Multi-Level Evaluation of Web Site Navigability

Stephane Vaucher, Houari Sahraoui
Département d’informatique et de recherche opérationnelle
Université de Montréal
{vauchers, sahraoui}@iro.umontreal.ca

Abstract—Much research in recent years has focused on the evaluation of web site quality. The majority of this research has focused on evaluating the quality of individual pages or that of a site as a whole. In this paper, we propose an evaluation approach that combines evaluations at the page level with the one of the web site by means of a page-importance weighing model. We illustrate our approach with the particular characteristic of navigability. Both navigability models at the page and site levels are implemented as Bayesian Belief Networks (BBNs) to manage explicitly the uncertainty of web site evaluation. The page importance weighting is implemented using a random walk strategy. We evaluated the resulting model using a mean-different hypothesis testing. The results showed that our model allows discriminating correctly between web sites having a good navigability and randomly-selected web sites.

I. INTRODUCTION

Much research in recent years has focused on web quality. Web site quality in particular is a major concern because it influences heavily the business performance on many companies [1]. An important aspect of web site quality is the usability [2], [3]. Two main reasons explain this importance. First, the user interactivity with the applications is very high compared to the one of traditional software. Second, the number of potential users is very large and their profiles can be difficult to characterise. Traditionally, usability assessment is done using a (formal or informal) survey of users who evaluate how easy it is to perform specified tasks[3]. Even though these are the most complete ways to evaluate a system, they are also very expensive and cannot be fully integrated into a continuous development process.

The problem we address in this paper is how to integrate user assessment without the prohibitive costs of actually surveying users. Quality models are well suited for this task as they simulate the opinion of a user. These models can be based on the results of a literature review or even built and calibrated using data from surveys. Their execution should be automated [2] so that developers can run them and get feedback quickly much like the use of unit tests within an agile development process.

In this context, we propose an approach for assessing web site quality characteristics related to usability, and illustrate it on the particular case of navigability [4], the ease with which users can locate and access relevant information. This is a key concern to users of a web site who need to find the page containing the information they seek either using links or by means of other mechanisms like search engines or site maps. The majority of existing research has focused on evaluating the quality of individual pages (e.g. [5]) or that of a site as a whole (e.g. [6]). We take a different approach and focus on how to combine the evaluation of individual pages to assess the navigability of a whole site.

This research is based on a model that assesses the navigability of a web site using Bayesian Belief Networks (BBNs). This model combines metrics extracted from the site with an evaluation of individual web pages whose importance is weighted by a navigation model. There are two advantages to using BBNs. First, BBNs can model the uncertainty inherent to decision-making by using Bayes’ theorem. Second, the model can guide improvements because the evaluation process can be used backwards: for a desired output, it can identify the required inputs.

We tested the usefulness of the quality model approach in a preliminary study where two groups of sites were evaluated and compared : sites known to have good navigability and randomly selected sites. The good sites are either Webby1 nominees or winners. The results show that the navigability of the good sites was correctly assessed as being statistically better than randomly selected sites.

The rest of the paper is structured as follows. Section ?? describes the problem of assessing the navigability of web sites. Section ?? provides an overview of related work in the field of navigability assessment. In Section ??, we present our approach which is empirically evaluated in Section ??.

II. PROBLEM STATEMENT

The navigability of a web site is a measure of how easily a user can locate and access the information he needs [2]. For this task, he typically has two options. He can either explore the site, going from page to page by following links or, if available for the web site, he can use other features like a search engine to access pages directly. Both navigation types need to be considered when building a model.

For the assessment of navigability, a web site can be viewed as a directed graph; formally, $G = \langle V, E \rangle$ where $V$ and $E$ are respectively the set of vertices representing the pages and the set of directed edges representing links between pages. An edge $(u, v)$ represents a link in the page $u$ to the page $v$. Vertex $u$, is called the head of the link and $v$, the tail. For vertex $u$, the out-links is the set of links with $u$ as the head, representing the links to the other pages, and in-links is the set

1http://www.webbyawards.com
with $u$ as the tail, representing the links to $u$ from the other pages.

