

Measuring Task Learning Curve with Usage Graph Eccentricity Distribution Peaks

Vagner Figueredo de Santana
IBM Research
IBM
São Paulo, SP, Brazil
vagsant@br.ibm.com

Rogério Abreu de Paula
IBM Research
IBM
São Paulo, SP, Brazil
ropaula@br.ibm.com

Claudio Santos Pinhanez
IBM Research
IBM
São Paulo, SP, Brazil
csantosp@br.ibm.com

Abstract— Interaction logs (or usage data) are abundant in the era of Big Data, but making sense of these data having Human-Computer Interaction (HCI) in mind is becoming a bigger challenge. Interaction Log Analysis involves tackling problems as automatic task identification, modeling task deviation, and computing task learning curve. In this work, we propose a way of measuring task learning curve empirically, based on how task deviations (represented as eccentricity distribution peaks) decrease over time. From the analysis of 427 event-by-event logged sessions (captured under users' consent) of a technical reference website, this work shows the different types of learning curves obtained through the computation of how deviations decrease over time. The proposed technique supported the identification of 6 different task learning curves in the set of 17 tasks, allowing differentiating tasks easy to perform (e.g., view content and login) from tasks users face more difficulties (e.g., register user and delete content). With such results, HCI specialists can focus on reviewing specific tasks users faced difficulties during real interaction, from large datasets.

Keywords— Interaction log analysis, task modeling, task deviation, empirical user studies, client-side events, usage logging, usage modeling, usability.

I. INTRODUCTION

This work is part of a long-term project on the understanding of interaction logs, aiming at making sense of such data, supported by Human-Computer Interaction (HCI) theory, techniques, and methods. A proposal about using **eccentricity distribution peaks** (EDPs) as a way of modeling task deviations was published in the proceedings of the *XVI Brazilian Symposium on Human Factors in Computing Systems*. Eccentricity is a vertex metric—in the context of Graph Theory—that measures the longest possible number of hops to take in a graph without revisiting vertices [27]. From the analysis of eccentricity distribution of a graph, it is possible to grasp graph topology in a summarized way. Hence, if there is a peak in an eccentricity distribution for a directed and connected graph, it means that there are multiple ways of reaching the most distant node. For graphs representing detailed interaction events as mouse events, keyboard actions, and browser functioning, this means that there are multiple ways of performing a certain task.

The overlap between the previous work and this one encompasses the dataset studied and the use of EDPs.

However, this work expands the previous one by using the EDPs as a metric for measuring task learning curve.

The literature counts on different tools and models that support the understanding of user actions while interacting with websites by the analysis of **server log files**. This approach is being considered for long for different reasons (e.g., ease of obtaining such data from Web servers). Examples of such tools are *Descubridor de Conhecimento en la Web* [9], LumberJack [7], Web Utilization Miner [23], WebCANVAS [5], WebQuilt [26], and WebSIFT [8]. However, server logs do not provide details on how users interacted with user interface (UI) elements. More recent initiatives consider **client-side data** in order to understand user actions in details, for instance, MouseTrack [1], MultimodalWebRemUSINE [19], UsaProxy [2], WELFIT [22], WUP [6], WebHint [25], and USABILICS [24]. Hence, client-side data emerged as a way of gathering **detailed interaction data**, allowing a better understanding of user actions while they are interacting with UIs.

One of the invariants regarding the existing systems is that tools focus on providing insights about the usability level of the evaluated UIs. According to the International Organization for Standardization (ISO) [13], usability is the capacity of a product to be used by specific users to realize certain tasks with efficacy, efficiency, and satisfaction, in a certain context of use. Nielsen presents that usability can also be defined in terms of 5 quality components [17]:

- **Learnability:** How easy users accomplish basic tasks the first time they use the design?
- **Efficiency:** Once users have learned the design, how quickly can they perform tasks?
- **Memorability:** When users return to the design after a period, how easily can they reestablish proficiency?
- **Errors:** How many errors occur, how severe are they, and how easily can users recover from them?
- **Satisfaction:** How pleasant is for the user to use the design?

Considering these definitions, task emerges as a key term. According to Lewis and Rieman [15], "*To get a good interface you have to figure out who is going to use it to do what.*" Thus, supporting the understanding of the **task learning curve** is

fundamental for grasping the overall user behavior during interaction and the usability level of the UI being used. Key questions emerging in the context task learning curve are:

- How can we measure learnability?
- How can we measure memorability?
- How can we measure task learning curve from real and detailed interaction logs?

When dealing with big datasets, answering these questions can reduce the amount of data to consider in in-depth analysis. For example, knowing what task (or type of task) is hard for certain users, supports filtering data considering specific tasks for specific user profiles. In addition, previous work [21] presented first results on how graph topology (e.g., number of shortest paths and betweenness) can represent proxies for usability metrics considering data gathered during a user test that counted on 10 participants, run in a controlled environment. In this paper we extend the idea of usability proxies to investigate learning curves from a real dataset of detailed interaction data. Our proposal involves using EDPs to measure task learning curve from the usage data logged during real tasks, remotely, and asynchronously, totaling 427 sessions. Hence, this work contributes with a method to measure task learning curve by using EDPs and the clusters of sessions based on the number of peaks.

This work is organized as follows: section II presents related works; section III details the method, dataset, and the graph structure considered; section IV shows and discusses the results obtained, and; section V concludes by presenting outcomes, limitations of this study, and future steps.

II. RELATED WORK

Initiatives on modeling user interface usage and tasks in Human Computer Interaction grown after initiatives as GOMS (Goals, Operators, Methods, and Selection rules) [12] and CTT (Concur Task Trees) [18]. Task modeling aims at decomposing tasks into smaller activities and defining distance between the task specification and the task executions. When analyzing this distance overtime, it is possible to identify learning curve. However, such approaches depend on creating and maintaining task models as system evolves.

One alternative to task modeling involves data-driven approaches. Literature counts on proposals for representing actions as graphs considering clicks [16] or detailed client-side data, including the identification of interaction patterns and usability problems [20]. In addition, probabilistic models also are present in the literature, supporting the analysis of common taken paths [3]. Moreover, additional approaches involving graphs consider user profile [9] or search query logs performed by users [10]. Data-driven approaches represent valuable initiatives towards evaluating interaction data at scale. However, to the best of our knowledge, none of the presented prior-art count on a method to identify task learning curves from detailed interaction data.

