Adaptive Web Navigation Construction: A Machine Learning Approach
Yinjun Gu, Xiaoying Kong, David Lowe
University of Technology, Sydney

Adaptive Web Navigation Construction: A Machine Learning Approach

Yijun Gu[1], Xiaoying Kong[2], David Lowe[2]

[1] Dept of Computer Science and Technology,
Beijing Institute of Technology, China

[2]Faculty of Engineering, University of Technology, Sydney, Australia

Email: guyjun@gmail.com, xiaoying.kong@uts.edu.au, david.lowe@uts.edu.au

ABSTRACT

Support for effective stakeholder interaction with an organization's web site has increasingly played a critical role in business success. An important factor in this effective web interaction is the ease with which a user can locate information and complete tasks within the web navigation structure. In this paper, we present a web navigation design method which adaptively constructs the web navigation structure using users task cases. A machine learning approach is applied to estimate the user's tasks and construct the web navigation based on tasks.

Keywords

E-business design, Web design, adaptive web navigation

INTRODUCTION

The Web has become a key vehicle for accessing organisational applications over the last decade. Web applications often provide crucial business, organisation or government services, and hence the quality of the interaction is vital to their success. Effective design of the web application interface is crucial, and a key aspect of this effectiveness is the extent to which the application interface, and particularly the navigation structures, support users in achieving their goals. This is complicated by the characteristic that different users have different expectations and objectives.

Optimizing the web navigation structure to meet different web users' needs becomes a difficult activity involving tradeoffs of multiple constraints. 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).

In essence, this earlier approach proposed a metric that gave an estimate of the expected navigation effort required of a user in achieving a specified task (such as navigating to a particular target item of information). This metric was calculated by looking at each navigational step along the overall path required for the task and estimating the difficulty associated with identifying the correct navigational link – which in turn was based on the number of different options available at that point, and the extent to which the correct option was aligned with the ultimate navigational destination (i.e. how difficult it would be to intuit from a specific link whether it would lead to the correct destination).

The result would be that different navigational structures could then be compared based on the extent to which they minimised the navigational effort required for a particular task. By determining weightings for different tasks based on their perceived significance, a comparison of the support of different structures for this weighted basket of tasks could also be achieved – leading to an effective tool for determining the most effective structure from a candidate set.

The earlier work then went on to propose an algorithm for applying these metrics prescriptively to determine an approximation of the theoretically optimal navigational structure. Whilst this approach is effective in terms of constructing a globally optimised navigation structure, it does not necessarily produce a structure which is optimal for a particular individual, nor appropriate given changing patterns of usage for a site.

In this paper we present an alternative approach based on constructing the navigation structure dynamically based on adapting to individual users' activities.

In the next section we begin by discussing related work in web navigation structure design approaches. In section 3 we describe our work to adaptively construct the web structure. Finally, in section 4, we discuss future directions for this research.

LITERATURE REVIEW

Our work aims to design web navigational structures which minimize the navigational effort required by users. Most existing design approaches are based on intuition, general heuristics, or experimental refinement. Current Web navigational design approaches include: structured design, usage analysis approaches and machine learning approaches.

Early approaches in web structure design tended to emerge from the Hypertext community and evolved out of work on Entity-Relationship modelling – particularly in terms of modelling the information domain associated with applications. For example RMM (Relationship Management Methodology)(Isakowitz *et al.* 1995) claims to provide a structured design methodology for hypermedia applications. In reality, the focus is very much 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.

OOHDM (Object-Oriented Hypermedia Design Model)(Schwabe *et al.* 1998) is a similar approach, though somewhat richer in terms of the information representations and based on object-oriented software modelling approaches. Other similar examples include EORM (Lange 1994) and work by Lee (1997). 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, work on WebML (Web Modelling Language) (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, the focus (as with the above techniques) 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.* 2000, German *et al.* 2001) 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.

There has been substantial research investigating the design of Web systems, 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 (Constantine *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.

