NavOptim Coding: Supporting Website Navigation Optimisation using Effort Minimisation

David Lowe, Xiaoying Kong
University of Technology, Sydney, Australia
{david.lowe, xiaoying.kong}@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 navigation within the application can occur, and especially the extent to 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. Despite the fact that this implies that navigation design is inherently an optimisation problem, few optimisation techniques have been applied in this domain – with most design techniques being based on intuition, general heuristics, or experimental refinement.

In this paper we discuss this problem, and propose a navigation representation which can become the basis for optimisation techniques.

1. Introduction

The Web has become a key vehicle for accessing organisational applications. In many cases, these applications provide crucial business services to external stakeholders, and hence the quality of the interaction is vital to business success. Given this, effective design of the web application interface is crucial.

So what do we mean 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.

Most existing Web navigational design methods take an essentially informal approach to this optimisation, though often based on techniques such as general guidelines, design heuristics, and user feedback. Without any formalism built into the design process it is, however, impossible to ensure full optimisation.

In this paper we discuss the issue of navigational optimisation, and provide a representation to support this optimisation which has been inspired by entropy coding techniques (such as Huffman coding). These coding techniques typically look at the design of information codes which minimise the average code length (and hence the required communication bandwidth or storage capacity). Rather than minimising the weighted average code length, we aim to minimise the weighted navigational effort, i.e. as specific information (or functionality) within an application becomes more important, it should become easier it locate.

This means we need to understand (and have measures for) navigational effort, as well as understanding the importance to prospective users of various information and functionality.

In this paper we describe an approach to navigational optimisation of Web applications – which we refer to as NavOptim - based on modelling of both navigational effort and the significance of the information, and subsequent optimisation of these metrics across the overall application. In the next section we begin by discussing related work, including approaches to the design of Web navigation structures. In section 3 we describe the metrics we use for measuring navigational effort. We then discuss (in section 4) the application of this technique. Finally, in section 5, we discuss future directions for this research.

2. Background and 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 Web navigational design approaches.
2.1. Structured Design Approaches

Over the last decade numerous approaches have been developed for performing the design of the navigational structure of Web systems. Early approaches in this area 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 [1]) 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 [2]) 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 [3] and work by Lee [4].

WSDM [5] 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 [6]) 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 [7].

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 [8]. 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 [9, 10] 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.

2.2. 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 [11, 12]. 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 [13] the focus is on usage of the system, rather than users of the system — though again the optimisation is subjective.

2.3. Optimisation

In each of the above cases the approach is focussing 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 (we will discuss this assumption more shortly).

Much of our inspiration for this work has come from an analogy with Huffman coding [14]. In Huffman coding, we can algorithmically select variable length codes for a set of symbols which lead to an overall minimum length message constructed from these symbols. The selection of these codes is based on the probability of each symbol occurring. We want to develop a similar approach which minimises navigational effort (rather than code length), but which takes into account the significance of each set of information (rather than the probability of the symbol occurring).

This, of course, implies that we need to understand navigational effort. There have been various approaches proposed for measuring the effort required to navigate around a site. Most of this work has been based on either subjective evaluation or rudimentary metrics (e.g. [15]).
Whilst this existing work on navigational effort shows how it can be modelled and measured, it has not then extended to understanding how to optimise the navigational effort across an entire site. For example, [16] describes the application of Huffman coding to hypertext structures, but this is about comparison of these structures rather than their optimisation.

3. NavOptim Representation

As discussed above, in order to optimise navigational structures we need effective representations of navigational effort. In this section we look at how this effort can be represented.

In order to represent effort we need to understand which navigational paths are appropriate to consider (and to weight these accordingly). This path selection is supported through the use of usage-centred design techniques. Usage-centred design makes use of task cases, which describe user intentions and system responsibilities.

3.1 Navigational Architecture

User navigational events are consumed within the web navigational architecture to achieve the goal of a given task such as retrieving information or accessing an available service. A web navigational architecture can be represented as a graph of web pages and links. See Figure 1. A web page can contain a number of links to connect to other pages.