Finally, although the literature counts on approaches for detecting deviation from previously defined task models or data-driven approaches for identifying interaction patterns via

graphs, queries, and profiles, existing works do not provide means of evaluating task learning curve from interaction logs at scale, without depending on task specifications.

III. METHOD

This section details the dataset analyzed, the data structure used, and how the analysis was performed.

A. Dataset

The dataset considered in this work is composed of 427 logged sessions captured during a two-year period¹. The website where the sessions occurred is called WARAU (Websites Adaptation to Requirements of Accessibility and Usability). WARAU is a technical reference and UI evaluations repository. The website supports the development of high quality websites integrating technologies as HTML, CSS, and JavaScript, aiming at Accessibility and Usability.

The event streams related to the 427 sessions analyzed were captured by the evaluation tool WELFIT (Web Event Logger and Flow Identification Tool)[22]. The logging of UI events occurred remotely under users' consent, after the acceptance of an invitation to participate in this study. The invite was presented once for every user that accessed the website. The data logged comprises all events triggered at the UI while users performed real tasks. The dataset counts on 241,413 events (mean of 564.4 events per session).

The following list presents descriptive information regarding accesses to the reference website in the period this study took place:

- Total of 220,448 sessions (mean of 9,185 per month);
- New sessions represent 89.03%;
- The average duration of the session is 38 seconds;
- Users view in average 1.21 page per session.

These characteristics highlight the role of the website as a source of technical information, since most of the users land in the website coming from a search engine, interact with the content, and then leave the website.

The following tasks were identified in the dataset:

- 1) Perform login;
- 2) Search for or filter topics;
- 3) View content;
- 4) View accessibility evaluation form sample;
- 5) View the "about page" presentation;
- 6) View heuristic evaluation form sample;
- 7) View topics index;
- 8) Create an accessibility evaluation;

¹ Dataset available at:

https://www.researchgate.net/publication/320336010_WARAU_event_streams_captured_by_WELFIT

- 9) View publications page;
- 10) View references page;
- 11) Access the administration page;
- 12) Reset password;
- 13) View comment;
- 14) Delete content;
- 15) Create a heuristic evaluation;
- 16) Register user;
- 17) View “about page”.

This dataset was considered because it counts on details performed during the usage, allowing the present analysis of showing task deviations. Moreover, since the website is commonly used as a reference, it would be interesting to identify task deviations and tasks completion characteristics in order to characterize how users use a technical reference website.

B. Data structure

In order to perform the data analysis and to compare with other techniques summarizing usability information of observational data, the logged data was structured according to the technique presented in [20]. The graph structure considered (also called as usage graph) is a weighted directed graph $G = (V, E, w)$, where:

- $V = A \cup \{start, end\}$ is the set of actions/events triggered at a certain user interface element (e.g., the event *mouseover* triggered on a submit button is represented by one vertex, say v_i , and then a click over the same submit button is represented by a second vertex, say v_j). The vertices *start* and *end* represent the start and the end of the logged session, visit, pageview, or any other period being represented in the usage graph.
- Each $v \subset V$ counts on information representing the mean distance in hops from start to v , represented as $d(v)$, the mean timestamp in milliseconds from the *start* to v , represented as $t(v)$, and the total occurrences of the same event over the same UI element.
- $E \subseteq V \times V$ is the set of directed edges, where e connects two vertices v_i and v_j if v_j occurred immediately after v_i in the logged data, represented as (v_i, v_j) .
- w : the total occurrences of (v_i, v_j) in the event stream.
- A vertex v_i is marked as usability problem candidate according to the following heuristic: if $d(v_i) > mean(d(v_j)) + 2 stdev(d(v_j))$, for v_j representing all outgoing vertices of v_i . The intuition behind this is to identify cyclic actions, indicating repeated attempts of performing a task or using UI elements.
- Considering time differences, nodes are also marked as usability problems candidates if any of the following is true:
 1. $t(v_i) > mean(t(v_j)) + 2 stdev(t(v_j))$; or
 2. $t(v_i) - t(v_j) > 10 seconds$.
 Where v_j represents all outgoing vertices of v_i . The 10 seconds limit follows Nielsen’s 3 Important Limits, which presents that 10 seconds is about the time limit for users to keep attention on the task at hand.

Figure 1 presents an example of the usage graph of one of the sessions analyzed and how the usability problems candidates are pointed out by the heuristic used. In the figure the ellipses represent UI events; boxes represent UI elements. This example shows how cyclic actions impact in the distances (d) and how usability problem candidates are pointed out in highlighted ellipses.

C. Data analysis

As mentioned previously, data capture was performed via WELFIT [22]. Then, raw logs were downloaded from the tool and usage graphs were built following the algorithm presented in [20]. Usage graphs were built for each of the sessions in separate DOT files. DOT file is a graph representation format used by Graphviz² software. Graphviz was used to generate visualizations of usage graphs as Figure 1, Figure 2 (b), Figure 3 (b), and Figure 4 (b). In addition, DOT files were also used as input for computing metrics related to diameter, centrality, degree, community detection, among others, via Gephi³ software. Finally, for each session, the eccentricity distribution was analyzed and the main characteristics of the distributions were summarized as:

- The presence of peaks in the eccentricity distribution;
- Number of peaks;

The eccentricity of vertex v in a connected graph G is the maximum graph distance between v and any other vertex u of G [27]. In the eccentricity distribution, a peak is considered a point in the distribution, say x , with a respective count value $f(x)$ surrounded by $x-1$ and $x+1$, so that $f(x-1) < f(x)$ and $f(x) > f(x+1)$.

The next examples show how the EDP supports insights in relation to task deviations. Moreover, it also allows the comparison of large datasets of detailed actions, supporting the understanding of how users performed tasks and when in the session deviations/cyclic actions occurred.

