Journal of Web Engineering, Vol. 9, No. 4 (2010) 347–368

VISUAL WEB MINING FOR WEBSITE EVALUATION

VICTOR PASCUAL-CID
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra
08108 Barcelona, Spain
victor.pascual@upf.edu

RICARDO BAEZA-YATES
Yahoo! Research
Diagonal 177 planta 9
08018 Barcelona, Spain
rbaeza@acm.org

J. CARLOS DüRSTELER
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra
08018 Barcelona, Spain
jcarlos.dursteler@upf.edu

Received February 5, 2010
Revised August 8, 2010

In this paper we present an interactive system named Website Exploration Tool (WET) that aims at supporting the evaluation of websites through the exploration of web data. Our prototype offers a set of coordinated visual abstractions in the form of interactive graphs and trees to analyse data graphs within meaningful contexts such as the website structure and users' flow. Apart from classical approaches, our highly interactive system introduces a wide variety of information visualisation techniques that provide visual cues to assist in the process of digging into usage data. Among them, we present a new hierarchical approach for characterising users' flow which simplifies the intricate graphs generated by real users browsing, and a technique for extracting contextual subgraphs simplifying the task of visualising very large websites. The interface of the system has been evaluated with expert analysts that validated the usefulness of the tool for analysing a wide variety of websites.

Keywords: Information Visualisation, Web Mining
Communicated by: D. Lowe & J. Freire

1 Introduction

The increasing importance of the Internet and its wide adoption requires website operators to continuously improve their site. To accomplish this goal it is crucial to understand and evaluate the usability of a Web site. On top of that, information architects, web analysts and website evaluators (from now on, analysts) play the role of interpreting web data, aiming at inferring users behaviour. However, although current Web Analytics tools have suffered a tremendous evolution, analysts must still examine large amounts of statistics that end

up providing little insight into website data. For instance, an example of a typical and straightforward result that one may come up with after a page views analysis (i.e. counting the number of requests that each page has suffered) may hypothesise that pages non visited embraces non relevant content while in highly visited ones occurs the opposite. As pointed out by Spiliopoulou [28], this assumption may only be valid if the user perceives the site in the way that the designer did it or, as proposed by Baeza-Yates and Poblete [3], because there might exist no clear way to reach such pages (either by browsing or by searching). Therefore, to make more suitable and realistic assumptions, we argue that there is a need to provide exploratory tools that allow the examination of website data within a context, such as the hyperlink structure of a website.

On the other hand, since two decades ago the subject of visualising web data has been a recurrent topic of the Information Visualisation community [30] (from now on, Infovis). The aim of Infovis is to provide interactive visualisations that enable the exploration of abstract data to amplify cognition [5]. In that sense, the most used approach has been the usage of node-link diagrams to represent either the hyperlink structure of the site or users' navigation. However, to the best of our knowledge, there is not a system which integrates together both types of visualisation in a coordinated environment, enabling the analysts use both website aspects at the same time to derive conclusions.

In this paper we introduce a highly interactive tool that aims at providing visual contexts to interpret website data. The tool has been designed through an iterative process based upon a field study conducted with experts analysts and enables the exploration of the data through coordinated abstractions from website structure and users' navigation. The tool, named WET, incorporates a set of built-in interactions that enables the analysts to customise the visualisations according to their information needs. Apart from the system itself, our main contributions are a new approach for dealing with large websites supported by the extraction of meaningful subgraphs, and a new hierarchical visual abstraction of users' paths based on a maximum branching algorithm.

The paper is organised as follows: Section 2 presents a review of the most relevant research developed in the area of website data visualisation. Section 3 introduces the Website Exploration Tool (WET), and Section 4 goes into detail to what concerns the Data Processing System. Section 5 describes our new method for dealing with large websites while in Section 6 we address the approaches used in the visualisation system and their main implications. Finally, we present in Section 7 the results obtained from a user qualitative study, which will be further discussed in Section 8 along with the overall conclusions of our work.

2 Related Work

The first visual representations of web data were devoted to ease the navigation of Web spaces, providing interactive representation of the hyperlink structure to avoid the lost in cyberspace phenomenon [30]. As a first example, Andrews proposed a 3D landscape metaphor [1] to represent a set of connected web pages. In the 2D domain, the graph structure of websites was represented as node-link diagrams where pages are considered as nodes and links as edges between them. These approaches deemed the web graph as a hierarchy visualising structural links and hiding cross-reference ones [4]. Cone Trees [27] and Hyperbolic Trees [24] are two examples of early Web visualisations that followed this approach. Nevertheless, Munzner

concluded in [22] that searchers do not need a visual representation of such hyperlink structure as they add cognitive load to interpret them. However, Munzner argued that these techniques could benefit a specific target community such as webmasters or website operators.

Following research efforts focused on visualising users' paths to provide visual cues of their behaviour. Early works extracted navigation sequences from log files [26, 12, 19] that were represented also using node-link approximations. However, there were no reported clues to understand in which context such visualisation may be useful.

The main research trend in the late nineties considered mixing structural and navigational data at the same time, using website structure as a background where navigational data was overlaid [10, 12]. With this approach, Chi *et al* presented a system based on the visualisation of website structure using the radial tree visual metaphor [9], encoding traffic occurred in a link using edge's thickness, and content type as nodes' colour. The same authors also introduced Time Tube [10], which consists of a set of snapshots that represent the evolution of the website with time; and the Dome Tree [8], which is a modification of the radial tree that uses a third dimension where users' paths are displayed.

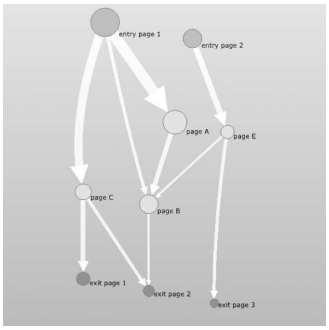
Following the same approach, Chen *et al* [7] proposed a visual data mining system with the ability to visualise multi-layer web graphs. Their approximation is based on the combination of layers that contain information from web usage overlaid on top of the website structure. The authors also proposed a visual metaphor called Polygon Graphs [6], which extends the concept of the radial tree metaphor by generating polygons that appear from the connection of parent nodes in the hierarchy with representative points in the edges calculated according usage data values of its children nodes. This approach generates visual artefacts within the hierarchy, facilitating the detection of outliers and patterns within the data.

In VISVIP [12], Cugini *et al* proposed another tactic for visualising web navigational data using a simplified representation of a webgraph laid out using a force directed algorithm where nodes were colour-coded to denote page type. Users' paths were then represented using smooth curves upon the graph. The time that subjects spent at each page was also represented as dotted vertical lines which height was proportional to the amount of time spent in that node.

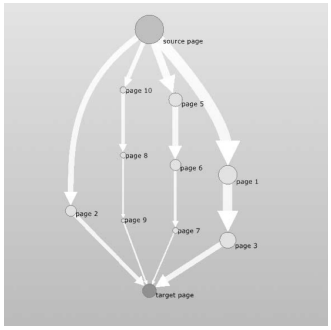
In an effort of studying in depth users behaviour on a site, Keahey and Eick decomposed this problem in a set of visualisations focused on representing the paths between predefined pages [20]. Funnel Charts visualisation, in Figure 1(c), was proposed as a solution for representing dropout rates along a designated sequence of pages. This approach has been widely adopted by the web analytics community, and is available in most of the current web analytics solutions. Other approaches were also presented for depicting the traffic between some entry and exit pages in the site as can be seen in Figures 1(a) and 1(b). These visualisations resemble a vertical tree layout where a small number of entry pages are located in the top of the display, and ending points in the bottom side. Another technique proposed by the authors worth to be commented is the usage of semi-transparent colour coded rectangles superimposed to links on a single web page, representing click density. This approach has also become part of current web analytic tools, although it present problems in pages with a big concentration of links.