A new direction of optimization approaches is to construct the navigation structures dynamically and adaptively to meet the users' needs. These approaches mainly use machine learning. The web system will learn from users' navigation experience and construct the navigation structure based on users' objectives. Among these approaches, Chen (1996) applied usage mining and introduced the notion of a mining algorithm for a Maximal Forward Chain of web pages. The WUM system (Spiliopoulou *et al.* 1998) applies sequence mining to analyze the navigational behavior of users in a web site. WUM also supports an integrated environment for user navigation history log preparation, querying and visualization. Cooley et al. (Cooley *et al.* 1999) describe various data preparation schemes for facilitating web mining. Recent advances and a more detailed survey on various aspects of web mining spanning content, structure and usage discovery can be found in (Kosala *et al.* 2000, Masand *et al.* 2000, Punin *et al.* 2001). One final approach worth mention is that by Andreas Thor, who tried to construct the navigation based on Decision trees (Thor *et al.* 2004).

Our work in this paper presents a web design method to adaptively construct the web navigation structure to fit the users' tasks. The method is based on usage centered design (Lowe *et al.* 2004). The navigation structure is optimized by the integration of usage mining and content mining. The web structure is built dynamically and adaptively for each individual user. Therefore the navigation structure is optimal for every individual user. The detailed design method is described in the following section.

ADAPTIVE WEB NAVIGATION DESIGN ALGORITHM, A MACHINE LEARNING APPROACH

Our web navigation design includes the following phases: estimation of user task cases; web pages positioning; construct the site navigation structure.

Estimation of user task cases

Our web design is based on usage-centered design Usage-centered design makes use of task cases, which describe user intention and system responsibility (Lowe *et al.* 2004). In our early work, web designers subjectively identify task cases by analyzing and grouping potential users' (stakeholders') intentions (Lowe *et al.* 2004). See Figure 1.

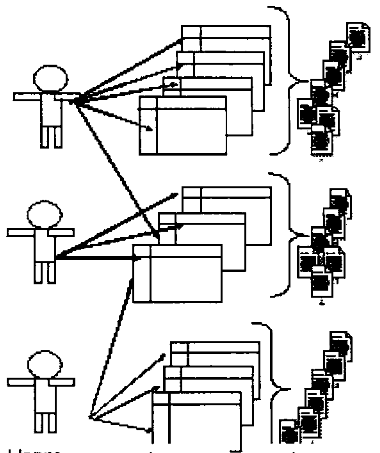

Figure 1: Task cases and web pages

In this paper, the user task cases are estimated automatically using the machine learning approach. The user task cases are analyzed from the user navigation log. The history of users' navigation path is recorded in user log.

A user's navigation history can be represented as a set G=(G1,G2,... Gi,...Gn). Gi (i=1:n) is a navigation chain. Gi can be presented as two sets Gi =(PGi, RGi). PGi is a finite non-empty set of web pages of the web site. RGi is a set of links to represent the order and relationship of visited pages.

PGi = { P1, P2, ..., Pm}

The relationships can be represented to a set E:

$$\mathrm{E}(P_i, P_j) = \begin{cases} 1 & \text{if} \quad P_i \text{ links to } P_j \\ \infty & \text{if} \quad P_i \text{ does not link to } P_j \end{cases}$$

See Figure 2.

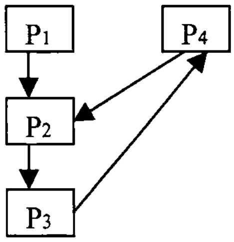

Figure 2: Web pages and linking relationships

Users' task cases are estimated by clustering the users' paths. Users' navigation may contain some "noise" (i.e. pseduo-random navigation events that are unrelated to the users' intentions). Before clustering pages, navigation noise should be reduced. We assume that users spend less time in unrelated pages. The time threshold for noise pages is set to be σ. The reduction of navigation noise is achieved by the removal of those web pages that a user stay in for less than σ. The algorithm of navigation noise reduction is shown in Figure 3.

