

ANALYSIS OF USER SATISFACTION WITH THE ORGANIZATION OF CONTENT

Vangel V. Ajanovski
Institute of Informatics,
Faculty of Natural Science and Mathematics,
Saints Cyril and Methodius University
Skopje, Macedonia

ABSTRACT

This article gives an analysis how users click-streams in a web directory represented as a general categorization tree can be assessed in order to make deductions on the goals the user had when browsing and the choices the user made while trying to get to the needed information. The click-stream analysis is then used to propose an algorithm for restructuring of the categorization tree in order to optimize the structure for better navigation and easier finding of content. Based on several real-world metaphors of roaming in spaces and choosing paths, the algorithm that was based on a single user behaviour, is scaled to a general case. Finally, the article discusses how the access logs of a web site can be used to gather info on usage patterns of the organization structure itself and how it can be improved with the help of the restructuring algorithm and the assessment of several criteria named together - user satisfaction with the organization of content.

INTRODUCTION: TO SEARCH OR NOT TO SEARCH

The introduction to this article starts with a simple scenario: a computer user visits a knowledge base and tries to find the web page with the requested content. When the appropriate content is found he marks it as such.

The main question to ask is how to get to the content, to search for it or navigate towards it?

Presumably the content is well categorized and organized under easily understandable headings. This usually means that a n-ary tree was used for the categorization of content and that the user will be presented this tree for navigation.

So, the number of mouse clicks required to find the requested content in a set of 1000 pages will be:

- 10 clicks at most when a binary tree is used
- 6 clicks in a ternary tree
- even less for larger breadth of the tree

In the case of performing a search through the web's search engine, the user will typically need at least:

- 2-3 keywords (10-15 keyboard clicks on average)
- 3 mouse clicks

Even if in consideration are taken the largest indexes of web pages available today, which are in the rank of tens of billion pages (see [1]) and are reported to include the whole of the world wide web, the comparison still stands ground:

- 35 clicks in a binary tree
- 22 clicks in ternary tree ...

It is obvious that larger breadth of the navigation tree will lead to lesser number of mouse clicks. On the other hand it also means that there will be more options to choose at each

navigation step, which possibly increases the overall cognitive load required to perform the navigation to the final content page.

Mobile users have even more benefit from the directory scenario, especially when one takes into account the small screen and inability to browse through a dozen of search result pages with linked documents.

These comparisons only take into account an ideal scenario where in the case of the search engine the first or second search result page will show the requested link, and that in the case of the directory scenario there is no confusion in the choice at each level of the navigation tree.

Problems occur when the user is unable to find the requested info. Usually in a navigation tree, the user starts to browse back and forth the levels and chooses other paths, entering other branches of the tree. When using a search engine, the user browses through even more pages of the result set and then introduces new or different set of keywords in the search – thus starting from scratch.

ANALYSIS OF A DIRECTORY BROWSING SCENARIO

Figure 1 shows a navigational tree structure of a knowledge-base or a directory of web-pages organized in a hierarchical manner. This is the intended user perception of the organization of the directory. Internally there can be another formal structure, but the user is usually presented a navigational interface using a hierarchical category tree.

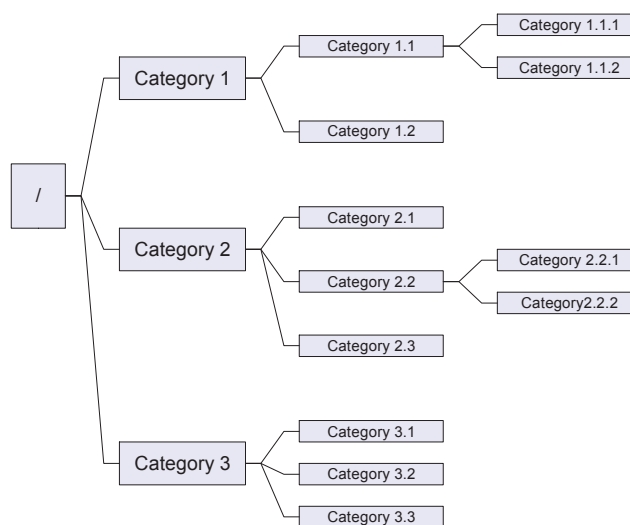


Figure 1: Perceived navigational tree structure.

The following scenario is observed: a visitor of the website is trying to find the right category with the required content and

follows a path that seems most obvious at the moment, as shown with the dark-grey dashed arrow in Fig. 2.

When the visitor finds the requested content, he should mark it as appropriate via a special check mark. This leaf of the navigation tree will be called *Success-category*.

