from a design perspective. It is generally found that it is easier for users to reach the target page in a broader architecture than in a deeper style (Bernard 2004, Norman et al. 1988, Powell 2002). Research has found that the deeper the navigational level the more a user has to reply on short

term memory. Sites with deeper navigational structures typically also have more general link descriptions at the top level. This may make it more difficult for users to choose the correct navigation path required to satisfy a particular user's task case (Larson et al. 1998, Bernard 2004).

It is argued that the average depth of a node in a hypertext space is 4 (Shneiderman et al. 1989). Powell suggests that a depth of 3 is more appropriate (Powell 2002). For sites that must have deeper structures (4 or more levels), Bernard (2004) argues that users browsing for specific information they want will find this information faster if the structure is concave. That is it should be broad at the top level and at the lowest or base level, while the interior of the web structure should have a narrow level of breadth.

NavOptim Tree construction

In considering the navigation depth for various content we can begin by returning to information theory. The codes with minimal transmission effort are those which have the smallest average code length. The appropriate length of each codeword can be calculated by the following inequality (Applebaum 1996).

$$-\log_{N_code}(P_i) \leq Length_i < 1 - \log_{N_code}(P_i) \quad (7)$$

where

- $Length_i$ is the length of the codeword to represent the symbol S_i ;
- N_code is the size of the code source;
- P_i is the probability of the symbol S_i appearing in the message.

If we can apply this to navigation design then we can identify an appropriate depth for given content (based on the significance of the content for satisfying user tasks).

Suppose a web site has the following attributes: Total number of web pages (web user interfaces) of this site is $N_totalpages$. The web pages can be represented as $\{Page1, Page2, \dots, Pagei, \dots, PageN_totalpages\}$. Each web page $Pagei$ ($i=1 \dots N_totalpages$) is associated with a probability $Probi$ which represents users' accessing frequency to this page. The number of layers in this site is N_layer . The depth of a web page $Pagei$ ($i=1 \dots N_totalpages$) to the root page will be represented as $Depthi$. For example, if a page $Page43$ is on *Layer 2* of the site, the depth of this page will be 2. $Depth43=2$. Using this information it is possible to identify the optimal depth of content - $Depthi$ - based on access, but ignoring issues of semantic coupling. This can be achieved as follows:

Inputs:

- Total number of web pages: $N_totalpages$;
- Web pages: $\{Page1, Page2, \dots, Pagei, \dots, PageN_totalpages\}$;
- Probabilities of each page being accessed: $\{Probi\}$

Outputs:

- Total layers of the web site: $N_totallayer$;
- Maximum fan-out of pages: N ;
- Depths of each web page to the root: $\{Depthi\}$;
- Positions of each web page.

Step 1: Determine the total layers and page fan-out. If we follow the 3-clicks-rule (Powell 2002) and concave tree structure suggestion by Norman et al. (1988) then in the vertical dimension the total number of layers in the web site $N_totallayer$ could be set to 3 or 4. Not losing its generality, we use a variable $N_totallayer$ here. In the horizontal dimension, a concave tree structure has broader leaves (pages) on top layer and bottom layer. The middle layers have narrow breadth. Each individual page in Layer 1 to $N_totallayer-1$ can have outgoing links of a number less than or equal to N . The maximum fan-out of pages on the bottom layer can be more than N . The entire site structure will be a N -ary tree.

Step 2: Calculate the optimal depth. In an optimal message coding, the length of each codeword satisfies the inequality shown in [equation 7](#). The depth $Depthi$ of a web page $Pagei$ to the root page is equivalent to the length of a coding message. Therefore, with the visitor accessing probability $Probi$ of the page, $Depthi$ can be calculated using the following inequality:

$$-\log_N(Prob_i) \leq Depth_i < 1 - \log_N(Prob_i) \quad (8)$$

$Depthi$ is the smallest natural number satisfying the above inequality. The result of this calculation would be the optimal navigation depth for each information node. Note that this provides an indication of the preferred layer. The actual layer may end up being different on the basis of the clustering described in the next step.

Clustering adaptation of depth

Having identified the optimal depth of content, this then needs to be moderated by the need to appropriately cluster content to form effective semantic portioning (as implied by [equation 3](#)). There is a substantial body of work on approaches to clustering, including how it

can be applied in the context of Web applications. Good examples of this work include (Loia et al. 2003, Friedman et al. 2004). Possibly more relevant is work on hierarchical clustering, particularly the use of Multitrees (Furnas et al.1994, Feldmann et al. 2003) - a form of directed acyclic graph which includes easily identifiable tree substructures.