After removal of the unrelated noise pages, the remaining pages are clustered into task cases by analyzing the page relationships with the users' navigation behavior. The problem is set out as following:

Assemble all the remaining pages to a set PN:

PN={ P1, P2, ..., Pn}

Suppose there are k task cases in the web site. The task cases are represented to a set TC:

TC={TC1, TC2, ..., TCk}

and

$TC_1 \cup TC_2 \cup \ldots\ldots \cup TC_K = PN, \quad TC_i \neq \emptyset$.

Task case TC_i is estimated by clustering pages as following:

Define a set U to represent the clustering results:

$$U = (u_{ij})_{n \times k}$$

where $u_{ij} = \begin{cases} 1 & if \quad P_i \in TC_j \\ 0 & if \quad P_i \notin TC_j \end{cases}$ (1)

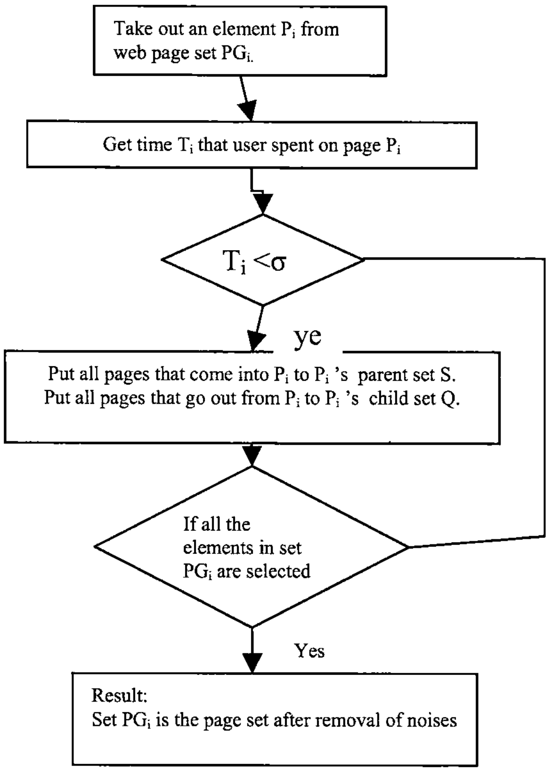

Figure 3: Navigation noises reduction algorithm

Generally, the clustering problem is to group a set of n elements to k independent clusters. Here a web page cluster is a task case TC_i. The cluster constraint is:

$$TC_1 \cap TC_2 \cap \ldots\ldots \cap TC_k = \emptyset.$$

In a web structure, a web page may belong to more than two task cases (see Figure 4). The above clustering constraint is not met in this case..

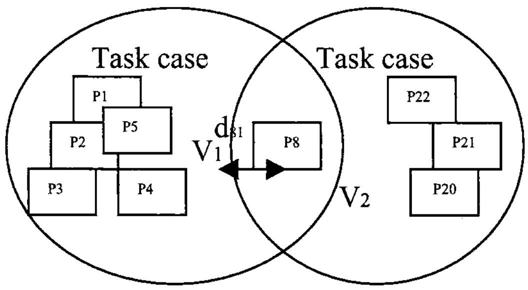

Figure 4: A web page belong to 2 task cases

To solve the web page clustering problem, we use fuzzy clustering by minimizing the following target function $J_m(U,V)$ (Chen 2005, Gu 2005).

$$\min J_m(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^m d_{ij}^2(p_i, v_j) \qquad (2)$$

with the constraint:

$$\begin{cases} \sum_{j=1}^{K} u_{ij} = 1 & \forall i \\ u_{ij} \in [0,1] & \forall_{i,j} \end{cases} \qquad (3)$$

where *n* is number of web pages, *k* is number of clusters or task cases. v_j is the centre of the cluster TC*j* (refer to Figure 4). The distance of a web page P*i* to the centre of a cluster TC*j* can be presented as $d_{ij}(P_i, v_j) = \|P_i - v_j\|$. *m* is an integer between 1 and ∞. *m* represents the fuzzy degree of U. Larger the integer m, higher the fuzzy degree of the cluster.

If m=1, $u_{ij} \in \{0,1\}$;

If $m \to \infty$, $u_{ij} \to \frac{1}{k}$.