² <http://graphviz.org/>

³ <http://gephi.org/>

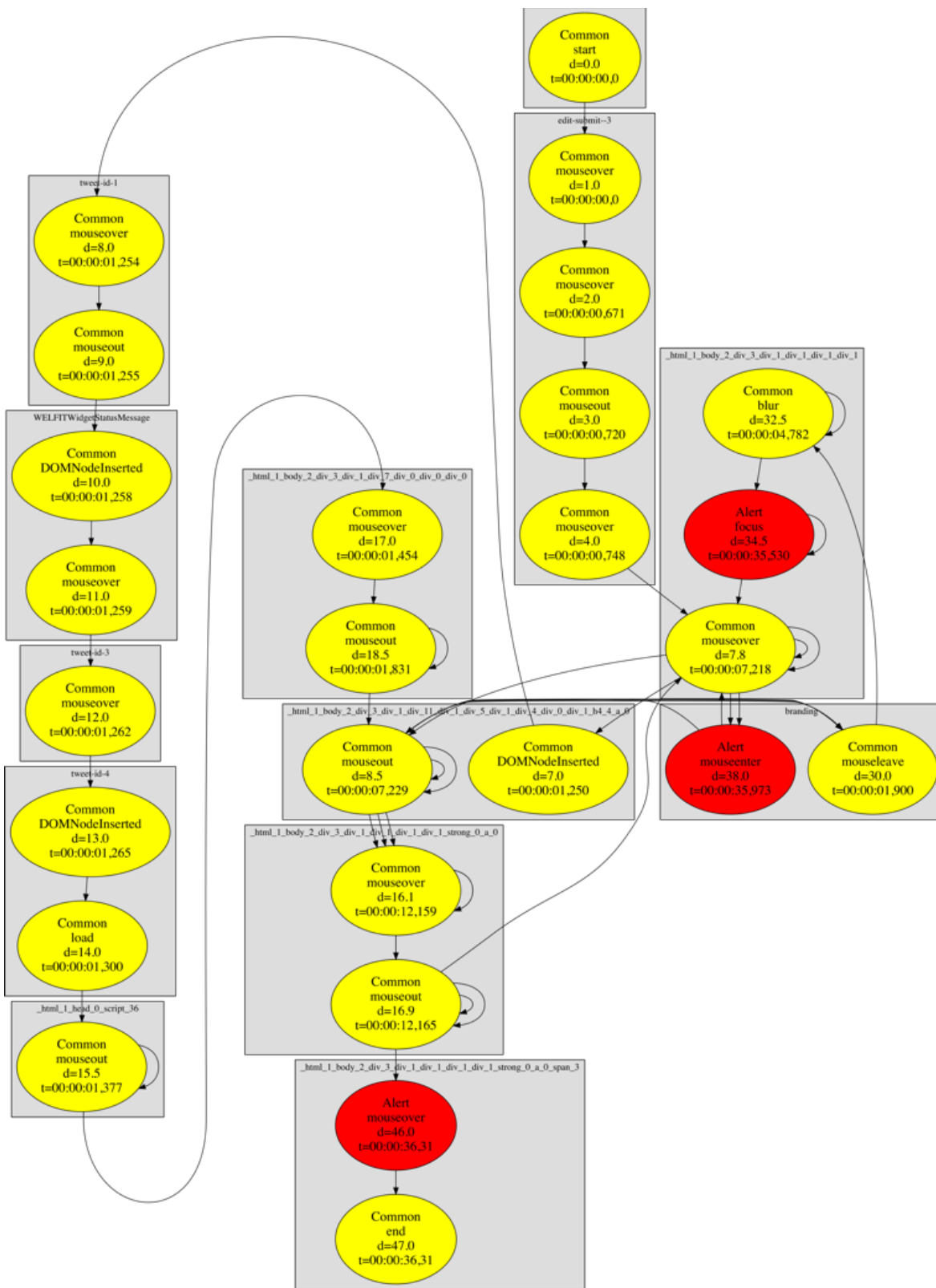


Fig. 1. Example of the usage graph of one of the analyzed sessions; highlighted nodes are the usability problems candidates pointed out by the heuristic.

First, consider a simple event stream resulting in the following usage graph vertices and edges, respectively:

$$E = \{A, B, C, D, E\};$$

$$V = \{(A, B), (B, C), (C, D), (D, E)\}.$$

Now consider the eccentricity distribution for the resulting graph (Figure 2 (a) presents the graph and Figure 2 (b) the eccentricity distribution). Note that when analyzing task completion, Figure 2 represents a desirable performance, without deviations and without cyclic actions before reaching the end of the usage graph.

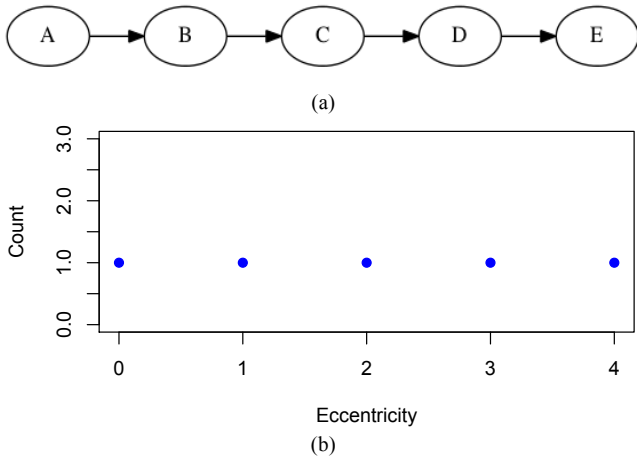


Fig. 2. Simple graph (a) and its corresponding eccentricity distribution (b).

Now consider an event stream with a repeated action (cycle) in a certain UI element, resulting in the following usage graph vertices and edges:

$$V = \{A, B, C, D, E\};$$

$$E = \{(A, B), (B, C), (C, D), (D, B), (B, C), (C, D), (D, E)\}.$$

Moreover, consider the eccentricity distribution for the resulting graph after considering that a task deviation occurred (Figure 3). Note that the peak represented in the Figure 3 indicates that a deviation occurred in some of the nodes with eccentricity equal to the value indicated in the peak, in this case, nodes C and D have eccentricity equal to 2.