![Navigational architecture](image)

Figure 1: Navigational architecture

A user’s navigation is related to a path-choice of links and pages (governed by a task case) within the navigational architecture. The navigation effort for a user is the amount of navigation consumption starting from an entry page, then going through each chosen web pages with certain probabilities to a destination web page to complete her task case. The total navigation effort for a whole web system will be the total navigation efforts against all task cases of the system with all possible path-choices. These path-choices have certain probabilities and weighted significance.

The probabilities of the choice of links at each step of a navigational path are determined by the semantic cohesion of a user’s task case and the information of linked pages. To model the representation of the system navigation efforts, we need to understand these probabilities and weighted significance that affect the user navigation behaviour. Firstly, let’s quantify the semantic cohesion using a semantic vector space model.

3.2 Semantic Cohesion

Semantic cohesion is a way of defining the degree of consistency of a given set of information. In this case we can consider the cohesion of the information that lies along a navigation path associated with given task case. We use “SmtCoh” to represent the measurement of the semantic cohesion.

Each web page typically contains both diverse content and anchors for links to other pages. The anchors (and associated content) can be considered as attributes that define the relationships to other pages. Figure 2 is an example of these attributes for a web page in a university website that provides information and services for teaching and learning online.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute 1</td>
<td>Information of student profile short</td>
</tr>
<tr>
<td>Attribute 2</td>
<td>Information of subject short</td>
</tr>
<tr>
<td>Attribute 3</td>
<td>Timetable information</td>
</tr>
<tr>
<td>Attribute 4</td>
<td>Information on research profiles</td>
</tr>
</tbody>
</table>

Figure 2: attributes of a web page

Each task case and each attribute of a web page can be represented by a vector in a semantic vector space (see Figure 3). The direction of a vector represents its semantic direction. The angle between the task case vector and the attribute vector represents their quantified semantic difference [17]. The semantic cohesion is defined by the cosine of this angle. If a task case and an attribute match, we say, the angle between their semantic vectors is 0. Their semantic cohesion is 1. If there is no cohesion between the two semantic vectors, this angle is 90° (i.e. they are orthogonal) and the semantic cohesion is 0.
We use an example to demonstrate the quantification of semantic cohesion. Let’s set a task case “TaskCase1” as “a student searches for the classroom of a subject in which the student has enrolled”. A student user with “TaskCase1” navigates and arrives at a web page of Figure 2. The semantic vectors of TaskCase1, and Attribute 1~4 of this web page are presented as $V_{AttrCase1}$, $V_{Attribute1}$, $V_{Attribute2}$, $V_{Attribute3}$ and $V_{Attribute4}$ respectively in Figure 3.

![Figure 3: Semantic cohesion and semantic vector space](image)

![Figure 3: Semantic cohesion and semantic vector space](image)

The semantic differences between ($V_{Attribute1}$ and $V_{TaskCase1}$), ($V_{Attribute2}$ and $V_{TaskCase1}$), ($V_{Attribute3}$ and $V_{TaskCase1}$), ($V_{Attribute4}$ and $V_{TaskCase1}$) are $\alpha_1$, $\alpha_2$, $\alpha_3$ and $\alpha_4$, respectively. The semantic cohesions are:

- $SmtCoh(V_{Attribute1}, V_{TaskCase1}) = \cos(\alpha_1)$
- $SmtCoh(V_{Attribute2}, V_{TaskCase1}) = \cos(\alpha_2)$
- $SmtCoh(V_{Attribute3}, V_{TaskCase1}) = \cos(\alpha_3)$
- $SmtCoh(V_{Attribute4}, V_{TaskCase1}) = \cos(\alpha_4)$

(1)

We can see from Figure 3, “Attribute 4: information_on_research_profiles” is not semantically cohesive to “TaskCase1”. Therefore $\alpha_4 = 90^\circ$, and $SmtCoh(V_{Attribute4}, V_{TaskCase1}) = 0$.