If there is a path between two pages, a relation to a task case exists for these two pages. The shortest path D*ij* between two pages P*i* and P*j* can be defined as following:

$$S_{ij}^{k} = \frac{1}{2^{L(p_i,p_i)}} + \frac{1}{2^{L(p_j,p_i)}} \qquad (4)$$

$$D_{ij} = \frac{1}{\sum_{k=1}^{n} S_{ij}^{k}}$$

where $L(P_i, P_j)$ is the least links number between P_i and P_j.
If there is not a path between P_i and P_j, $L(P_i, P_j)$ is infinite. Based on the above definition of distance, a Fuzzy Minimum Generation Tree method (Gao 2004) is applied to cluster web page to task cases. The clustering algorithm is as follows:

Step 1: Create a fuzzy graph G=(P,D), where P={P1,P2,...Pn} is a set of n web pages with fuzzy relation R=$[r_{ij}]$mxn, D=$[d_{ij}]$mxn is a relation matrix, D*ij* is calculated by Equation (4).

Step 2: Chose the maximum value of d_{ij}; put it in a set Tr; put Pi and Pj into a set Pw.

Step 3: Check all of the relationships between the pages in set Pw and the pages not in Pw. Find a relation with the maximum value d_{ik}; put d_{ik} into the set Tr; put the new node P_k into Pw.

Step 4: If set PW does not include all the web pages in set P, go back to step 3. If set Pw includes all the web pages, construct a Minimum Generation Tree Tmax based on the fuzzy graph G.

Step 5: Chose a threshold value α as the criterion to define the level of clustering. Disconnect all the page links in the tree if the relation value is less than α. A forest of the navigation trees is constructed. Every tree is a cluster of a user task case.

Web page positioning based on task cases

Once the task cases are estimated in the previous step, the web pages can be allocated into each task case. The minimum generation tree method is applied to this algorithm to allocate pages to task cases. The position of each page in the web structure is determined. The algorithm is described as follows.
A task case TC*i* can be represented in a set G*i* and an associated fuzzy-graph.

G*i*==(PTC*i*,E(P*i*,P*j*)).

where PTC*i* is the set which includes all the web pages of this task case: $PTC_i=(P_1, P_2, ..., P_m)$. The weight that represents the linking relationship between two pages P*i* and P*j* is E(P*i*,P*j*). E(P*i*,P*j*) is the distance E(P*i*,P*j*)=D*ij*. If E(P*i*,P*j*) is infinite, there is no path between these two pages. Figure 5 illustrates an example of the pages and their weights in a graph.

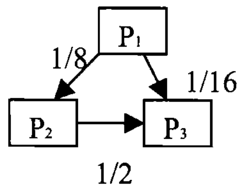

Figure 5: pages and weights

A minimum generation tree TR*i* is then built based on the fuzzy-graph. Pages are allocated into the tree. Figure 6 shows a result of this allocation.

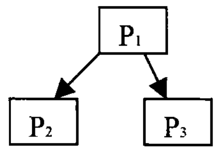

Figure 6: Minimum Generation Tree

Figure 7 gives an example of the fuzzy clustering using the Minimum Generation Tree.