In essence, by undertaking effective semantic clustering we are improving the cohesion within collections and decreasing the coupling between collections. If we consider again equation 3, then the probability of a correct navigation choice being made ($p_{n,i}$) can be increased by maximizing $SmtCoh(V_{n,i}, V_{tasecase})$ and minimizing $SmtCoh(V_{n,j}, V_{tasecase})$, $|i-j| > 0$. as

Figure 2 illustrates this issue graphically, where the spatial positioning of the pages represents the semantic distance between the pages, and the highlighted page is the target of a given task. In Figure 2a, the target is much more closely related to the central concept of cluster 2 than it is to cluster 1, and as a result the probability p of a correct navigational choice between cluster 1 and cluster 2 will be high. In Figure 2b, the target is well correlated to both cluster 1 and cluster 2, and as a result the probability p will be much lower.

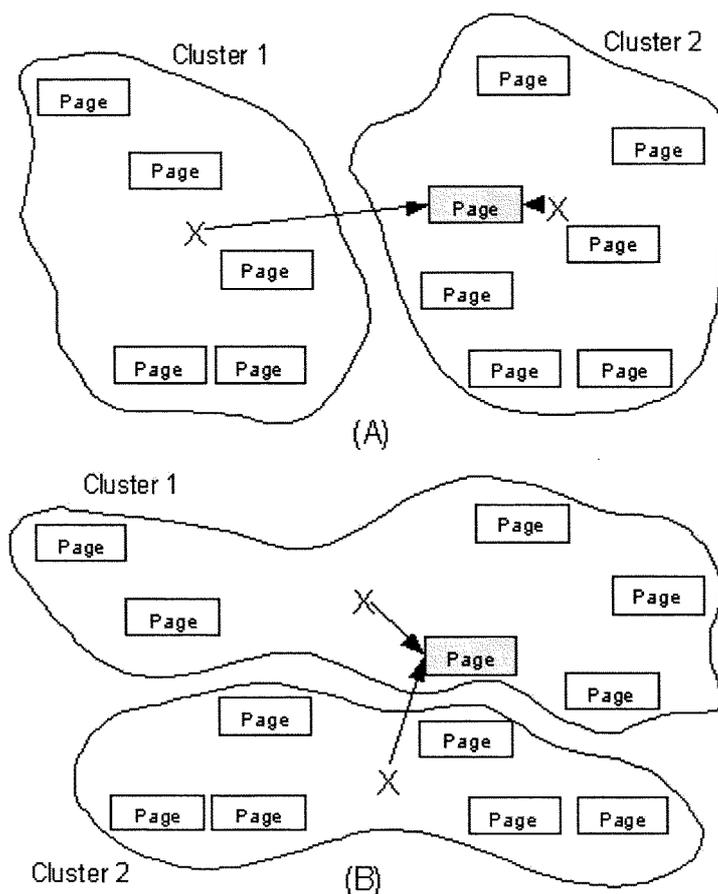


Figure 2. Example clustering

Resultant design algorithm

So, how are the above two concepts (optimal depth of content and clustering) merged into a resultant algorithm? We are proposing a bottom-up clustering which is guided by iterative calculation of the overall site entropy H_{sys} . The proposed algorithm is:

Step 1: Create initial clusters and structure. Semantic clustering is individually applied at each level of the initial content structure (which was created by determining the optimal depth of each content component) commencing from the bottom layer. Note that some clusters may contain a single node, whilst other clusters may contain many nodes. At each successively higher layer, the lower level clusters become nodes in that layer. The number of clusters to create at each layer and the maximum number of nodes in each cluster, will be determined by the required fan-out at each level. As each layer is clustered, it may require an iterative reclustering of the lower layers if it creates a fan-out that exceeds the specification.

These initial clusters are then linked to create a draft navigational structure. Figure 3 provides a simplified example where we have two layers, and the top-level fan-out is 3 and the bottom-level fan-out is 2 (i.e. layer 2 can only have a maximum of 6 clusters, and layer 1 can only have a maximum of 3 clusters). In Figure 3a, whilst we have 6 clusters at level 2, these are not evenly grouped in level 1, leading to cluster 1.2 having a fan-out which is too high. The relevant sections of Layer 2 are then reclustering to create Figure 3b. Note that this stage of the algorithm is almost certain to create sub-optimal clustering as it enforces content to belong to a certain layer even where the parent cluster contains no content of its own (as is the case with cluster 1.3).

Step 2: Adjust cluster layering. For each cluster, if it is the only cluster that exists within its parent cluster, then it should be elevated. For example, the content of cluster 2.6 can be validly elevated to belong to cluster 1.3.