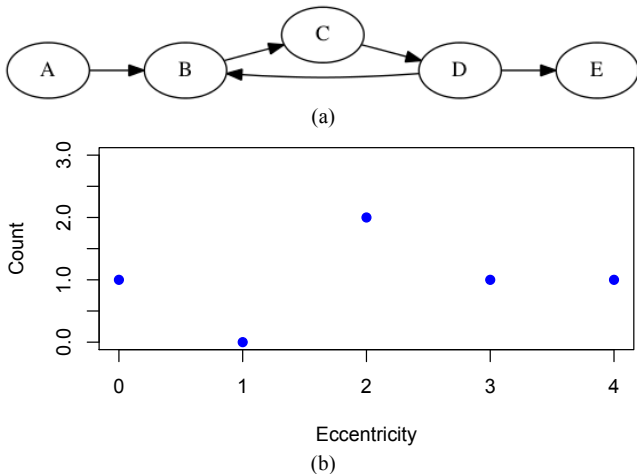


Fig. 3. Simple graph with a cyclic action (a) and its corresponding eccentricity distribution (b); the point (2,2) represents a peak in the distribution.

Now consider a simple graph showing a task deviation at the beginning of the task and then actions leading to task conclusion. In [21], one motif representing this type of task deviation is presented in Figure 4 (a). Building on top of this result, the eccentricity distribution for the motif is presented in Figure 4 (b), summarizing the same concept of task deviation. Note that the EDP represents inversely when, in the session, the deviation occurred. Hence, Figure 4 (b) presents that the deviation occurred in the first quarter of the session.

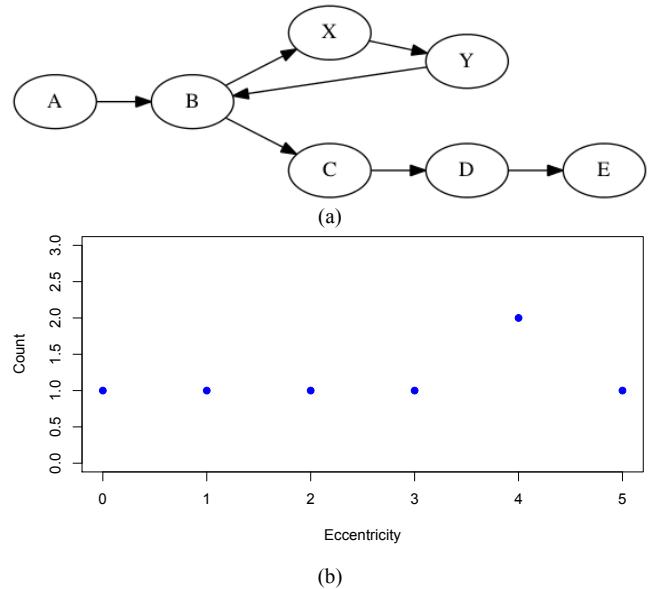


Fig. 4. (a) Motif identified representing a task deviation and (b) its corresponding eccentricity distribution.

Once that graph metrics were calculated, correlations were computed and the eccentricity distributions were analyzed, highlighting deviations from task and how the summarized results can provide details of how users performed tasks.

The information regarding the number of peaks was used to cluster sessions in order to point out tasks that users faced difficulties. After clustering sessions based on the number of peaks, each of the sessions was analyzed in order to identify the tasks they relate to. The rationale here is to cluster sessions that count on similar number of tasks deviations so that similar distributions will correlate similar behaviors related to task deviations across the evaluated website. Then, each task was evaluated considering the presence in the clusters containing different number of EDPs and how these peaks change for a specific task over time. Finally, task learning curves were analyzed in order to reveal patterns associated to certain types of tasks, aiming the analysis of tasks that need to be redesigned, restructured, or simplified.

IV. RESULTS

Table 1 presents a summary of descriptive information involving the 427 usage graphs. The presented information was generated for each usage graph through WELFIT and Gephi software. It is possible to see that the high standard deviation values are related to the multiplicity of tasks, i.e., some tasks resulting in small usage graphs with few tens of vertices, while other sessions resulted in usage graphs with few thousand vertices. This effect can also be seen in the number of shortest paths, vertices, edges, among others. On the other hand, the eccentricity distribution is proposed as a more valuable metric, highlighting deviations and providing a richer semantic result than a sole number, e.g., task deviations occurred mostly during the first quarter of the sessions.

TABLE I. SUMMARY OF THE 427 USAGE GRAPHS ANALYZED.

Metric	Mean	S.D.
Log lines	565.37	2,198.29
Vertices	89.25	147.30
Edges	566.37	2,198.29
Degree	2.03	0.54
Weighted degree	3.65	2.69
Diameter	27.76	12.91
Path length	9.58	4.53
Shortest paths	28,587	238,373
Density	0.07	0.08
Modularity	0.66	0.52
Communities	7.14	2.80
Weakly connected components	1.07	1.15
Strongly connected components	13.18	10.09
Average clustering coefficient	0.05	0.22
Page views	2.21	1.35
Usability problem candidates	19.29	36.54
Eccentricity distribution peaks	1.75	1.22

Considering correlations, Spearman test (ρ) was applied in order to find significant correlations between EDPs and other metrics computed from usage graphs. Spearman test is a nonparametric way of evaluating correlations between two variables. In this study we used R software to perform correlation analysis. Next, we present metrics with significant positive correlation with the number of EDPs:

- Average path length ($\rho = 0.618$, $p\text{-value} < 0.001$);
- Modularity ($\rho = 0.601$, $p\text{-value} < 0.001$);
- Diameter ($\rho = 0.595$, $p\text{-value} < 0.001$);
- Number of shortest paths ($\rho = 0.413$, $p\text{-value} < 0.001$);
- Number of communities ($\rho = 0.380$, $p\text{-value} < 0.001$);
- Number of usability problem candidates ($\rho = 0.354$, $p\text{-value} < 0.001$).

Figure 5 shows the heat map of all eccentricity distributions and provides a high level overview on how tasks were performed in the studied website. It shows that task deviations usually occur in the first quarter of the session. Recall that the EDP represents inversely when in the session the deviation occurred.