A user requiring information located at page $p_{dest}$ needs to find a path $\langle p_1, p_2, \ldots, p_{dest} \rangle$ in $G$ that takes him from his origin $p_1$ to his destination $p_{dest}$. In terms of the graph, this is a greedy path-finding problem where at any given page a user needs to figure out which out-link leads him closer to his destination. A site with good navigability should ensure that few steps are required to reach any destination. Some potential navigation difficulties arise due to pages with inadequate link identification (e.g. bad anchor text) or to pages that overwhelming him with too much information (e.g. the user needs to scroll down to find the correct link).

Another option available to a user is to use a search engine. By using the search engine, the user jumps directly to a new page. Exploration is useful even if the site has been indexed by a search engine because, lacking the correct keywords, a user may not find the page he needs. The two methods of navigation are thus complementary.

Consequently, the general navigability of a web site needs to take into consideration both ways to navigate the site. First, it needs to evaluate the impact of visiting every page on the navigability of the web site. This means that the model should evaluate how easy it is to find the appropriate link to follow on every page and combine this to the probability that a user will be on that page. Thus two models are involved: a page quality model and a navigation model. In addition, our method must take into account the alternate navigation mechanisms that are provided by the web site.

III. RELATED WORK

The automation of usability evaluation of user interfaces is gaining popularity. Many approaches are proposed to reduce the cost of this evaluation (see a summary of some techniques in [?]). In the specific case of web sites, the level of automation is however relatively low. Web usability has been addressed extensively, and its evaluation generally takes the form of principles, guidelines, questionnaires, and inspection procedures. We only give in this section an example for each of the above-mentioned techniques. Nielsen and Loranger address Web usability issues by giving advice and guidelines [?]. Kirakowski et al. developed the WAMMI questionnaire, based on user satisfaction. It provides developers with a list of elements to diagnose usability problems and improve Web sites quality [?]. Finally, Frøkjær and Hornbæk developed a usability inspection technique, called MOT, that uses the metaphors of human thinking [?]. An empirical validation showed that this kind of inspection, although manual, finds more problems that classical inspection techniques.

Much work considered explicitly the navigability of web applications. Zhang et al. [?] propose complexity metrics to evaluate the navigability. Newman and Landay considered it as one of three aspects affecting the quality of the interface design of Web applications [?]. Olsina et al.[?], decompose quality hierarchically and navigability is a factor affecting the suitability quality sub-characteristics. Zhou et al. [?] proposed a navigation model that abstracts the user Web surfing behavior as a Markov model. This model is used to quantify the navigability. Finally, Cachero et al. [?] used a model-driven approach to define a model for the measurement of navigability and a process for evolving this model.

Several methodologies have been proposed to assess multiple quality characteristics of web applications. For instance, Olsina et al. [?] define WebQEM (Web Quality Evaluation Methodology). Albuquerque et al. [?] suggest FMSQE (Fuzzy Model for Software Quality Evaluation) model. The model uses fuzzy logic and presents a quality tree for e-commerce applications. It takes into account problems related to uncertainty during quality evaluation. Shubert et al. [?] develop EWAM (Extended Web Assessment Method). The method is based on Fishbein’s behavioral model and Davis’ technology acceptance model. It is applied to e-commerce web sites and is supported by a tool. Recently, Mavromoustakos et al. [?] use importance-based criteria for evaluating requirements in their quality model WAQE (Web Application Quality Evaluation model). Regarding the use of probabilistic approaches for quality assessment, Malak et al. [?] propose a method for building web application quality models using Bayesian networks. The approach of Malek et al. was used by Caro et al. [?] for the particular case of web portal data quality.

IV. ASSESSING SITE NAVIGABILITY

In this section, we present the model used to assess navigability. As shown in Figure ??, our model is composed of three sub-models allowing to determine respectively three kinds of decisions: the navigability at individual pages (white boxes), the importance of each page in the site to weigh the contribution of its navigability (light-gray box), and the navigability at the site level (dark-gray box).

Both navigability sub-models (page and site levels) are implemented as Bayesian Belief Networks. Before giving the details about our model components, we justify the choice of using such probabilistic models.