Finally, Eick [15] proposed another approach present in current tools, based on three columns, where the left column contain the most frequently referrer pages used to reach a

desired page, placed in the middle column. Destination pages of users after visiting the focus node are placed in the right column. This representation is quite intuitive for understanding users local decisions, although provides very little information of complete users routes.



(a) Paths between source/target pairs



(b) Paths between a set of pages

183	Home Page
77	Widget Products
52	Product XYZ
44	Shopping Cart
41	Register
9	Confirm Purchase
7	Purchase Complete

(c) Funnel chart allows to find dropout rates along a designated sequence of pages

Fig. 1 Some of the most interesting approaches for studying users behaviour proposed in [20].

3 System General Overview

The Website Exploration Tool (WET) is a visual tool that supports the analysis of the usability of a website through the exploration of structure and usage data. By providing a set of flexible representations, WET enables website evaluators to customise the visualisations according to their information needs, generating an environment that supports the *dialogue between an analyst and his information to produce a judgement about an issue*, also understood as Analytic Discourse [29].

Contrary to current Web analytic tools, WET provides an interactive context-based environment that enables the dynamic representation of web metrics on top of visual abstractions of the website structure and usage.

WET is a research prototype prepared to support a wide range of web mining techniques. Figure 2 represents its architecture, structured in three main blocks: the Data Processing System, the Graph's Logic System and the Visualisation System.

First, the Data Processing System collects the topology of the target website and parses

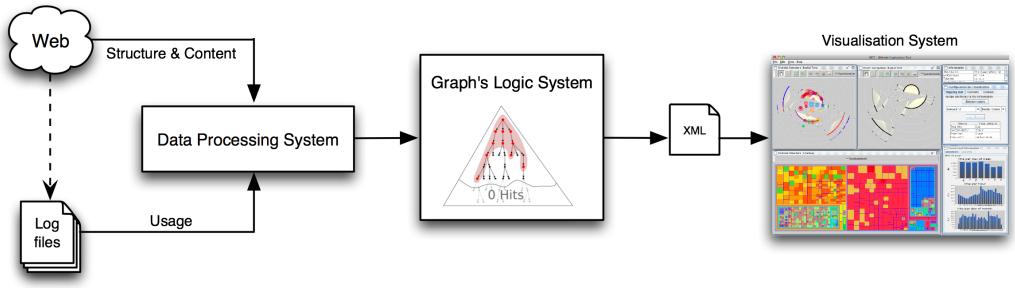


Fig. 2 The WET pipeline is based on three main blocks, the Data Management and Web Mining System, the Graph's Logic System and the Visualisation System.

usage data from raw access log files through the Data Management module. After that, the Web Mining module extracts users' sessions and site metrics from the available data. Finally, the Graph's Logic System extracts target-oriented subgraphs reducing the complexity of very large webgraphs as will be presented in Section 5. A GraphML[a] file containing the final subgraph with the web metrics is used as input to the Visualisation System which is the visual interface that enables the dialogue between the analyst and the data.

4 Data Processing System

The Data Processing System is responsible of gathering, preprocessing and manipulating web data to generate input files that can be loaded into the Visualisation System. The Data Processing System can be decomposed into two modules: the Data Management and the Web Mining module.

4.1 Data Management Module

The Data Management (DM) module is in charge of collecting and storing the data in a manageable and accessible way.

The first task of this module is to collect the topology of the website, that can be accomplished by using two different methods: the first one is to use a crawler, which is a software agent capable of traversing the Web. In general, it starts with a list of URLs to visit, in our case, the home page of the target site, called seed. As the crawler visits this URL, it identifies the hyperlinks existing in the HTML code, storing them in a list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies, such as URL pattern or maximum reachable depth. In our case, we provide the crawler with a URL pattern which enables to distinguish between inner pages of the site and outlinks that point to non desired other sites.

The main advantage of crawlers is that their catch a snapshot of the website, enabling the discovery of the whole structure in a certain instant of time, including unvisited pages that do not appear in access logs. However, this methodology depends too much on the response

[a] http://graphml.graphdrawing.org/

of the network and website host's bandwidth requiring to limit the number of pages to be crawled.

The second method to extract the topology of a website is based upon inferring the graph structure from the access trails left in web server logs. Such trails contain data regarding content requested and referrer page, which enables to infer its corresponding link. The main drawback of this technique is that it does not detect the existence non visited pages, which might generate unconnected graphs. On the other hand, this method can be used offline, avoiding possible network problems.

From our experience, both methods complement each other to ensure the most complete website topology.

After the extraction of the website's topology, the DM module gathers usage data by parsing access log files which are stored, indexed and logically linked to the pages of the webgraph already stored in the database. By now, our system supports both, an intrusive way which requires the usage of a Javascript tracker code that has to be be added in the source code of all the pages in the site; and a non intrusive way which uses log files available in the web server provided by the user.

Finally, the DM module cleans up the irrelevant records existing in the log file that may produce inconsistencies, such as entries related to embedded objects like multimedia files or scripts. Furthermore, access log files are also cleaned from entries generated by crawlers from search engines or spam agents. To do so, we are currently using two methods: the first one focuses on deleting entries with known user agents (such as the ones that identify themselves as bot or crawler); the second one is based on a heuristic approach that takes into account the number of visited pages and the time average spent on them in every user session. After a set of tests, we are currently erasing sessions with more than 50 pages, with an average time of 10 or less seconds. Nevertheless, these parameters may be optimised to the type of usage of any specific target website.

4.2 Web Mining Module

The Web Mining Module (WM) is in charge of converting records of the database in useful and comprehensive information. The most important feature of this system is its flexibility, which is obtained through the usage of a second database that stores queries that will generate metrics from the data existing in the database created by the DM module. With this methodology, the addition of a new metric only requires the development of a new query that gets the information from the main database.

A Java module is in charge of extracting the queries and executing them on the DM database, and finally writing the results in the GraphML file for the Visualisation system. The whole process can be seen in Figure 3.

This architecture enables to easily add and remove new metrics, enabling the reuse of the metrics database for any mining purpose.

4.2.1 Identifying users' paths

The second part of the WM module is in charge of extracting users' sessions. Following the approach proposed in [11], user sessions are extracted by clustering log entries that share the same IP and user agent, with time gaps shorter than 30 minutes (this value can also be defined according to the needs of the Web site).

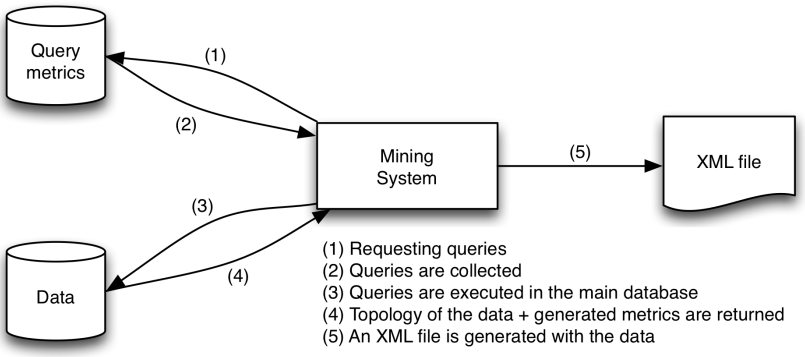


Fig. 3 The web mining system of the DPMS is based on a Java based GraphML builder that applies SQL queries to extract web metrics over the data stored in the DM database.