Step 3: Iterative refinement. The structure can then be iteratively refined by considering the position of each content node and comparing the navigation entropy H_{sys} for the overall structure against H_{sys}' for the structure as it would be if that node was moved into every possible

cluster. The minimum H_{sys} is determined and the content node is moved if appropriate. This process is repeated for every node and cluster, and then process is repeated until no further improvement in H_{sys} can be made.

We are currently developing a tool to perform this incremental refinement and will be undertaking evaluations of the results.

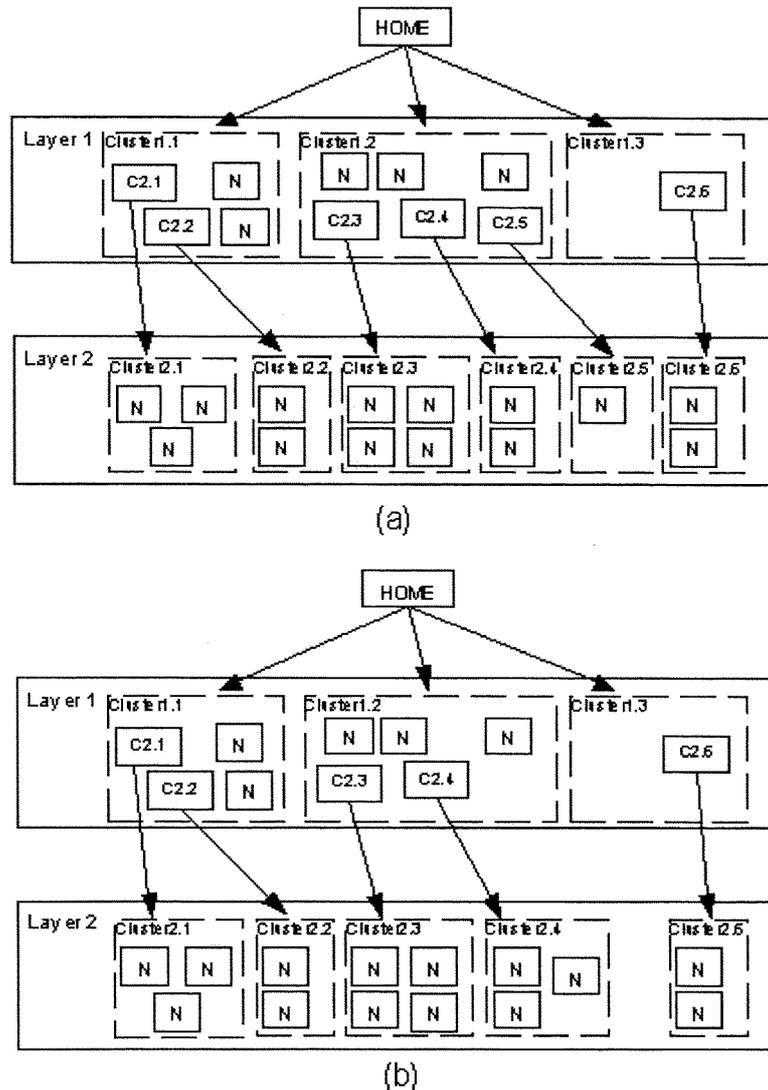


Figure 3. Example initial clusters and structuring

Conclusions

In this paper we have discussed an approach to the design of navigational structures based on the optimisation of a navigational effort metric weighted by the task significances. This approach potentially will lead to a reduction in the average effort required to locate information or services within websites. Future work will be focusing on the refinement of the algorithm to better integrate the competing components (navigational depth versus content clustering). We are also looking at the development of tools to facilitate the design process and the simulation of different websites to see how they are optimised using our approach, followed by subjective evaluation to determine whether this does indeed lead to qualitative improvements.

References

- Applebaum, D. (1996) Probability and information : an integrated approach. Cambridge University Press, Cambridge, U.K,
- Bernard, M. (2004) Criteria for optimal web design, designing for usability. Available online [[HREF5](#)]
- Ceri, S., Fraternali, P. and Bongio, A.(2000), Web Modeling Language (WebML): a modeling language for designing Web sites. in Proceedings of WWW9 Conference, Amsterdam, 2000, 137-157.
- Constantine, L.L. and Lockwood, L.A.D.(1999) Software for Use: A Practical Guide to the Essential Models and Methods of Usage-Centered Design. Addison-Wesley, Reading, MA .
- De Troyer, O. and Leune, C.(1997), WSDM: A user-centered design method for Web sites. in 7th International World Wide Web Conference, Brisbane, Aust, 1997, Elsevier, 85-94.