A. Probabilistic Quality Models

Evaluating navigability is a highly uncertain process. There are different types of users, and depending on their experience, their opinions on how a site should be organised differ. Thus consensus is very unlikely and a probabilistic approach, such as Bayesian Belief Networks (BBNs), is warranted. BBNs have been successfully used in fields as diverse as risk management, medicine, and computer science. We therefore believe that their use is appropriate for determining the certainty of web site usability.

Probabilistic modeling is based on Bayes’ conditional probability theorem which combines the inherent probability of an output ($A$) with its dependence on inputs ($B$) as well as the probability of ($B$) occurring. This is expressed by the following equation:

$$P(A|B)=\frac{P(B|A) \cdot P(A)}{P(B)}$$ (1)
A problem with this type of modelling is that to consider the effect of \(n\) possible binary inputs on an output, we would need to consider \(2^n\) combinations. BBNs use a graph structure to organise conditional dependencies and limit the size of the problem making the problem tractable and simpler to reason with.

A BBN is a directed, acyclic graph that organises the dependencies of a set of random variables \(X\). Every vertex of the graph corresponds to a variable and every edge connecting two vertices indicates a probabilistic dependency from the head, called parent, to the tail, called child. The random variables represented in the graph are only conditionally dependent on their parents. The joint distribution of \(X\) can be written as:

\[
p(x) = \prod_{v \in V} p(x_v | x_{\text{parent}(v)})
\]  

When building a BBN, each vertex should correspond to either a concept that is observable (and measurable) or to a decision point given inputs defined by parents. The edges should be used to represent causal relations between vertices and allow a developer to interpret the results of an evaluation (e.g. the output is caused by this input). This structuring can either be done automatically using heuristics found in the literature or manually to correspond to a specific decision process. Since there is abundant information on how to evaluate the quality of web sites, we chose the latter.

The structure only determines the dependencies between variables, the exact joint distribution needs to be defined. With discrete variables, each variable \(X_i\) is configured with a conditional probability table (CPT) that establishes its probability distribution given the values of its parent nodes. These tables can be learned or entered by an expert.

### B. Assessing a Web Page

To assess the navigability of a web page, we adapted a BBN from [7]. In their study, they presented and validated a model that evaluates the navigability of a page by considering both page and site-level mechanisms. This model was obtained using a GQM approach [7] to refine navigability characteristics collected from ten sources (model proposals, standards, guidelines, etc.). In our approach, as we separate page and site influences, navigation mechanisms are moved to the site-level model. Figure 2 presents this modified BBN.

The decision of whether or not a page is easily navigable (navigability node) directly depends on three sub-characteristics: the ability of a user to identify the correct link to follow (UserFeedback), his access to available navigation mechanisms (Bind), and the size of a downloaded page (PageSize). Both the UserFeedback and the Bind nodes are intermediate decision nodes which depends on other sub-characteristics that are then decomposed into metrics. The inputs of the model are described in Table 2.

| TABLE I | INPUTS TO THE NAVIGABILITY MODEL |
|---|---|---|---|
| Page Size | PageSize | measure (count) |
| Ratio of links with titles (LTitle) | LinkTitle | measure ([0,1]) |
| Ratio of links with text (LTxt) | LinkText | measure ([0,1]) |
| Significance of page URL (URL) | URL | binary |
| Indication of location in web site (CPL) | CurrentPositionLabel | binary |
| Visited links change color (VLC) | VisitedLinkColor | binary |
| Breadcrumbs (BRC) | PathMechanism | binary |
| Number of links in page (NoL) | LinkNumber | measure (count) |
| Link to home (Home) | LinkToHome | binary |
| Support for Back Button (BB) | BackButton | binary |

**a) Input Metrics:** In our model, every input needs to be converted to a random discrete variable. Binary metrics with two values: \(\{T, F\}\) are handled directly; but numeric metrics must be converted to a set of discrete ordinal values, e.g., low, medium, high. These values should reflect what a user might answer in a survey. For example, he might state that, for a particular page, there are breadcrumbs, \(P(\text{Breadcrumbs} = T) = 1\) and that there are many links \(P(\text{LinkNumber} = \text{High}) = 1\). However, it is not only unfeasible to maintain a group of users on hand whenever a developer wants to evaluate his site, it is unlikely a user will be able evaluate every page in a large site. Consequently, this process needs to be automated as well.