Due to the stateless nature of the HTTP protocol, the existence of proxies and caches, and the so often use of browser's "back button", webserver log files usually contain incomplete users' sessions. To overcome from this problem we have developed a path completion algorithm based upon the approach proposed in [11], which is based on filling the non contiguous sequence of pages with a "back link". This link is added to the sequence if there exist a previous visited page in the path that has a link to the non connected page. Hence, although there might be no connection between these two pages, the system will show that the user moved to that page, leaving to the analyst the task of analysing how that happened using the available metrics and the website hyperlink structure.

Finally, non-connected user sessions that can not be reconstructed with "back links" are split and considered as different paths. Therefore, a user session may be composed by more than one path, each one representing a different goal or information need.

5 Dealing with big websites: The Graph's Logic System

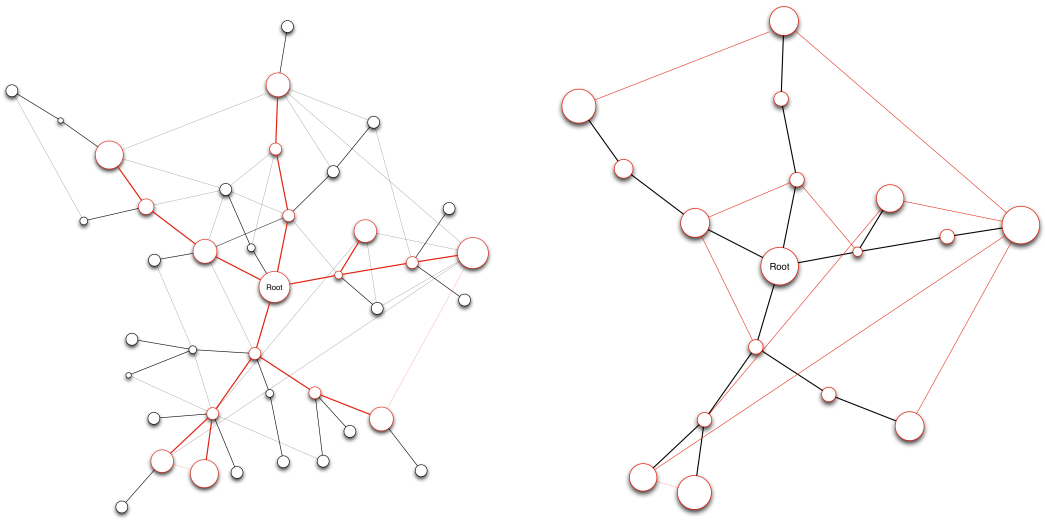
The problem of visualising big websites can be understood as a much broader problem: visualising large graphs. Their visualisation usually aims at discovering topological patterns such as communities or outliers. However, algorithms for laying out such networks tend to generate computational overloads and, more important, create layouts that present readability problems due to the limited screen space available. Even if an algorithm is capable of placing all the nodes in the display so they do not overlap, annoying edge crossings may clutter the visualisation, compromising again the readability.

Most common techniques to overcome this problem apply layouts that use the screen space more efficiently, such as matrix representations [18]; or generate clusters from similar or highly connected components in the graph [21] aiming at compacting them, and hence reduce the amount of space needed to represent the graph. However, this kind of clusters often derive to groups of nodes with no clear semantics.

Despite that, we claim that in some cases users might prefer to understand something from a specific portion of the dataset rather than obtaining global patterns of the complete struc-

ture. Hence, another approach to deal with large graphs is the extraction of representative and meaningful subgraphs that comprehend a set of relevant nodes according to data attributes that provide valuable information to the end user according to a specific information need.

In an effort to extract meaningful hierarchies from very big trees, Furnas proposed the concept of *degree of interest* (DOI) [16] as a numerical value that can be calculated for each node in a tree. This approach enables to create compact abstracted views by selecting nodes above a certain level of interest or threshold. Such concept was applied to nested structures, and can be calculated as a two parts function composed by an A Priori Interest function (API) that defines the importance of a node in the structure, and a distance function D that depends on the distance of the node y to a focus node x. However, this approach only works in hierarchical structures, where the DOI of a parent is always bigger than the DOI of its children [17]. Otherwise, disconnected graphs may appear when applying the threshold, hiding the implicit structure of the graph.



(a) Initial graph. In red, the spanning tree that reaches the relevant nodes from the focus node

(b) Complete subgraph containing links among all the selected nodes.

Fig. 4 The GLS selects relevant nodes (big ones) as well as irrelevant nodes that are part of the structure (highlighted in red).

In our case we applied a similar approach to what Van Ham and Perer proposed in [31] to our Graph's Logic System (GLS) extending Furnas' technique to graphs. To do so, we first generate an ordered list of candidates that contains the nodes that satisfy some weighted criteria based on the analyst needs (big nodes in Figure 4). Such criteria can be defined according to specific data attributes of the dataset, or to general topological metrics such as centrality measures that capture topological interest of the nodes. Furthermore, the system also enables to define a threshold, expressed as a percentage of the global value of a specific attribute, in order to assure the gathering of high percentage of relevant nodes. This can be useful, for instance, when trying to collect the most visited pages in a website, as usually the 20% of the pages retain the 80% of the total of visits.

Once the candidates list has been generated, the system computes a spanning tree of the graph, calculating the shortest paths from a focus node x to every candidate. Traversed nodes in the spanning tree are stored in a final nodes list. Those nodes out of the boundaries of the relevant nodes are discarded, assuring a minimum substructure. The system halts when reaching a maximum predefined number of nodes N, or completes a certain desired amount of levels L. Finally, the GLS retrieves the existing links among the nodes in the final list, to ensure the acquisition of a complete portion of the graph (red links in Figure 4(b)).

Considering that the amount of final nodes is limited and smaller than the number of vertices in the graph, the complexity of our methodology gets reduced to the complexity of the algorithm that computes the spanning tree. As we only concentrate in nodes, edges are considered to have the same weight, which enables the use of a breadth first search algorithm that has a time complexity of $O(|E| + |V|)$.

The main benefit of our approach is the possibility to segment large networks according to the analyst needs while preserving its structure, generating contextual subgraphs that reduce the amount of information to be shown while keeping the same amount of relevant data. Furthermore, assuming that the candidates belong to the same connected network, this methodology assures in all cases the existence of a connected subgraph.

Nevertheless, the main drawback of this technique is that it can not control the number of non relevant nodes that are finally collected in the traversal process, as they depend on the distance between the selected focus and the relevant nodes.

6 Visualisation System

The Visualisation System is an interactive tool that provides several coordinated visualisations of website data. The system has a menu, named mapping tool, which enables to visually encode web metrics in a dynamic way. Hence, the analyst can map at any time web metrics on top of the available visualisations, enabling the representation of multiple metrics at the same time. Web metrics can also be filtered using the built-in interactive legends, which enable to visually express dynamic queries about the data. Furthermore, the system incorporates *scented widgets* that provide guidance when using the edge filtering system, helping to reveal the distribution of links usage on the site.