- Feldmann, M. and Wagner, R. (2003), Structuring with multitrees: a technical concept for improved navigation in hypermedia. *Wirtschaftsinformatik*, 45 (6). 589-598.
- Friedman, M., Kandel, A., Schneider, M., Last, M., Shapira, B., Elovici, Y. and Zaafrany, O.(2004), A fuzzy-based algorithm for Web document clustering. in *NAFIPS 2004. 2004 Annual Meeting of the North American Fuzzy Information Processing Society, Banff, Canada, 2004, IEEE*, 524-527.
- Furnas, G.W. and Zacks, J. (1994), Multitrees: enriching and reusing hierarchical structure. in *ACM Conference on Human Factors in Computer Systems, Boston, USA, 1994, ACM*, 330-336.
- German, D.M. Hadez (2000) *A Framework for the Specification and Verification of Hypermedia Applications*, University of Waterloo, 2000.
- German, D.M. and Cowan, D.D. (1999) *Formalizing the specification of Web applications. Lecture Notes in Computer Science*, Springer Verlag, 1727. 281-292.
- Guell, N., Schwabe, D. and Vilain, P.(2000), Modeling Interactions and Navigation in Web Applications. in *World Wild Web and Conceptual Modeling'00 Workshop - ER'00 Conference, Salt Lake City, USA, 2000*, 115-127.
- Huffman, D.A. (1952) A method for the construction of minimum redundancy codes. *Proc. IRE*, 40 (9). 1098-1101.
- Isakowitz, T., Stohr, E. and Balasubramanian, P. (1995) RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, 38 (8). 34-44.
- Lange, D.(1994), *An Object-Oriented Design Method for Hypermedia Information Systems*. in *HICSS-27: Proc of the Twenty Seventh Hawaii International Conference on System Sciences, Maui, Hawaii, 1994*.
- Larson, K., & Czerwinski, M. (1998) *Web page design: Implications of memory, structure and scent for information retrieval*. 1998. Available online [[HREF6](#)]
- Lee, S.C.(1997), *A Structured Navigation Design Method For Intranets*. in *Third Americas Conference on Information Systems, Association for Information Systems (AIS), Indianapolis, 1997*.
- Loia, V., Pedrycz, W. and Senatore, S. (2003) P-FCM: a proximity-based fuzzy clustering for user-centered Web applications. *International Journal of Approximate Reasoning*, 34 (2-3). 121-144.
- Lowe, D. and Kong, X. (2004), *NavOptim Coding: Supporting Website Navigation Optimisation using Effort Minimisation*. in *2004 IEEE/WIC/ACM International Conference on Web Intelligence, Beijing, China, 2004, IEEE Computer Society*, 91-97.
- Norman, D.A. and Draper, S.W.(1986) *User-Centered Design*. Lawrence Erlbaum Assoc, Hillsdale, N.J., 1986.
- Norman, K.L. and Chin, J.P. (1988) The effect of tree structure on search in a hierarchical menu selection system. *Behaviour and Information Technology*, 7. 51-65.
- Noyes, J.M. and Baber, C. (1999) *User-Centred Design of Systems*. Springer Verlag, 1999.
- Powell, T.A. (2002) *Web Design: The Complete Reference*. McGraw-Hill, Sydney, 2002.
- Schwabe, D. and Rossi, G. (1998), *Developing Hypermedia Applications using OOHDM*. in *Workshop on Hypermedia Development Processes, Methods and Models (Hypertext'98), Pittsburgh, USA, 1998*.
- Shannon, C.E. (1948) A mathematical theory of communication. *Bell System Technical Journal*, 27. 379-423, 623-656.
- Shneiderman, B. and Kearsley, G. (1989) *Hypertext hands-on! an introduction to a new way of organizing and accessing information*. Addison-Wesley, Reading, Mass, 1989.
- Takahashi, K. and Liang, E.(1997), *Analysis and Design of Web-based Information Systems*. in *7th International World Wide Web Conference, Brisbane, Aust, 1997*.

Hypertext References

- HREF1 <http://services.eng.uts.edu.au/~xiaoying/>
- HREF2 <http://www.eng.uts.edu.au/>
- HREF3 <http://www.uts.edu.au/>
- HREF4 <http://services.eng.uts.edu.au/~dbl/>
- HREF5

<http://psychology.wichita.edu/optimalweb/>

HREF6

<http://www.research.microsoft.com/users/marycz/chi981.htm>

Copyright

Xiaoying Kong and David Lowe © 2005. The authors assign to Southern Cross University and other educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to Southern Cross University to publish this document in full on the World Wide Web and on CD-ROM and in printed form with the conference papers and for the document to be published on mirrors on the World Wide Web.