From the automation perspective, binary metrics fall into two categories. For the first one, it is possible to determine automatically if the value of the metric is true or false. For example, by analysing automatically the source, one can decide
whether the visited links change color ($\text{VisitedLinkColor} = T$).

The second category of metrics are those that require visual interpretation of displayed elements. This is the case of Breadcrumbs. Indeed, to detect automatically their presence, we need to rely on heuristics. Our rule rely on two symptoms: the use name and identifiers of divisions (div elements) and the names of css (presentation) styles. If either element type contains terms like “breadcrumbs”, then it is likely that the page contains breadcrumbs. We encode the interpretation process as a classification problem for which we use BBNs where, given a set of symptoms $S$, we calculate the probability that the metric belongs to a class $T$ or $F$. An example of a BBN implementing the rule for breadcrumbs is given in Figure ??.

To transform the numeric metrics into probability distributions, we calculate the probability that the metric would be classified, for example, as “low”, “medium” or “high” given a metric value. The exact number of classes depends on the attribute. The process followed is described in [?]. First, we extracted metrics from a set of randomly downloaded pages (over 1000). Second, for every metric, we derived the classes’ membership functions using fuzzy clustering. We used fuzzy kMeans clustering to find $k$ centroids corresponding to the $k$ classes (three in the case of low, medium, high). Finally, the probability of a metric is calculated on its relative distance to the value with its surrounding centroids. This process is shown in Figure ??.

$\text{Fig. 2. The page-level navigability model}$

$\text{Fig. 3. Binary input classification}$

$\text{Fig. 4. Converting metrics to probabilities}$

b) Executing the Model.: The navigability of a page is assessed by evaluating the probability that the navigability node is true ($P(\text{Navigability} = T | \text{inputs})$) given the BBN structure and parameters (CPTs). Navigability only depends on its three parents nodes, two of which need to be recursively evaluated until an input node is reached. The value of input nodes need to be computed on the evaluated web page. The precise CPT is shown in Table ??.

The probability of having a good navigability ($\text{Nav}$) would be calculated by evaluating the effect of all possible values ($d$) of the parent nodes (shortened to BD, SZ and FB) on Nav.

$P(\text{Nav}=T) = \sum_{BD,SZ,FB\in\{d\}} P(\text{Nav}|BD,SZ,FB)P(BD,SZ,FB)$

(3)

C. Navigation Model

The role of the navigation model is to evaluate the importance of each page, i.e., the probability that a user will transit by that page to reach the desired page. This information is used to weight the navigability scores obtained by the page.
navigability model in order to produce a unique input to the web site navigability model.

Many existing algorithms decide on the importance of a page given the topology of the site. One of the best known algorithms is PageRank [7]. It determines the probability that a user randomly clicking links will reach a given page. This particular algorithm discriminates against pages with large numbers of outgoing links and few incoming links. Yet, with regards to navigation, these pages are relatively important since they provide a way for users to reach many possible destinations.

Our algorithm to compute importance is also based on random walks. A user starts from the home page. Then, for every given page \( p \), it assumes that a user will follow a randomly chosen link with a uniform probability \( 1/outlinks(p) \). The precise algorithm used is based on a breadth-first search and is presented in Algorithm 2.