6.1 Visualising Website Structure

Representing a website as a hierarchy means that from all the existing links (up to $n(n-1)$ since webgraphs are directed), $n-1$ must be selected with the condition that every node except the root must end up having only one incoming edge. As appears in the literature [4], a common technique to achieve that is to extract the website structure using a Breadth First Search algorithm (BFS). This algorithm selects the first non-expanded incoming edge per node that appears in the data traversal. This simple approach enables to extract the theoretical structure of sites designed in a hierarchical way. That is, sites organised with sections, subsections and so forth. However, due to the nature of this algorithm, there might exist ambiguities when parents of a node coexist in the same level. While the BFS algorithm selects the very first parent found, we use a heuristic approach based on the usage of the links to inform this decision. Consequently, the most used links are selected in case of ambiguity. Although this technique does not necessary reveal the theoretical structure of the site, it helps

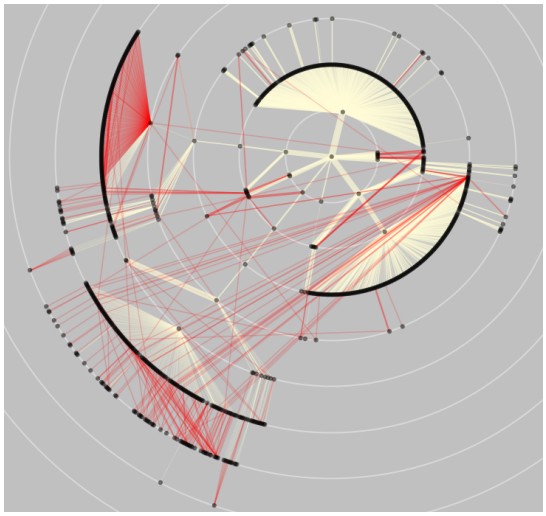


Fig. 5 Visualisation of the structure of a website. Broken links (in red), are visualised on top of the structure.

on making interesting links apparent.

The resulting hierarchy provides an overview of the whole webgraph, locating every page at the minimum distance of the root node selected by the analyst in the GLS. Hence, this method can still be valid for revealing interesting information from non-hierarchically-designed sites.

We use the radial tree visual metaphor to represent such hierarchy, where the root is placed in the centre of the display, and its children are positioned around it in a concentric and recursive manner. We selected this representation as it makes better usage of screen space and because a prior study revealed that it is a visual metaphor easy to understand. As will be covered in further sections, this tree is interactive in the sense that the user can change its focus of interest modifying the root node, which provides new perspectives from the same data.

We also applied visual codings to the radial tree edges, using width and transparency to represent their usage, facilitating the discovery of relevant links. Furthermore, hidden links in the webgraph can be visualised on demand using a toolbar that enables, for instance, to visualise broken links found by the crawler in a superimposed manner on top of the hierarchy. As can be seen in Figure 5, red-coloured edges enable to detect the relative distance of pages containing broken links to the focus of interest (the selected node in the GLS).

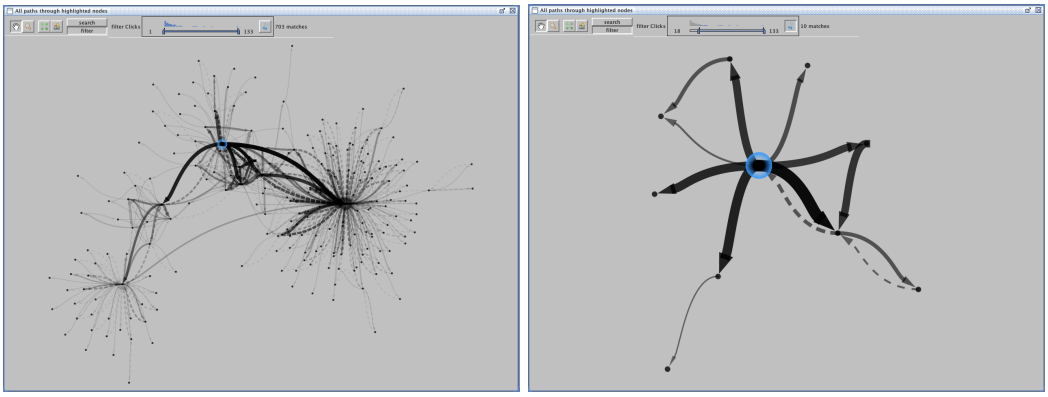
6.2 Visualising user paths

Up to now, we have presented our approach for visualising a webgraph through a hierarchical abstraction based on the website structure. However, as we have seen, the extraction of hierarchies imply hiding links that might be relevant. Therefore, we have developed a visualisation dedicated to the representation of users' sessions called Session Graph, which lays out a graph composed by all the routes performed by users who navigated through one or more selected pages in the target website. During the process of composing this graph, metrics such

as the number of times that a link has been clicked within this context, and the percentage of traffic that it is taking from its source node are displayed on demand.

The Sessions Graph is laid out using a specific force directed algorithm provided by the visualisation package Prefuse, which uses the classic approach of creating repulsion forces among nodes, and spring forces among links. The layout is iteratively calculated until the system reaches a steady state, calculated according the amount of movement of the nodes in a fraction of time. However, the non-planarity of the graphs that characterise users movements usually needs too much time to reach a steady state. Therefore, we use a timeout parameter, with a default value of 6 seconds. The algorithm running time is $O(max(NlogN), E)$, as reported in the package documentation.

Edges of the graph are visually encoded using transparency and thickness, with a square root scale as can be seen in Figure 6(a). The main benefit of our approach is that it enables to explore the navigated pages between several nodes of interest. To do so, WET provides a filtering system that enables to dynamically filter links according to their usage, making navigation trends apparent. Once links have been filtered, the user can re-run the layout algorithm with the remaining links and nodes reaching more understandable visualisations as can be seen in Figure 6(b).



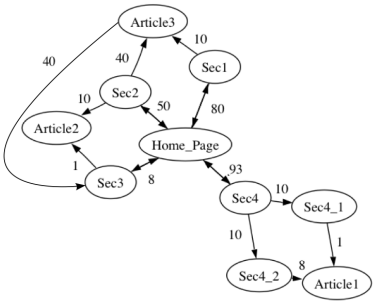
(a) The Session Graph

(b) The graph gets understandable and revealing after filtering non relevant edges and recomputing the force directed layout.

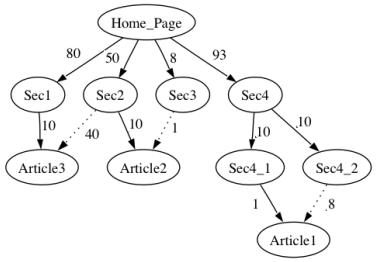
Fig. 6 The Session Graph is generated by superimposing all the paths that pass through one or more selected pages of the website (left). The filtering system enables to remove less frequent links, facilitating the discovery of main usage trends (right).

This approach has two main benefits compared to current methods used in web analytics packages such as the funnel chart:

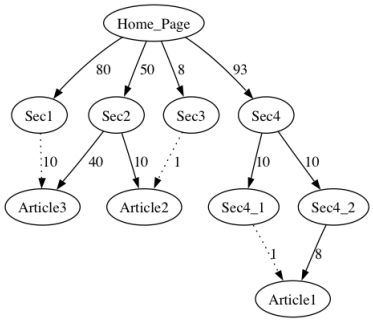
1. Enables to understand the ecosystem of a page: we define the ecosystem of a page as the set of pages visited by the users in the paths across a specific page. This definition came out through a discussion with an expert analyst (see Section 7), who stated that current tools barely provide such information, which might be of crucial importance in specific contexts such analysing the navigated pages around a conversion page.