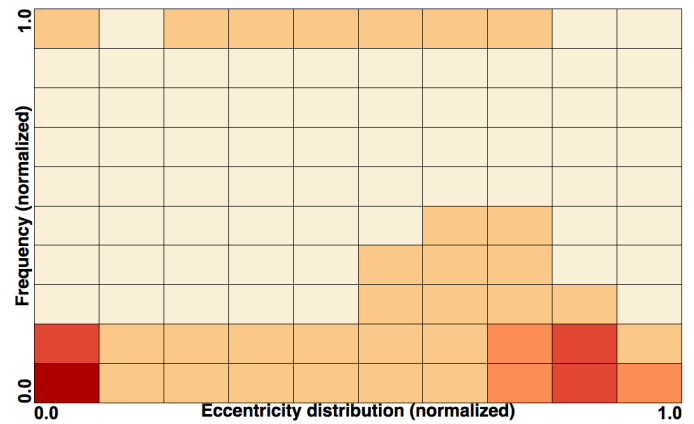


Fig. 5. Eccentricity distribution heat map for all the studied sessions; eccentricity and frequency are normalized.

In order to relate tasks and eccentricity distribution, each of the event streams was analyzed in detail to identify the tasks the users were performing, then clusters were generated based on the number of EDPs and tasks performed, allowing the analysis of tasks that users faced difficulties.

Fig. 6 shows tasks occurrence in the clusters generated by considering the number of peaks in the eccentricity distribution. It points that, in absolute numbers, sessions in the cluster of 5 peaks require detailed analysis, more specifically how users performed tasks 3 (View content), 4 (View accessibility evaluation form sample), 5 (View the “about page” presentation), 6 (View heuristic evaluation form sample), and 7 (View topics index). The presence of the same task in different clusters considering EDP shows that the same task is performed considering different possible paths present in the event streams, which paves the path for measuring task learning curve considering the presence of peaks in the eccentricity distributions. Moreover, in huge datasets it is also possible to identify the minimum number of EDPs for a certain task, for instance, tasks 10 and 14. Thus, the eccentricity distribution used to measure task learning curve can be considered to summarize a huge number of sessions that represent detailed interaction data. Moreover, Fig. 6 shows that the tasks that are present in the sessions with 4 and 5 peaks are not the most common ones, highlighting that these tasks need to be reviewed and that related UI components might need improvement.

Figure 7 shows the all 17 learning curves resulting from the analysis of EDPs in the dataset. It is possible to see that most tasks (9 out of 17) count on distributions with highest frequencies in the 1-peak mark, which is expected given that the website does not require login for most of identified tasks and most of the tasks identified are related to simply viewing different types of content. In addition, tasks with isolated peaks often counted on fewer observations. Task 4 (View accessibility evaluation form sample) has a high frequency on cluster of 5 EDPs, indicating that this task should be analyzed in detail to verify what happened and how it can be simplified. Task 7 (View topics index) and task 11 (Access the administration page) have high frequency on the cluster of with 3 EDPs, which indicates that these tasks require further analysis aiming at simplifying, causing them to appear in the

clusters containing eccentricity distribution with less than 3 peaks. No specific task was identified with 0 EDP, indicating that, in this dataset, most tasks identified have a minimum of 1 EDP.

The following analysis involved grouping tasks with similar learning curves. Task 13 (View a comment), task 15 (Create a heuristic evaluation), and task 17 (View the "about page") represent the fastest learning curves in this dataset (Figure 8). There is no occurrence of sessions with 5 and 4 EDPs, a less frequent occurrence of sessions with 3 and 2 EDPs, and predominant occurrence of sessions with 1 EDP.

Figure 9 shows a group of tasks with slightly different shape. They count on few instances with 4 and 3 EDPs. Although, most of observations of these tasks resulted in eccentricity distributions with 1 or 2 EDPs.

Figure 10 shows a group of learning curves with a valley-like shape, with important number of occurrences of sessions with 5, 4, and 3 EDPs; however, observations most commons related to these tasks convey eccentricity distributions with 1 peak. Finally, Figures 11, 12, and 13, show the groups of learning curves presenting observations with worst cases for eccentricity distributions in the dataset. Sessions with 1 EDP are less frequent in these groups and curves are shifted towards the 3, 4, or 5-peak marks.

A. Discussion

The correlations found reveal that the EDPs relate to different topology metrics. The average path length and diameter are impacted as more task deviations occur, although these graph-wide metrics represent less detailed information regarding performance than eccentricity distributions. Modularity and community detection were not as correlated with EDPs, probably due to the way the directed graph is built from detailed interaction data.

Bearing in mind different types of learning curves found, the proposed method can be used to support redesign of UI and user tasks. The proposed method of measuring task learning curves can be used to identify different ways users perform the same task. Moreover, it is possible to identify how EDPs change over time and the empirical minimum/maximum values reached by real users. In these cases, values differing from

expected would reveal situations unforeseen during project phases and reinforcing approaches involving real users, in real environments, performing real tasks.

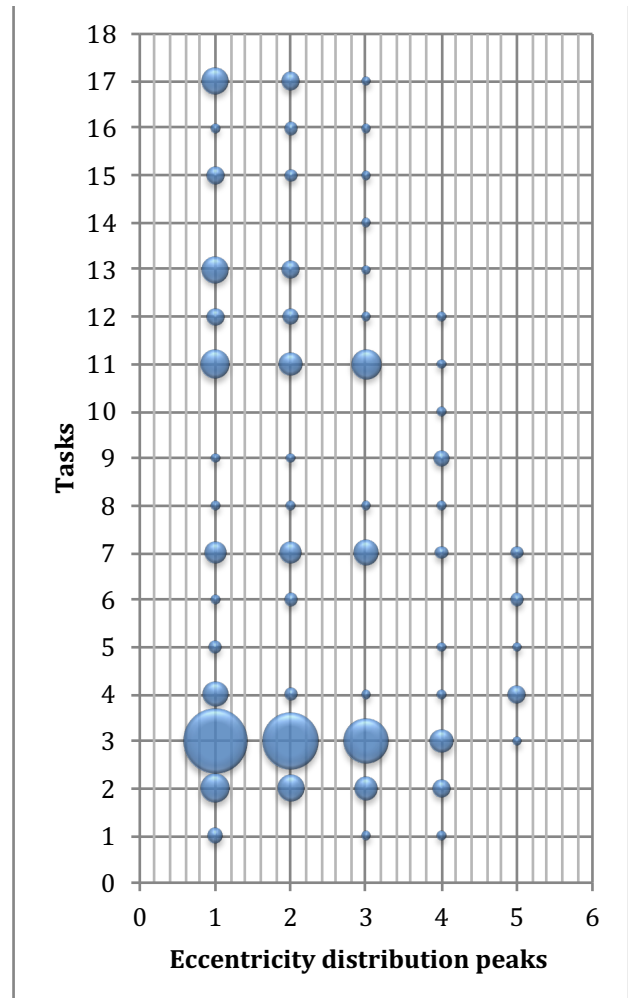


Fig. 6. Tasks presence in the clusters of sessions built based on the number of EDPs; size of bubbles represents the number of occurrences found in the sessions.