“Attribute 3: timetable_information” is quite cohesive to “TaskCase1”. The semantic cohesion $SmtCoh(V_{Attribute3}, V_{TaskCase1})$ is defined as 0.89.

Semantic cohesion between attributes can be defined in the same way. Under a usage-centred approach, we only use the semantic cohesion between an attribute and a task case. A semantic cohesion matrix for the system could be established during the web system information design stage. This matrix contains the semantic cohesions between all attributes and task cases. Since one attribute in a web page could appear in other web pages, one attribute will be listed only once in the matrix. Attributes are categorized in information architecture design. Since each individual task case has different significance for the entire system (i.e. certain tasks are more likely than others to be of importance to users), a significance factor ($Sgnf$) is assigned to each task case. Figure 4 is an example of a semantic cohesion matrix.

![Figure 4: System Semantic Cohesion Matrix](image)

Once the semantic coherences are defined for all attributes and task cases, the probabilities of user’s navigational decision can be understood.

3.3 Probability of choice of path

To simplify the problem, firstly we use a single task case to calculate the probabilities of user’s navigation choice in a path. And we assume that the semantic cohesion matrix is defined.

Suppose in a web page $Pg_0$, there are $m$ links, each link is highlighted with an $Attribute_i$, $i=1...m$. See Figure 5. Each choice of link will direct the user to next web page $Pg_i$. Each decision of choice of link has a probability $pi$, $i=1...m$.

![Figure 5: Probability and semantic cohesion](image)

A user will make a navigation decision based on how an attribute is semantically related to user’s task case. When in a web page, the user has two types of options: (a) quit the current page backwards or changing URL or (b) goes to next linked page by operating the linking attributes.
In option type (a), the user has completed the task case, or none of the linked attributes is related to user’s task case, semantic cohesions between the attributes in the current page and the task case is 0, the user makes a decision to quit. Hence

\[ p_i = 0, \quad i = 1:m \]  

(2)

In option type (b), at least one of the attributes in the current page is related to the task case (a label TaskCase1 is used here to illustrate this navigation task case), that is

\[ \max(SmtCoh(V_{attribute_i}, V_{TaskCase})) > 0, \quad i = 1:m \]  

(3)

In this case, the user has \( m \) options to go to \( m \) linked pages via \( m \) attributes. The probability of her navigation choices to go from current page \( Pg0 \) to next page \( Pg_i \) is proportional to the semantic cohesion between \( Attribute_i \) to the task case \( TaskCase1 \). Then the following presents this relationship:

\[ p_i = k \cdot SmtCoh(V_{attribute_i}, V_{TaskCase1}), \quad i = 1:m \]  

(4)

where \( k \) is a constant for \( m \) such proportional relationships in current page. Then

\[ \sum_i p_i = 1 \]  

(5)

Consequently, the probability \( p_i \) can be represented by deriving Equation (4) and (5) as follows:

\[ p_i = \frac{SmtCoh(V_{attribute_i}, V_{TaskCase1})}{\sum_{i=1}^{m} SmtCoh(V_{attribute_i}, V_{TaskCase1})}, \quad i = 1:m \]  

(6)

Once the probabilities of navigation choice are calculated, the navigation architecture can be reconstructed graphically and quantifiably with web pages, links with attributes, and probabilities of choice of links (see Figure 6). The probabilities of choices are presented on the links to from pages.

If considering all task cases of the system, in the system navigation architecture, the array of probabilities to a page for all task cases could be established. The array can be drawn on the links to this page. An example of a probability array could be represented as \( [p_{j_TC1}, p_{j_TC2}, \ldots, p_{j_TCn}] \)

where \( p_{j_TC} \) is the probability for Task Case \( j \), \( n \) is the total number of task cases of the system.

### 3.4 Representation of Navigation effort

The measurement of user navigation effort is analogous to the measurement of data compression in information theory.