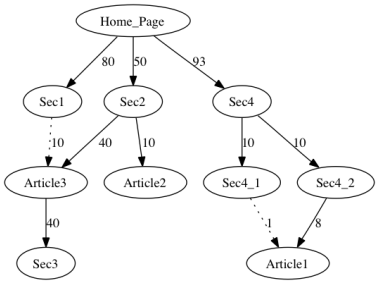
(a) A graph representing a website



(b) Hierarchical tree from the BFS algorithm



(c) Hierarchical tree from the W-BFS algorithm



(d) Hierarchical tree from the BM algorithm

Fig. 7 An example of the hierarchies obtained with the different algorithms.

2. Enables to discover highly repeated subsequences: rather than focusing on the most used paths performed by the users, our approach allows to visually discover highly repeated subsequences of paths.

6.3 Characterising users' navigation as a hierarchy

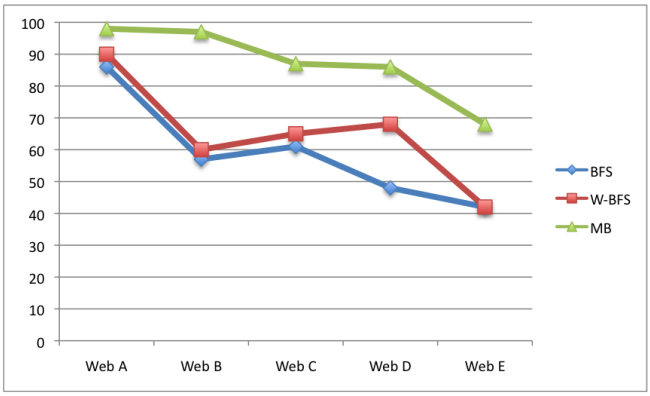
In previous sections we have shown how structural hierarchies can be extracted from a webgraph, as well as our approach for depicting users paths that can be dynamically filtered. Hereafter we introduce a new approach for generating hierarchical structures from websites called "Usage Tree".

The generation of hierarchies to model users paths was applied in previous research. For instance, Pei *et al.* [25] generated a hierarchical structure from users sessions to facilitate the execution of their mining algorithms. Moreover, Dierbold and Kaufmann [13] presented a visual approach for depicting site maps based upon a Sugiyama layout applied to users paths which was considered as a collaborative approach to distinguish possible interesting content.

To present our approach, we first define the concept of *Local Maximum Link (LML)* as a link that provides the highest amount of traffic to a node in the webgraph. Hence, each node in the webgraph, except the root (typically, the home page or any landing page on the site), will have at least one LML. Our goal was to evaluate if it is possible to generate a comprehensive

				BFS		W-BFS		MB	
website	#nodes	#links	Total LML	$\overline{X}$	σ	$\overline{X}$	σ	$\overline{X}$	σ
A	143	565	140	86%	10%	90%	12%	**98%**	0%
B	366	21865	91	57%	5%	60%	5%	**97%**	.4%
C	957	4898	287	61%	1%	65%	1%	**87%**	.1%
D	2812	51807	2017	48%	1%	68%	3%	**86%**	.1%
E	374	11767	291	42%	3%	42%	3%	**68%**	.2%

(a)



(b)

Fig. 8 Percentage of LML found by each algorithm. Green line represent the MB algorithm, which outperform the BFS algorithms in terms of LMLs found.

hierarchy containing the maximum number of LMLs, in order to get a visualisation based on users' behaviour, rather than on its theoretical structure. Our approach uses a Maximum Branching methodology based upon Edmond's algorithm [14], which collects the LML of every node in the site, and resolves the cycles that this process may produce.

6.3.1 Methodology evaluation

To evaluate our new technique we used a heuristic approach based on the number of LMLs selected by each one of the already presented algorithms for extracting a hierarchy from a graph: the Breadth First Search algorithm (BFS), the Weighted-Breadth First Search algorithm presented in Section 6.1 (W-BFS), and the Maximum Branching (MB) described above. A comparison between the three approaches can be seen in Figure 7.

The experiment consisted in counting the number of LMLs selected by each algorithm computed in five different websites, each one rooted randomly at a hundred different nodes. From the records generated, we deleted those executions that were unable to generate a tree because the chosen root didn't have any outgoing link. The selected sites had different sizes and are characterised in the top table of Figure 8, and were selected as they represent different types of sites, going from a personal website with a few hundred pages and links (website A), to a highly connected blog with about three thousand pages and fifty thousand links (website D).

Figure 8 shows the average percentage of LMLs found by each algorithm on each website.

These values have been calculated according to the Total LML value, which represents the desired number of LMLs (ideally $(n-1)$) to be found in a site minus the number of non-relevant LMLs. We define a *non-relevant LML* as a link that points to a page that has never been part of a navigation sequence. These edges are considered irrelevant, as none of them reported traffic to its target node.

As can be seen in the table of Figure 8(b), results changed according to the connectivity of the website. Nevertheless, the MB algorithm always shows more LML than the Breadth First Search approaches, improving the results in an average of a 20%.

From this experiment we conclude that the MB algorithm produces hierarchical visualisations that improve the accuracy of the breadth first search approximations in terms of number of LMLs shown. However, this approach takes into account all the paths performed in the site regardless its starting point. Therefore, and contrary to what the visualisation suggests, nodes' depth becomes meaningless. Figure 9 illustrates this phenomena, were the highlighted node is a very important landing page in the site. Thanks to the built-in filtering system, we can see that there are frequent routes from this node. However, due to the non existence of previous links with the same frequency to this node, we can infer that such nodes are only visited by users that landed in such page. Despite this fact, the MB approach locates them in depths 3 and 4, suggesting that this is the number of clicks made by the users to reach them from the root node. Consequently, our new hypothesis is that the MB approach would generate more intuitive results when applied only to the graph that comes out from the routes started at a given root node.

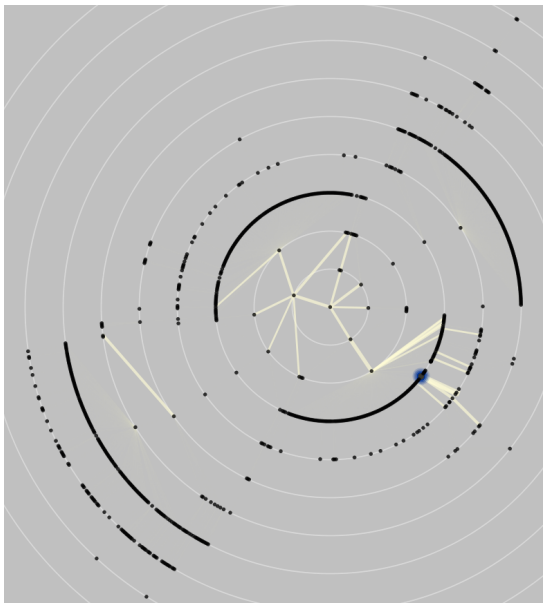


Fig. 9 The highlighted node in the bottom right branch of the tree is an important landing page of the site that brings traffic to a set of nodes.

To prove our hypothesis, we conducted again the same experiment, running the Edmond's

algorithm upon the graphs that comes out from aggregating all the routes that started at the given root node. This operation was repeated for the five sites used in the previous experiment. Collected results show that the extracted hierarchies provide an average of 99% of the LMLs existing on that navigation paths, which means that they are an accurate summary of the graphs composed by all the paths that started at every landing page.