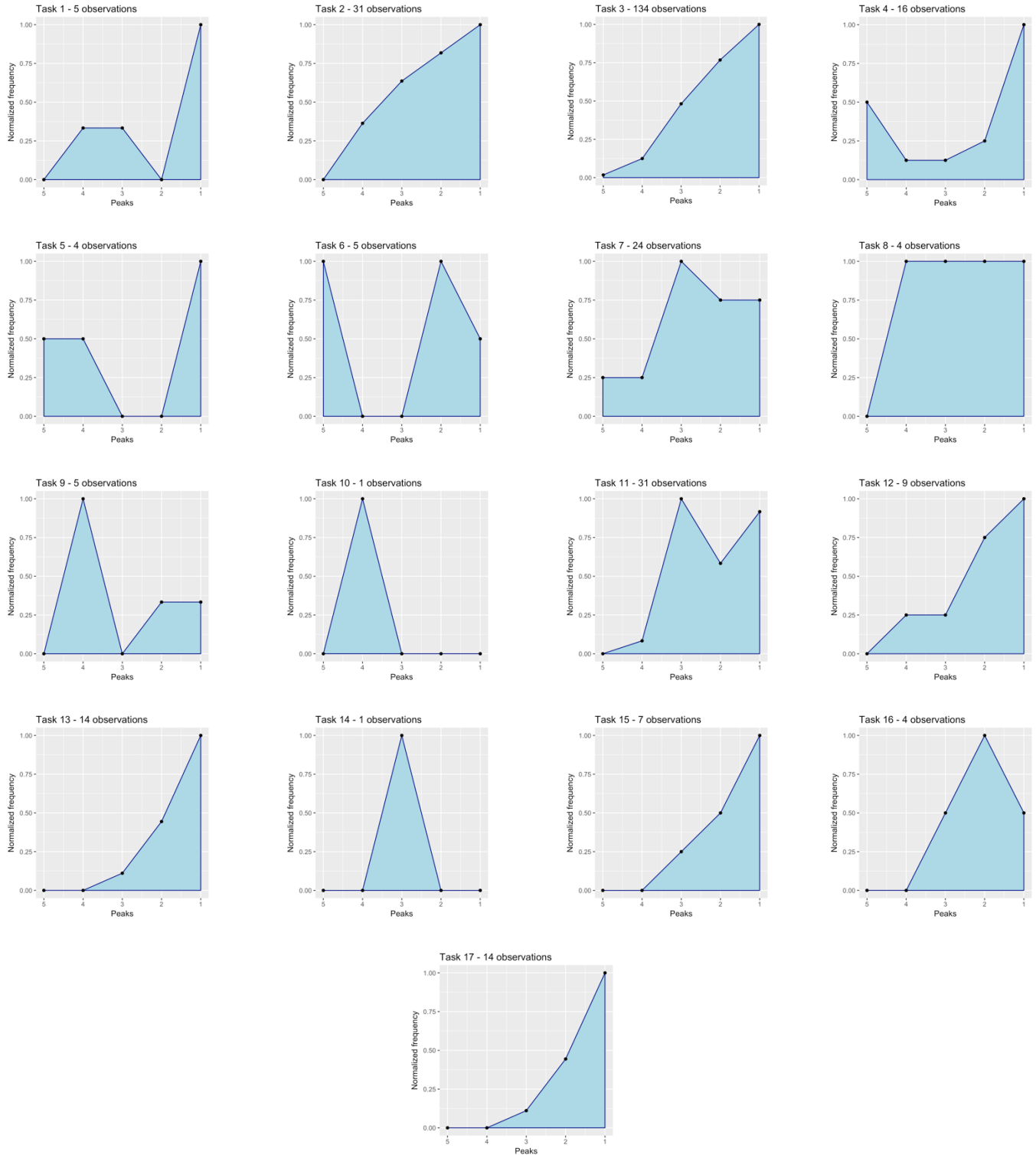


Fig. 7. Task learning curves computed for each of the tasks. The x-axis represents the presence of the task in the cluster of the respective number of EDPs. The y-axis shows the normalized frequency for the respective task in the clusters.

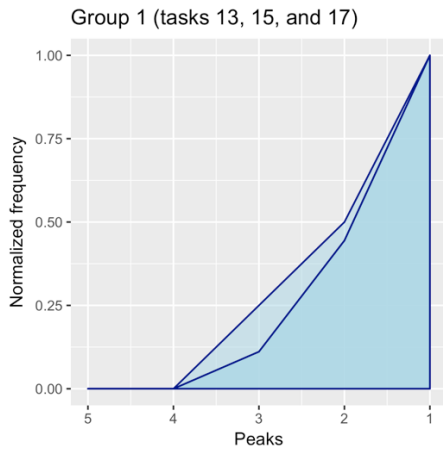


Fig. 8. Group of tasks with fastest learning curves. Tasks 13, 15, and 17 are, respectively: View a comment; Create a heuristic evaluation; View the "about page". This group sums up to 35 observations.