In the process of transmission data, a message “Msg” is transmitted from an information source to an information destination. The message “Msg” consists of \( M \) symbols \([a_1, a_2, \ldots, a_m]\). Each symbol \( a_i \) (i=1…m) has a frequency or a probability \( p_i \) (i=1…m) to appear in the message “Msg”. Each symbol is represented and transmitted by a number of bits. 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. This effort has been given using entropy of probability distribution as follows [18]:

\[ H = -\sum_{i=1}^{M} p_i \log p_i \]  

(7)

Compared with data compression theory, user navigation architecture has a similar structure and behaviour. A user navigation path is like a message. Each web page is like a symbol in a message. The probability of choosing of particular page during user navigation is analogous to the probability of the symbol in a message. Therefore the measurement of navigation effort is analogous to the measurement of total length of a message.

The representation of navigation effort has two cases in usage-centred approach: (a) navigation effort for single task case; (b) navigation effort for entire system with all task case with weighted significance.
Navigation effort of single task case

The navigation architecture for a single task case is illustrated in Figure 6. For each page, there are \( m \) choices of links. The probabilities of the choices of links for this page are \( p_1, p_2, \ldots, p_m \). Using entropy of probability distribution, the user navigation effort for a single task case \( \text{TaskCase}_j \) is represented as:

\[
H(j) = -\sum_{i=1}^{m} p_i \log p_i
\]

where \( j \) is the task case number.

Navigation effort of system

To measure the user navigation effort for the entire system, the significance of each task case is considered. In the graphical representation of the navigation architecture, the probability array for all task cases are presented on the links of pages as illustrated in Figure 7.

![Diagram of navigation architecture with all task cases and probability array](image)

Figure 7: Example of navigation architecture with all task cases and probability array

Suppose there are \( n \) task cases in the entire system as in Figure 4. For each task case \( \text{TaskCase}_j \ (j=1, \ldots, n) \), a significance factor \( \text{Sgnf}(j) \) has been assigned. The navigation effort for \( \text{TaskCase}_j \) is represented as \( H(j) \). The navigation effort for entire system \( H_{\text{sys}} \) is the sum of weighted efforts of individual task case. That is:

\[
H_{\text{sys}} = \sum_{j=1}^{n} [\text{Sgnf}(j) \times H(j)]
\]

\[
= -\sum_{j=1}^{n} [\text{Sgnf}(j) \times (\sum_{i=1}^{m} p_i \log p_i)]
\]

4. Discussion

So, we now have a metric which defines the navigational effort required to reach a given page, and a composite metric for an overall site. These metrics can be used in various ways. For example, we can define a set of task cases for a particular application, and then use these in evaluating different navigational designs by comparing the net system navigational effort for each design. Theoretically, we could identify the optimal structure by appropriate analysis of the effort metric (this is, however, the basis for ongoing work is not addressed in this paper).

By analysing the metrics that we have proposed, there are a number of additional observations which we can make regarding this approach – and the consequences for designing optimal navigational structures. The first is the invalidity of the assumption that less relevant content (or less frequently accessed content) should be buried more deeply into a navigational structure. This is because this content, whilst less relevant itself, may be tightly coupled with some other content which is very relevant. Separating them in the navigational structure will lead to a problem with cohesion, and hence adversely affect the navigational effort.

Another example conclusion which can be drawn from our mathematical models is that very flat broad navigational structures and very narrow deep structures are both sub-optimal. The former is problematic insofar as at each navigational step we would need to make very complex choices, which the latter is problematic in that whilst the navigational choices are simple, there will be many of them. An analysis of the model would allow the identification of an optimal hierarchy depth for any given hypertext size.

Further insights will be gained by future analysis of these metrics and their application in analysing navigational structures.
5. Conclusions and Further Work

In this paper we have discussed an approach to the evaluation and improvement of navigational structures based on the optimisation of a navigational effort metric weighted by the page 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 focussing on simulation of different websites to see how they are optimised using our approach and then subjective evaluation to determine whether this does indeed lead to qualitative improvements. We will also be undertaking case studies of optimisation of existing sites. Finally, we have begin considering the development of a tool to support automated analysis of websites to identify potential areas of improvement in terms of their navigational structures.

6. References