Taking into account that the algorithm runs in $O(EV)$, we characterised such graphs to see if the algorithm is computationally feasible to get responses in a reasonable response time. We used usage data from six months of those already mentioned 5 websites. The extracted model presents a polynomial curve corresponding to the function $E = 0.47 \times V^{1.42}$ in graphs with less than a hundred nodes. However, the degree of the function decreases in bigger graphs (i.e. graphs with more than 100 nodes), being the function $E = 1.14 \times V^{1.16}$, showing a quasi linear growth, as can be seen in Figure 10.

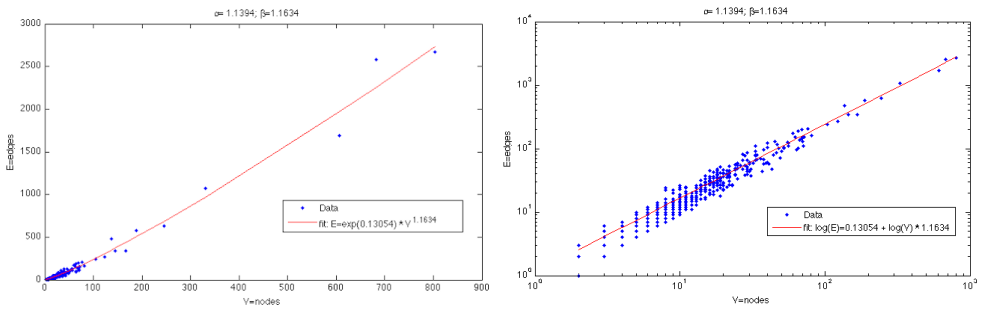


Fig. 10 Relation between nodes and edges to be handled by the Edmond's algorithm. The fitting curve represents the growth of graphs with more than 100 nodes. Right plot is in log-log scale.

We finally characterised results of time performance of the Edmond's algorithm upon such webgraphs. As can be seen in Figures 11, once outliers have been removed, the algorithm presents a linear behaviour (being the fitting curve $T = \exp(-9.7) \times V^{0.989}$) among the available graphs which confirms its adequacy considering the size and topology of the graphs that it has to deal with.

The main benefits of our approach are:

- The generation of a résumé of users navigation from a specific landing page which, taking into account that the maximum number of clicks in a site is fairly limited, generate small trees easier to layout and visualise.

- Nodes depth becomes meaningful, as it represents the average number of clicks that the users make to reach a page from the rooted node.

- The coordinated highlighted environment provided by WET enables to compare the shortest path to reach a node from a root page with the real path performed by the users, as can be seen in Figure 12. This feature was specially well rated for domain analysts that took part from the formative evaluation described in Section 7.

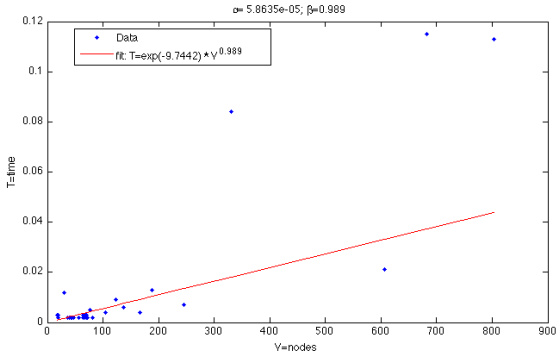


Fig. 11 Time performance of the Edmond's algorithm applied to the graphs formed by users paths starting at a specific page.

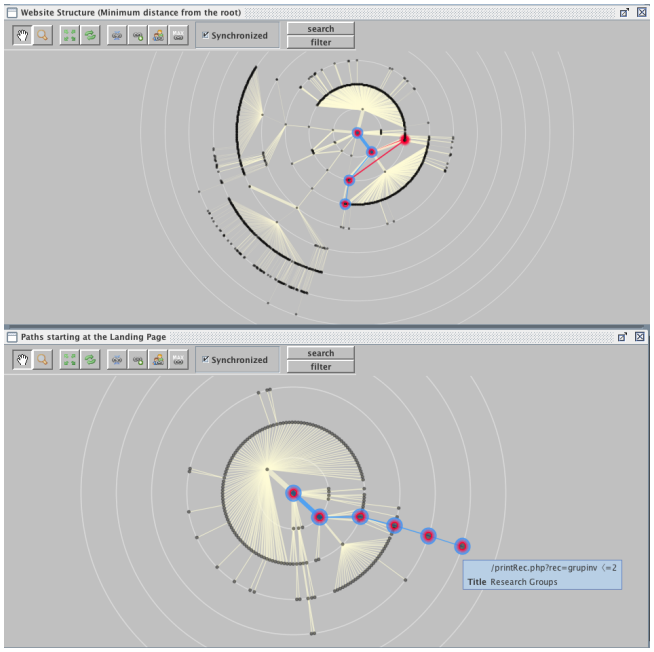


Fig. 12 The coordinated highlighting enables to explore the shortest path to reach a node from a specific root (top frame) comparing it with the real users behaviour (bottom frame). This image reveals that users made five steps to reach a specific page, while the website structure reveals that the same node can be reached with only three steps.

6.4 Interaction Design

The interaction design of WET has been developed as the vehicle that drives the analytic discourse, enabling the analyst to change the visualisations according to his information needs. The main interactive feature of WET is the Mapping Tool, which is the system that enables the mapping between web metrics and visual attributes (e.g. size, colour, shape) of the

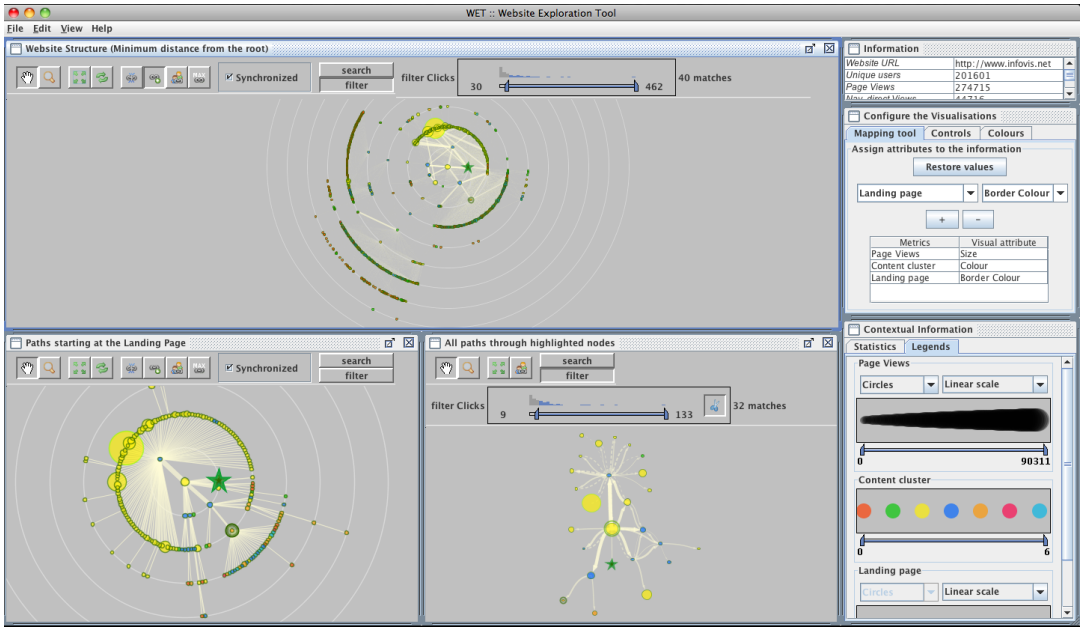


Fig. 13 A screen shot of WET where all the visualisation use the same visual encodings. Upper left frame displays the 'Structure Tree', bottom left frame shows the 'Usage Tree' and bottom right displays the 'Session Graph'.