**Algorithm 1:** Visit probability

```plaintext
Inputs : home: the start page
Inputs : outlinks: a vector of outlink for a page
Outputs: weight: a vector describing the relative weight of a page

Q = empty – queue
mark[home] = visited
weight[home] = 1
Clicks = 0
enqueue home into Q
repeat
    dequeue page from Q
    foreach Link(page, v) \in outlinks(page) do
        if mark[v] \neq visited then
            mark[v] = visited
            enqueue v into Q
        end
    Clinks = Clinks + 1
    until Q is not empty;
    foreach v \in weight do
        v \leftarrow v / Clinks
    end
return Visits
```

The way of modeling navigation lends more weight to pages near the home page, especially in shallow sites. This is expected because we consider that most users need to transit through these pages to reach their destination. As with the breadth-first search, the algorithm assumes that a user will only visit a page at most once. This is understandable since a user that is trying to locate information who returns to a previously visited page, will likely go back and try another link instead of continuing on. The number of total possible clicks is used to calculate the relative weight of a page between \([0, 1]\).

The contribution of a page to the navigability of the site is determined by the function \( \text{weighted importance} \) (Equation ??) where \( \text{prob}(page) \) is the probability that a user will visit \( page \) and \( \text{nav}(page) \) is its navigability score as judged by the page-level navigability model.

\[
\text{weighted importance}(page) = \text{prob(page)} \times \text{nav}(page) \tag{4}
\]

The total page navigation score \( \text{total page nav} \) of a site is calculated as the average \( \text{weighted importance} \) considering all the pages of the site (Equation ??). As expected, when all of the pages are of the same quality \( q \), this function also returns \( q \), no matter the contribution of the navigation model. Furthermore, a common situation is when there are many pages at a same depth that follow the same template. This is sometimes due to dynamic generation of pages. In this case, the influence of the template will be relative to the number of pages generated.

\[
\text{total page nav}(site) = \frac{\sum \text{page weighted importance}(page)}{|site|} \tag{5}
\]

D. Assessing a Web Site

The navigability of a site depends both on the ability of a user to navigate pages to find the desired page and the presence of site-level mechanisms. Three such mechanisms are the presence of a menu, a site map and a search engine. All three mechanisms allows for a quick moves around different parts of the site.

A standard navigation menu in a site allows users to quickly switch to different parts of the site. Even though the menu provides links which are taken into account when calculating the navigation path, semantically, a menu has a specific meaning for users which improves a site’s navigability.

The same can be said of a site index which links to many other pages, but essentially provides a sense to how the site is organised and how to navigate. Finally, a search engine allows access to parts of the site that might not be explicitly linked.

![Site-level quality model](image)

Fig. 5. Site-level quality model

The proposed model is illustrated in Figure ?? and combines the concept of navigation mechanisms (a subgraph) and page navigability (the other subgraph). For the subgraph NavigationMechanisms, we reused the CPT given at the page level.
of the work in [?]. The input PageNavigability takes the probabilities obtained from Equation ??.

V. Case Study

To be able to include quality models in a development process, we must first show that it is able to simulate the judgement of a user. In this section, we present a study that compares the evaluation of our model of web sites with the expectations of users.

A. Study Setup

Two groups of web sites were analysed: a group of “good” sites and a group of randomly sampled sites. The good sites are either Webby award winners or nominees. In the judging process, different experts evaluate the sites according to six criteria including their navigability. The randomly selected sites are pages linked by http://www.randomwebsite.com; they include both personal sites as well as large businesses. In this study, we test whether or not the model find a significant difference in navigability between the “good” sites and the random ones. To this end, we perform a mean-difference test. t-test is used if the data is normally distributed. If not, a Mann-Whitney test is used instead. Normality distribution is checked using a Kolmogorov-Smirnov test. The tested groups contain respectively 9 good sites and 14 randomly selected sites.

To download and evaluate the sites, we built a web crawler based on htmlunit, a web testing library. HtmlUnit2 supports JavaScript heavy pages which allows our crawler to treat the majority of sites. We also used this library to extract metrics.

Two quality models were built and executed using BNJ3, a library for probabilistic reasoning.

B. Navigability Evaluation Results

Figure ?? shows the box plots corresponding to the evaluation for the two groups of sites. The good sites on the average present much better navigability than the random sites (0.74 compared to 0.51). The scores are normally distributed as indicated by a Kolmogorov-Smirnov test. Using a t-test (Table ??), we found that the difference is statistically significant with a p-value of 0.00. We can then reject the null hypothesis that there is not difference between the two groups in terms of navigability as evaluated by our model. We alternatively confirm that the navigability model is able to correctly discriminate between web sites.