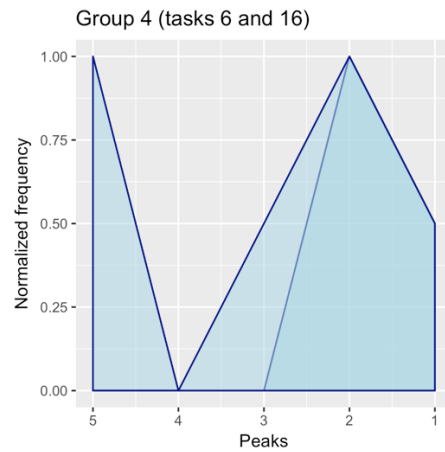


Fig. 11. Group combining learning curve for tasks 6 (View heuristic evaluation form sample) and 16 (Register user). This group sums up to 9 observations.

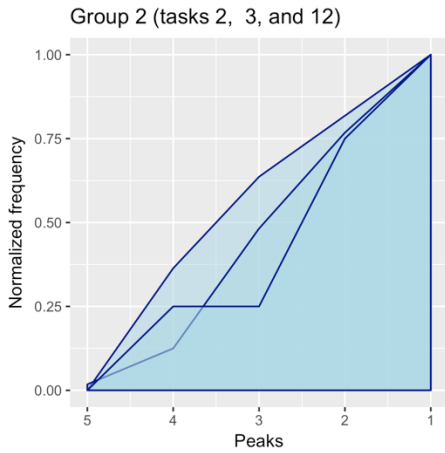


Fig. 9. Group combining learning curve for tasks 2 (Search for or filter topics), 3 (View content), and 12 (Reset password). This group sums up to 174 observations.

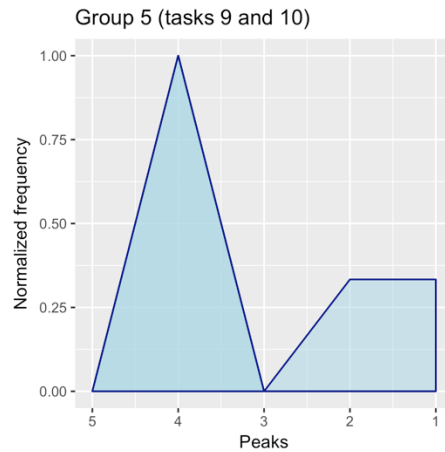


Fig. 12. Group combining learning curve for tasks 9 (View publications page) and 10 (View references page). This group sums up to 9 observations.

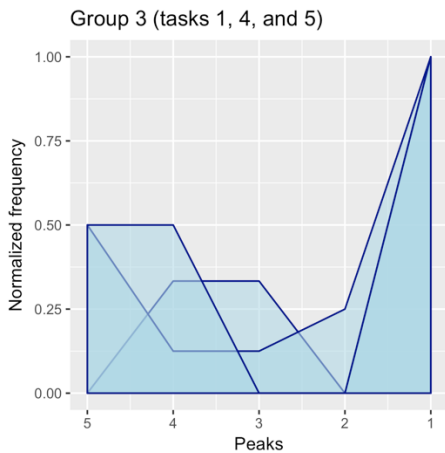


Fig. 10. Group combining learning curve for tasks 1 (Perform login), 4 (View accessibility evaluation form sample), and 5 (View the "about page" presentation). This group sums up to 25 observations.

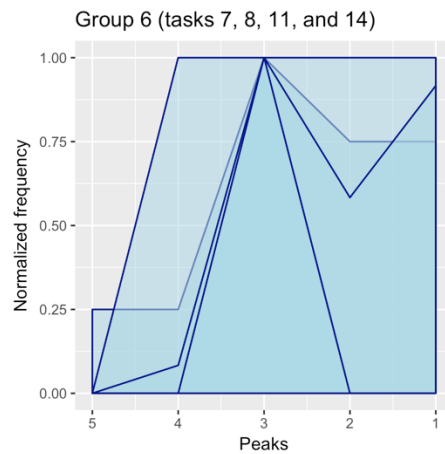


Fig. 13. Group combining learning curve for tasks 7 (View topics index), 8 (Create an accessibility evaluation), 11 (Access the administration page), and 14 (Delete content). This group sums up to 70 observations.

V. CONCLUSIONS

This work presented outcomes gathered from an investigation on how to measure task learning curves from detailed interaction logs captured while users performed real tasks. The proposed way of measuring task learning curves builds on top of modeling task deviations as eccentricity distributions peaks (EDPs). Hence, from the analysis of how frequent these deviations are for groups of tasks, it is possible to evaluate big datasets and to identify tasks that appear in sessions with higher/lower occurrence of EDPs. Thus, this approach supports the identification of the most/least complicated tasks based on task learning curves. This can be used to represent learning curve for the population of users, supporting the analysis of how task deviations decay over time.

The proposed way of measuring learning curve supports summarization of multiple sessions represented by event streams of highly detailed interaction data, allowing HCI practitioners to select groups of tasks, sessions, or groups of users related to the high occurrences of EDPs. Regarding how smooth the tasks were performed, the task learning curves based on EDPs are an interesting proxy, since the lesser the number of deviations from tasks, the smoother the eccentricity distribution will be. Regarding learning curves, the smoother the eccentricity distribution is, the closer to the 0 or 1-peak marks the resulting learning curve will be.

The proposed way of measuring task learning curves can be used by HCI practitioners and Data Scientists on multiple cases. For instance, in A/B tests comparing two solutions or in usability tests as a quantitative way of measuring learnability, memorability, or error rate. In the case of automated tools, the proposed way of measuring learning curves can be used in dashboards from whole systems to UIs elements in order to identify situations where users are facing difficulties and thus to offer online support, e.g., consider a shopping cart page of an e-commerce website.

This work is part of a long-term initiative to build a usage behavior model based on detailed logged data, identifying how to identify tasks, model task deviation, and measure learning curves. Regarding limitations of this work, the dataset considered is about a technical website. The target audience is composed of developers, content producers, and digital designers. Thus, it does not represent all the Web nor the whole Web audience. Although, the focus of this paper is to present how the eccentricity distribution supports depicting task learning curves. Another point to consider as a limitation is that from 220,448 sessions occurred in the last two years, only 427 (0.19%) were logged. Besides representing a small part of the users of the studied website, this occurred due to the need for users to accept participating in the study, allowing the data logger do capture detailed interaction data. Thus, the number of participants was impacted in favor of privacy and users' choice on providing or not detailed data related on how they perform tasks. Moreover, this is a requirement of the tool used.