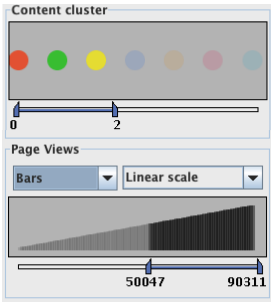
nodes. The visual codings selected by the user are propagated in all the active visualisations, providing the analyst the same information across the different visual abstractions explained above (see Figure 13).

As we introduced before, all the visualisations are coordinated, meaning that any node or edge highlighted in one visualisation is also stressed in the others. Furthermore, analysts may change the focus of interest at any time by dragging any node to the centre of the hierarchical visualisations. This action causes the recalculation of all the visualisations, using the selected node as main focus. To avoid a rapid change of the position of the nodes in the different visualisations, WET performs a transition [33] between the old and the new positions of the nodes, enabling the user to better understand the transitions between the layouts.

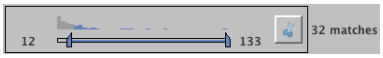
6.4.1 Guiding the analytic discourse using visual cues

As one of the main goals of the visualisation system is to provide interactions so the analysts can explore and navigate web data, we have provided it with small widgets enhanced with embedded visualisations that enable users to define filtering thresholds. Such components, also known as scented widgets [32], are basically double sliders that enable the analyst to define upper and lower filtering thresholds to delimit data values of items to be shown in the display. Scented widgets enable to understand the context of the data, allowing the reduction of data on demand.

The first widget is used in the legends that appear every time the analyst uses a new visual encoding for a certain metric. Figure 14(a) shows an example of two legends representing a



(a) The slider makes legends more powerful by enabling them to work as a filter.



(b) A histogram placed next to every edge filtering slider provides information scent on links usage.

Fig. 14 Filters in WET are accompanied with visual cues that assist the filtering process.

categorical metric revealing different clusters of the data, and a numerical one (number of page views). The bottom sliders are a visual way for representing dynamic queries that can be combined to perform complex queries on the data. To what concerns the visualisations, high transparency is applied to nodes that do not accomplish the selected thresholds, at the same time that they lose its interactivity capabilities. Moreover, as can be seen in the image, legends also enable to change the scale and shape of the nodes, affecting mainly the size of visual attribute, which can be represented in one dimension (using bars) or in two (using normal circles).

Regarding edge's filtering, the system provides scented widgets provisioned with small histograms of the usage of the visible links. This histograms aim at providing a visual cue about the number of edges that might be hidden after moving the sliders. Figure 14(b) shows an example of a scented widget used in WET, which displays the distribution of link usage in a site, revealing that a lot of links are barely clicked while very few ones are popular.

7 Evaluation

After performing a set of informal interviews with real users that aided the design process, we conducted a formative study to better understand the usefulness of WET. Andrews defined in [2] the concept of formative evaluations as "tests that involve a small number of users using an interface to gain insight into which problems occur and why they occur", and Munzner [23] states that this kind of evaluation are "intended to provide guidance to the designers on how to improve a system and answer the question 'can I make it better?"'.

The experiment was tested with one pilot user, and finally conducted with five domain experts, all of them with more than five years of experience in the field of web analytics, who analysed a website through a set of 10 tasks. The website had nearly 1.000 pages and 5.000 links, and usage information from six months, having almost 50.000 user sessions. The dataset was specially interesting as it had a clear structure, having a mirrored structure due to the multilanguage nature of the site. Users were given a 15 minutes training with a real dataset, and spent almost two hours with WET analysing the dataset.

Tasks were defined with the webmaster of the website, which used WET to analyse his

own site, and were defined to allow the users to use all the visualisations provided by WET, including tasks such as discovering the most visited pages, or understanding the navigation of the users from a landing page. The aim of these tasks was to see if providing typical web metrics within a context enabled analysts to come up with more insight on the data or, at least, more elaborated hypothesis according to their judgement. The rest of the tasks were more complex and required users to explore the visualisations to get a general understanding of the site. A post test questionnaire was followed by a final interview that helped to collect qualitative evidences of the usefulness of the tool as well as its main drawbacks.

7.1 Results

Users accomplished without much problems the tasks that involved the finding of specific pages such as the most visited one. Furthermore, we observed that in some tasks, like the case of deciding which page with broken links would they fix in the first place, analysts stated that the superimposed visualisation of broken links upon the website structure (see Figure 5) allowed them to form more complex hypothesis difficult to appear with current tools. While these tools provide only information about the number of broken links that each page has, our approach let the users incorporate more complex reasonings. Apart from stating that they would solve the page with more broken links, an analyst stated: "watching the data in that way, I would start solving the pages nearest the home page, or near the conversion page rather than the ones with more broken links". Moreover, analysts argued that the rest of the visualisations would help them to foster their attention into most navigated routes, in order to start fixing the pages that would affect users' navigation.

The visualisation of the website structure was rated very positively in the post test questionnaire. However, depending on the task, users reckon that sometimes such approach could be misleading as it does not always show the exact hierarchy of the site.

Regarding the two visualisations of users click stream, users had problems to understand the Navigational Tree as two of them misunderstood the difference with the data showed in the Session Graph. To improve its understanding the analysts suggested minor interface changes, such as the titles in the visualisation frames. On the other hand, the Session Graph was very positively rated. Users understood the visualisation without problems, and commented that this approach could help them, stressing that "with this visualisation one can see the ecosystem of pages involved in the routes to a specific content", which are the long tail of visited pages by users that did not perform most typical routes. All the users stated that this is really difficult to get with current tools. The Usage Tree, the Sessions Graph and a funnel chart could be complementary visualisations as all of them provide solutions to different problems. The funnel chart helps to reveal how many users go through a set of predefined steps needed to reach a desired goal, hiding information on what the users did before reaching every step. The Usage Tree helps on revealing the most common routes performed from a specific landing page, representing several funnels al the same time (considering that the edges' width is used to map amount of traffic), and allowing to discover the average number of clicks made by the users. The Sessions Graph appears to be the most complex yet powerful approach. Its ability to visualise the paths between any given set of pages presents a new possibility for analysts, which do not currently have any tool for digging into users paths in fine grain detail. Therefore, this approach may help on discovering different routes that may

converge to finally reach the same page, or to see how users that reach a specific page split to navigate through other pages containing related information.

All the users also stated that the tool is not easy to use, but they showed positive about improving their skills with it after more training. From our observations and the subsequent recordings review, we detected a main usability problem which involved the existence of node overlapping. Such problem used to prevent users from selecting desired nodes, and was reported as very annoying. This problem has been already addressed through the usage of a fish eye view lens that deforms the space, isolating nodes closer to the mouse.

This evaluation also helped us to rise new research questions: does all the sites of the same type share a common structure? If so, can our website structure visualisation be used for comparing such structure with the common one?

A further improvement of the tool should definitely consider the availability of temporal information, as it might provide more context on the data displayed.