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>INDEPENDENT SAMPLES TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>df</td>
</tr>
<tr>
<td>4.871</td>
<td>11.328</td>
</tr>
</tbody>
</table>

http://htmlunit.sourceforge.net

2http://bnj.sourceforge.net

C. Discussion

In addition to confirm statistically that our model is able to discriminate between web sites in terms of navigability, our results allows us to look closely to two additional points of interest: (1) is it necessary to evaluate exhaustively all the pages of a site?, and (2) how could we use the navigability model to improve a web site?.

The first point of interest is that we did not see any significant different score in navigability between analysing a complete site (a thousands of pages) and a portion of the site (a hundred of pages). These are good news since sampling pages from a large site is sufficient for its evaluation. However, we found that this sampling should include different parts of the site (possibly generated by different templates) since page quality tends to vary.

Another important point that is worth to discuss is how to use our model to improve navigability. This is important because, a recurrent concern when collaborating with industry is the relationship between quality evaluation and quality improvement [?]. Figure ?? shows for example, the result of the execution of our model on a good site. The navigability of this site is considered as good with a probability of 77.5%. The developers/managers could consider that this probability is not high enough and that they could improve the site to increase its navigability. Given the fact that there is a site menu (NavigationElements) and a search engine,
there are two possible ways to improve the site: improve the navigability of the pages, and/or add a site map. Already the site-wide mechanisms are judged to be good with \( P(NavigationMechanisms = \text{good}) = 85\% \), but the page navigability is poor: \( P(Pages = \text{good}) = 51\% \).

An important property of BBNs is that we can set the probabilities of any node (including the Navigability node) and that the other probabilities are updated accordingly. Let’s set a managerial objective of increasing general navigability to a level corresponding to a probability of 85%. A manager could set the output node’s value to 85% and try two configurations: keep the site without a map \( (P(SiteMap = \text{Yes}) = 0) \) and add a site map \( (P(SiteMap = \text{Yes}) = 1) \). With a site map, page-level navigability needs to increase to 54% (Figure ??) to reach the managerial objective of 85% (almost equal to the current value of 51%). Without the site map, page-level navigability needs to be increased to 72% (Figure ??). The development team could then estimate the cost of adding the map and the cost of modifying the pages and find the cheapest solution. For the second solution, as we know the individual navigability of the pages as well as their respective importance, some pages could be targeted. For a particular page, we repeat the output-probability setting to determine the improvement option at the page level.

VI. CONCLUSION

Prior work has proposed models for evaluating different characteristics of web sites. However, in general, these models target the page or the site level and do not provide explicit mechanisms to integrate both page and site evaluations. In this paper, we propose an approach that combines the evaluation on both levels using a third model that weight the contribution of the evaluations of the pages to one of the site. This multi-level evaluation is illustrated on the particular case of navigability.

Both page and site evaluations are performed using Bayesian belief networks. In addition to allow managing explicitly the uncertainty inherent to quality evaluation, these techniques provide a mechanism to learn the models’ parameters (probabilities). Moreover, we showed how these models could be used to guide the quality improvement.

To evaluate the proposed model, we conducted a study on a sample of real web sites. Our results show that our multi-level navigability model is able to correctly discriminate between sites having good navigability and randomly-selected sites. These encouraging results should be confirmed by a larger study similar to the one described in [?]. In that study, a large sample of sites was evaluated both by some subjects and by a navigability model. Both scores were compared and found to be significantly correlated.

In addition to a large-scale validation, our future work includes proposing a recommendation approach to improve web site quality. In [?], we proposed a method to recommend improvements to a web page on the basis of a quality model. Given a model, a set of possible transformations and an estimate of available resources, our method proposes an optimised sequence of transformations to apply to a page. Considering the problem complexity, meta-heuristics are used to find this sequence of transformation. We project to adapt this approach for the site level.

VII. ACKNOWLEDGMENT

This work has been partly funded by NSERC. We would also like to thanks Jean Vaucher for rereading the article.