Future work involves differentiating UI learning curves from content learning curves based on EDPs and interaction log analysis.

REFERENCES

- [1] Arroyo, E., Selker, T., Wei, W. 2006. Usability tool for analysis of web designs using mouse tracks. Proc. of ACM CHI 2006 Conf. on Human Factors in Computing Systems, 2 of Work-in-progress, 484-489.
- [2] Atterer, R., Schmidt, A. 2007. Tracking the Interaction of Users with AJAX Applications for Usability Testing. Proc. of ACM SIGCHI 2007 Conf. on Human Factors in Computing Systems, 1347-1350.
- [3] Beckert, B. and Wagner, M. 2009. Probabilistic models for the verification of human-computer interaction. In Proceedings of the 32nd annual German conference on Advances in artificial intelligence (KI'09). Springer-Verlag, Berlin, Heidelberg, 687-694.
- [4] Blondel, V. D., Guillaume, J.L., Lambiotte, R. & Lefebvre, E. 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment.
- [5] Cadez, I. V., Heckerman, D., Smyth, P., Meek, C., White, S. 2003. Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. Data Mining and Knowledge Discovery.
- [6] Carta, T., Paternò, F., Santana, V.F. 2011 Web Usability Probe: A Tool for Supporting Remote Usability Evaluation of Web Sites. Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT'11) - Volume Part IV (2011).
- [7] Chi, E. H., Rosien, A., and Heer, J. 2002. Lumberjack: Intelligent discovery and analysis of web user traffic composition. WEBKDD, LNCS, 2703, 1--16, Springer.
- [8] Cooley, R., Tan, P. N., and Srivastava, J. 2000. Discovery of interesting usage patterns from Web data. LNCS, vol. 1836, 163-182, Springer.
- [9] Daoud, M., Lechani, L., and Boughanem, M. 2009. Towards a graph-based user profile modeling for a session-based personalized search. Knowl. Inf. Syst. 21, 3. <http://dx.doi.org/10.1007/s10115-009-0232-0>
- [10] Donato, D. 2010. Graph structures and algorithms for query-log analysis. In Proceedings of the Programs, proofs, process and 6th international conference on Computability in Europe (CiE'10), Springer-Verlag, Berlin, Heidelberg, 126-131.
- [11] Domenech, J. M., Lorenzo, J. 2007. A tool for web usage mining. IDEAL, LNCS 4881, 695-704. Springer.
- [12] John, B. E. and Kieras, D. E. 1996. The GOMS family of user interface analysis tech-niques: comparison and contrast. ACM Trans. Comput. Hum. Interact. 3, 4, 320-351. <http://doi.acm.org/10.1145/235833.236054>
- [13] ISO - International Standard Organization. 1998. ISO-9241 - Ergonomic requirements for office work with display terminals (VDTs), Part 11: Guidance on usability.
- [14] Lehmann, J., Lalmas, M., Yom-Tov, E., Dupret, G. 2012. Models of user engagement. In Proceedings of the 20th international conference on User Modeling, Adaptation, and Personalization (UMAP'12). Springer-Verlag, Berlin, Heidelberg, 164-175. http://dx.doi.org/10.1007/978-3-642-31454-4_14
- [15] Lewis, C., Rieman, J. 1993. Task-centered user interface design - A practical introduction. University of Colorado, Boulder.
- [16] Mendes, P. N., Mika, P., Zaragoza, H., and Blanco, R. 2012. Measuring website similarity using an entity-aware click graph. In Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12). ACM, New York, NY, USA, 1697-1701. <http://doi.acm.org/10.1145/2396761.2398500>
- [17] Nielsen, J. Usability 101: Introduction to Usability. Jakob Nielsen's Alertbox. (2012) <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [18] Paternò F. 2003. ConcurTaskTrees: An Engineered Notation for Task Models. The Hand-book of Task Analysis for Human-Computer Interaction. pp. 483-503. Mahwah: Lawrence Erlbaum Associates.
- [19] Paternò, F., Santoro, C. 2008. Remote Usability Evaluation: Discussion of a General Framework and Experiences from Research with a Specific Tool. Maturing Usability. 197-221. Springer-Verlag.
- [20] Santana, V. F., Baranauskas, M. C. C. 2010. Summarizing observational client-side data to reveal web usage patterns. Proceedings of the 2010 ACM Symposium on Applied Computing (ACM SAC 2010).

- [21] Santana, V. F., Paula, R. A., Pinhanez, C. S. 2014 Insights on User Actions from Log-based Graph Topology. Workshop on Log-based Personalization (4th WSCD Workshop). Proceedings of 7th ACM WSDM Conference (WSDM2014), New York.
- [22] Santana, V. F., Baranauskas, M. C. C. 2015. WELFIT: A remote evaluation tool for identifying Web usage patterns through client-side logging. *International Journal of Human-Computer Studies*, 76, 40-49.
- [23] Spiliopoulou, M., and Faulstich, L. 1998. WUM - A Tool for WWW Utilization Analysis. In *Selected papers from the International Workshop on The World Wide Web and Data-bases (WebDB '98)*. Springer-Verlag, London, UK, UK, 184-103.
- [24] Vasconcelos, L. G., Baldochi Júnior, L. A. 2011. USABILICS: avaliação remota de usabilidade e métricas baseadas na análise de tarefas. In *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction (IHC+CLHC '11)*. Brazilian Computer Society, Porto Alegre, Brazil, Brazil, 303-312.
- [25] Vargas, A., Weffers, H., Rocha, H. V. 2010. A method for remote and semi-automatic usability evaluation of web-based applications through users behavior analysis. In *MB '10 Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*. ACM New York, NY, USA.
- [26] Waterson, S. J., Hong, J. I., Sohn, T., Landay, J. A., Heer, J., and Matthews, T. 2002. What did they do? Understanding clickstreams with the WebQuilt visualization system. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '02)*. ACM, New York, NY, USA, 94-102. DOI=10.1145/1556262.1556276 <http://doi.acm.org/10.1145/1556262.1556276>
- [27] Weisstein, Eric W. Graph Eccentricity. From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/GraphEccentricity.html>.