In the final post questionnaire interview, users stated that the interactive visualisations provided by WET would be very beneficial to their work, but they argued the need to integrate the tool with the data coming from analytic tools such as Google Analytics or Site Catalyst.

8 Concluding Remarks

Evaluating the usability of a website is a problem that goes beyond the analysis of statistics that mainly provide information from website usage. However, usability experts, information architects and web analysts in general are still struggling with tables full of statistics that end up providing little insight on how a website is used.

In this paper we have presented the development of the full architecture of a system that takes care of the collection, preprocessing and generation of metrics that can be visually explored. Contrary to current Web analytic tools, WET uses a context-based environment that enables the representation of web metrics on top of visual abstractions of the structure and usage of a website, mapping them with visual attributes that change the look of the nodes. Moreover, WET offers a coordinated environment that provides different visualisations as well as interactive techniques to asses the usability of a website.

To what concerns the visualisation of the website structure, we have developed the 'Structure Tree', which is a new approach based on a modification of the well known Breadth First Search algorithm that is able to disambiguate coexisting parents of a node in the same depth. The system also incorporates interactions to recalculate the site's hierarchy starting at any page on the site, enabling to understand the number of clicks needed to reach the rest of the pages in the site. We have also seen that this approach allows to easily discover possible design errors in the hyperlink structure of the site.

Furthermore, we have also studied the viability of representing users' click stream in a hierarchical manner using five websites with different sizes and topologies. We found that a Maximum Branching algorithm is a suitable approach for extracting hierarchies from the graph formed by all users' paths that start at a certain page. Such visual abstraction enables to easily compare the website hierarchical structure with users' behaviour, and hence, to easily compare if users make an optimal usage of the designed hyperlink structure. Complementary to this visualisations, WET also provides the Sessions Graph, which uses a force directed layout to display all the routes performed by the users in the site.

Moreover, we have tackled the problem of visualising very large websites by defining a system that is able to extract relevant subgraphs according to the analysts needs. This system can then be used for segmenting the site in areas of interest that can be analysed independently.

We finally evaluated our visual system conducting a qualitative study with five domain experts. Our observations revealed that although users had some difficulties with the system mainly due to the lack of training, the visualisations allowed analysts to generate more complex conclusions about the data. In particular, the visualisation of users' paths were very positively rated among the users.

References

1. Andrews, K. Visualising cyberspace: information visualisation in the harmony internet browser. In *Information Visualization, 1995. Proceedings.* (Oct. 1995), pp. 97–104.
2. Andrews, K. Evaluating information visualisations. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors* (2006), ACM, pp. 1–5.
3. Baeza-Yates, R., and Poblete, B. A website mining model centered on user queries. *Semantics, Web and Mining. M. Ackermann et al. (Eds.): EWMF/KDO 2005 Springer LNAI 4289* (2006), 1–17.
4. Botafogo, R. A., Rivlin, E., and Shneiderman, B. Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Transactions on Information Systems (TOIS) 10*, 2 (1992), 142–180.
5. Card, S. K., Mackinlay, J. D., and Shneiderman, B., Eds. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
6. Chen, J., Zheng, T., Thorne, W., Huntley, D., Zayane, O. R., and Goebel, R. Visualizing web navigation data with polygon graphs. In *IV '07: Proceedings of the 11th International Conference Information Visualization* (2007), IEEE Computer Society, pp. 232–237.
7. Chen, J., Zheng, T., Thorne, W., Zaiane, O. R., and Goebel, R. Visual data mining of web navigational data. In *IV '07: Proceedings of the 11th International Conference Information Visualization* (2007), IEEE Computer Society, pp. 649–656.
8. Chi, E., Pirolli, P., and Pitkow, J. The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a web site. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems* (2000), ACM, pp. 161–168.
9. Chi, E. H. Improving web usability through visualization. *IEEE Internet Computing 6*, 2 (2002), 64–71.
10. Chi, E. H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R., and Card, S. K. Visualizing the evolution of web ecologies. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems* (1998), ACM Press/Addison-Wesley Publishing Co., pp. 400–407.
11. Cooley, R., Mobasher, B., and Srivastava, J. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems 1* (1999), 5–32.
12. Cugini, J., and Scholtz, J. Visvip: 3d visualization of paths through web sites. In *DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications* (1999), IEEE Computer Society, p. 259.
13. Diebold, B., and Kaufmann, M. Usage-based visualization of web localities. In *APVis '01: Proceedings of the 2001 Asia-Pacific symposium on Information visualisation* (Darlinghurst, Australia, Australia, 2001), Australian Computer Society, Inc., pp. 159–164.
14. Edmonds, J. *Optimum Branchings.* J. Res. Nat. Bur. Standards, 1967.
15. Eick, S. G. Visualizing online activity. *Communications. ACM 44*, 8 (2001), 45–50.
16. Furnas, G. Generalized fisheye views. *ACM SIGCHI Bulletin 17*, 4 (1986), 23.

17. Heer, J., and Card, S. Doitrees revisited: scalable, space-constrained visualization of hierarchical data. *Proceedings of the working conference on Advanced visual interfaces* (2004), 421–424.
18. Henry, N., and Fekete, J. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 677–684.
19. Hong, J. I., and Landay, J. A. Webquilt: a framework for capturing and visualizing the web experience. In *WWW '01: Proceedings of the 10th international conference on World Wide Web* (2001), ACM, pp. 717–724.
20. Keahey, T. A., and Eick, S. G. Visual path analysis. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)* (2002), IEEE Computer Society, p. 165.
21. Mladenic, D., and Grobelnik, M. Visualizing very large graphs using clustering neighborhoods. *Lecture Notes in Computer Science: Local Pattern Detection 3539/2005* (2005).
22. Munzner, T. Drawing large graphs with h3viewer and site manager. In *GD '98: Proceedings of the 6th International Symposium on Graph Drawing* (1998), Springer-Verlag, pp. 384–393.
23. Munzner, T. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 921–928.
24. Munzner, T., and Burchard, P. Visualizing the structure of the world wide web in 3d hyperbolic space. *Proceedings of the first symposium on Virtual reality modeling language* (1995), 33–38.
25. Pei, J., Han, J., Mortazavi-Asl, B., and Zhu, H. Mining access patterns efficiently from web logs. In *PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications* (London, UK, 2000), Springer-Verlag, pp. 396–407.
26. Pitkow, J. E., and Bharat, K. A. Webviz: A tool for world-wide web access log analysis. In *Proceedings of the 1st International WWW Conference* (1994), pp. 271–277.
27. Robertson, G., Mackinlay, J., and Card, S. Cone trees: animated 3d visualizations of hierarchical information. *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology* (1991), 189–194.
28. Spiliopoulou, M. Web usage mining for web site evaluation. *Communications of the ACM 43*, 8 (2000), 127–134.
29. Thomas, J., and Cook, K. Illuminating the path: The research and development agenda for visual analytics. *IEEE Computer Society* (2005).
30. Turetken, O., and Sharda, R. Visualization of web spaces: state of the art and future directions. *SIGMIS Database 38*, 3 (2007), 51–81.
31. van Ham, F., and Perer, A. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 953–960.
32. Willett, W., Heer, J., and Agrawala, M. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1129–1136.
33. Yee, K., Fisher, D., Dhamija, R., and Hearst, M. Animated exploration of dynamic graphs with radial layout. *Proceedings of the IEEE Symposium on Information Visualization 43* (2001).