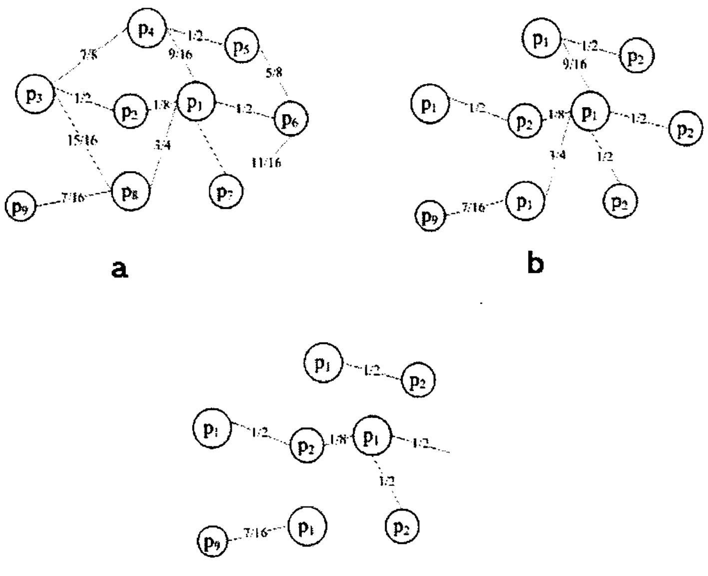

Figure 7: Web page clustering using Minimum Generation

DISCUSSIONS AND FURTHER RESEARCH

In the algorithm above, users' task cases are estimated by analyzing users' motivations. The position of each web page in navigation trees can then be determined. A Web structure is then built based on the navigation trees. We assume that a user's motivation meets an existing task case. If a user's motivation does not meet any of the existing task cases, a new task case will be generated.
The algorithm for generating new task cases will be studied in our next paper. We will apply a Vector Space Model to create a new cluster.

Future work will also involve an evaluation of the algorithm outlined in this paper using a prototype tool to automatically construct the web navigation structures. This will support verification of the extent to which the resultant navigation structure supports improved navigation.

REFERENCES

Lowe, D., & Kong, X. (2004) NavOptim Coding: Supporting Website Navigation Optimisation using Effort Minimisation. Paper presented at the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, September 20-24, 2004, Beijing, China. 2004

Isakowitz, T., Stohr, E. and Balasubramanian, P. (1995) RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM, 38* (8). 34-44. 1995

Schwabe, D. and Rossi, G. (1998)Developing Hypermedia Applications using OOHDM, Presented at Workshop on Hypermedia Development Processes, Methods and Models (Hypertext'98), Pittsburgh, USA, 1998.

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.

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.

De Troyer, O. and Leune, C. (1997) WSDM: A user-centered design method for Web sites. in *7th International World Wide Web Conference*, Elsevier, 85-94. Brisbane, Australia, 1997,

Ceri, S., Fraternali, P. and Bongio, A. (2000) Web Modeling Language (WebML): a modeling language for designing Web sites. in *Proceedings of WWW9 Conference*, p137-157, Amsterdam, 2000.

Takahashi. K. and Liang E. (1997) Analysis and Design of Web-based Information Systems, presented at 7th International World Wide Web Conference, Brisbane, Australia, 1997.

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*, p115-127, Salt Lake City, USA, 2000.

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.*p281–292. 1999.

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, 1999.

Chen. M.S., Park, J.S. and Yu, P.S. (1996) Data mining for path traversal patterns in a web environment. In *International Conference on Distributed Computing Systems*, 1996.

Spiliopoulou, M. and Faulstich. L.C. (1998) WUM: A Tool for Web Utilization Analysis. In *EDBT, Workshop WebDB'98, LNCS 1590*. Springer Verlag, March 1998.

Cooley, R. , Mobasher, B. and Srivastava, J. (1999) Data preparation for mining world wide web browsing pattern. *Knowledge and Information Systems*, 1(1), 1999.

Kosala R. and Blockeel. H. (2000) Web mining research: A survey. *SIGKDD Explorations*, 2(1), June 2000.

Masand B. and Spiliopoulou, M. editors.(2000) *Advances in Web Usage Mining and User Profiling: Proceedings of the WEBKDD'99 Workshop*. Number 1836 in LNAI. Springer Verlag, July 2000.

Punin, J., Krishnamoorthy, M. and Zaki, M. J. (2001) *Web Usage Mining: Languages and Algorithms*, to appear in Studies in Classification, Data Analysis, and Knowledge Organization, Springer-Verlag, 2001.

Andreas Thor, Erhard Rahm (2004) AWESOME - A Data Warehouse-based System for Adaptive Website Recommendations. VLDB 2004: 384-395

Chen. J.B. (2005) Study of knowledge discovery based on multi-vector, PhD thesis, Beijing Institute of Technology. 2005.

Gu. Y.J. (2005) Feature Extraction and Feature Selection in Automation Chinese Text Categorization, PhD thesis, Beijing Institute of Technology. 2005