If the visitor doesn't find satisfactory content in this path or gets to a leaf category in the path, he will go back and try another path until satisfied, the last category (or list of categories) being viewed before going back will be called an *Error-category*.

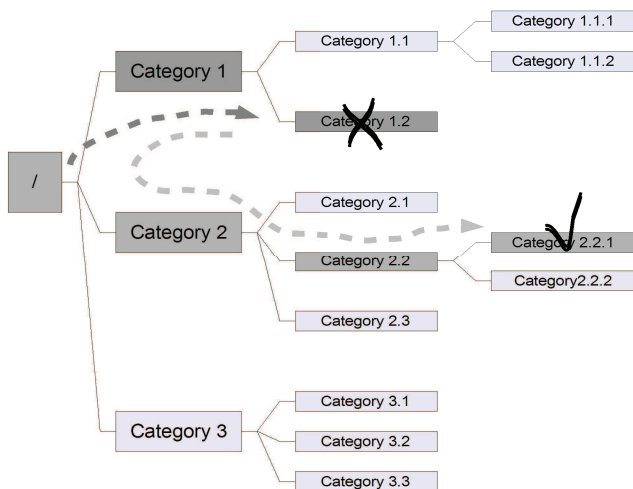


Figure 2: Visitor goes back and tries another path til satisfied

If no marker was checked during the whole user session it can not be assumed that the content was not found, neither that the user was satisfied with the last presented content, because the user might have stopped browsing, left the computer, or simply went directly to another website without following any links.

In order to distinguish between these cases, all outgoing links must be proxied and, if possible, inactivity timers must be employed per page.

INTERPRETATION OF THE USER CLICK-STREAM

When the user marked Category 1.2 as inappropriate (by simply going back one level higher in the navigation tree), it became obvious that he did not find the requested information.

It can also be concluded that the user simply disregarded Category 1.1 as not relevant. Then, because of the lack of other appropriate choices the user went back to the beginning.

In the second try the user is in Category 2. In this path, only Category 2.2 and Category 2.2.1 are rendered as good because the user found the needed page in Category 2.2.1.

Everything else should be dimmed as inappropriate. Thus, the whole branch Category 1 can be marked as inappropriate.

The fact that after all the browsing the user managed to find the required information, suggests that he was misled with all the titles (and descriptions) of all the categories he browsed into with no success.

Such examples are Category 1 and subsequently Category 1.2.

Category 1 could still be a valid option, because after reading its description the user went into it, but not for the specific information the user was searching for. It can only remain valid for the users search if it included a short-cut link to Category 2.

SIMPLE RESTRUCTURING ALGORITHM

The idea behind this algorithm is to move sub-trees of the navigation tree to the place where the user expected them at first, instead of their original place. To support this algorithm the navigation tree content – category titles and descriptions, must be written in such a way to be able to merge and split categories without the possibility for them to be misunderstood.

One way it can be done is if the leaf category descriptions will only include tag words or even brief sentences. The higher level categories can glue the tag words or sentences together as an overall category title or description.

The merger of a set of categories will make a union of all the tag words and sentences of its subcategories, and if a subcategory is moved out of a category, than naturally its tag words will be subtracted from the set of the master category.

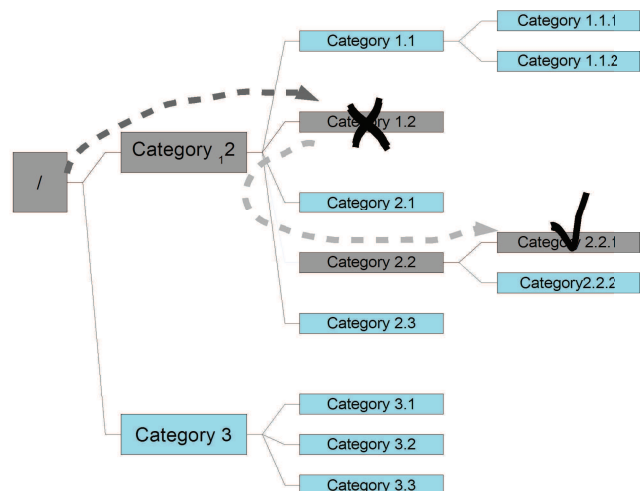


Figure 3: Merged category 1² results in less clicks

What if no results are found. How can it be detected?

- The user never submits a satisfaction report and
- the user manages to go through all the pages and does not find the content (never happens on large sites)

So, we must not say that the resulting page does not exist. We can only conclude that the user was not able to get to it, and this would mostly happen due to a miss-categorization.

In such case it is possible to discard the first level(s) of categorization and reorganize the whole navigation tree - merge the second level categories with smallest depth of dissatisfaction and move the remaining second level categories to the first level.

USER SATISFACTION WITH CONTENT ORGANIZATION

The described algorithm and structure is a plain case when only one user is visiting the website. The question rises when to perform the restructuring and in what places of the tree,

since one user's increase in satisfaction could mean a decrease in satisfaction of a whole group of other users.

Based on the described user browsing session, and the algorithm for a single user, a dynamic navigation tree restructuring algorithm can be envisioned which will take into account click-streams of all the visitors of the directory, and try to improve the browsing process in specific or even general scenarios.

The motif of the algorithm draws from an analogy with park or forest trails where it is sometimes obvious that some people always follow pre-made paths and some people tend to search for short-cuts. After many have taken similar short-cuts through the grass, new more natural and more functional paths occur. The tress-passers individually are not able to see the change they make, but all of them put together do make a difference.

This paper proposes a definition of a measurable parameter that will be used to assess if the current organization is better than the situation before the reorganization. This parameter will be named *user satisfaction with the organization of content*. This measure will be used in the implementation of the algorithm to achieve a better organization for a group of users, not just a single user.

The user satisfaction can be measured by several criteria:

- Number of clicks per user click stream
- Tree depth of the success-category
- Cumulative depth of the error-categories and the success-category
- Number of choices per screen
- Number of errors-categories
- Cases when content was not found at all
- Some authors even define a measure called navigational effort [2], which can also be used to calculate the difference among two different navigation structures.

Depending on the chosen weight of these criteria, an assessment and customization of the layout of the navigation structure can be made, in order to give satisfaction to a specific user group. For example, if it is assumed that business users have many tasks in their mind, the cognitive load should be lessened and thus the number of choices per screen should be smaller, so this criterion will have more weight than the others.

Once the weight of the criteria is set, the calculation of the new structure can be performed. This calculation should happen per category sub-tree, so each category sub-tree can be evaluated. If the satisfaction measure for some sub-trees is significantly less than the general average in the tree, those sub-trees should be reorganized.

This assessment can be done in breadth-first order until we pin-point the problematic sub-categories. After that, in each one of them, the restructuring algorithm can be performed based on the path traversal statistics gathered from site access logs.

The measurement is always based on the records in the web access logs, where depending on the logging scheme that is used, several data can be found (or recorded if its a custom build application):

- access time and page that is accessed

- previous page that forwarded to the current page
- user session

With these data, click-streams can be computed and attributed to paths in the categorization tree (or graph if it is so) and to user sessions, so it can be observed what a specific user was doing. After that the measurement can begin and assess on every user session if it has reached the goals.

The click-stream data acquired from the logs give opportunities to compute the criteria parameters per click-stream and than overall statistics for each parameter on the global level. This will be the *reference set*.

For each click-stream there will be a set of records which will indicate the path the user took and the values of all the criteria based on the click-stream. After that the restructuring algorithm can be simulated and information will be gathered of all possible elementary changes. It is yet to decide which of this changes will give benefit globally.

In the final phase there will be records indicating all possible atomic changes for all click-streams. Based on the metaphor of forest paths, not all atomic changes required per click-stream will be tried. Instead a threshold is set and only the most "popular" changes will get through.

Finally, for all atomic changes a new cycle of calculations for each of the click-streams will be performed, as if the changes were all applied. This will give the second set of criteria measurements the *reorganized set*.

If the reorganized set has better measure than the reference set, all the changes of the final set are applied and the new organization structure is published.

AUTOMATED RESTRUCTURING

Where does it happen to have a need for such automated restructuring of the organization of content? This is certainly not needed in books, but there are places where it can be appropriate. Such examples are yellow pages, web directories, knowledge bases and other similar repositories that have in common:

- great number of documents and
- different unrelated aspects according to which a categorization can be done

In fact any categorizing scheme can benefit from the assessment that is proposed in order to be able to evolve and be better.

These is also another aspect to automated restructuring that should not be neglected. This is the case where a huge set of documents should be made visible to a group of users or to the public and when there is no possibility to involve in manual categorization.

In such cases techniques from data-mining can be employed in order to analyse the document set and gather a preliminary topic hierarchy [3]. After that the directory can be made visible to the users and the techniques envisioned in this article can further optimize the structure of the site.

REFERENCES

- [1] Maurice de Kunder, <http://www.worldwidewebsite.com/>
- [2] David Lowe, Xiaoying Kong, NavOptim Coding: Supporting Website Navigation Optimisation using Effort Minimisation, 2004
- [3] N. Liu, C. Yang, Mining Web Site's Topic Hierarchy, 2005