One Metric for All: Calculating Interaction Effort of Individual Widgets

Julián Grigera
LIFIA, Fac. de Informática, Univ. Nac. La Plata
& CICPBA. La Plata, CP 1900, Argentina
julian.grigera@lifia.info.unlp.edu.ar

Juan Cruz Gardey
LIFIA, Fac. de Informática, Univ. Nac. La Plata
La Plata, CP 1900, Argentina
jcgardey@lifia.info.unlp.edu.ar

Andrés Rodriguez
LIFIA, Fac. de Informática, Univ. Nac. La Plata
La Plata, CP 1900, Argentina
arodrig@lifia.info.unlp.edu.ar

Alejandra Garrido
Gustavo Rossi
LIFIA, Fac. de Informática, Univ. Nac. La Plata
& CONICET. La Plata, CP 1900, Argentina
garrido@lifia.info.unlp.edu.ar
gustavo@lifia.info.unlp.edu.ar

ABSTRACT

Automating usability diagnose and repair can be a powerful assistance to usability experts and even less knowledgeable developers. To accomplish this goal, evaluating user interaction automatically is crucial, and it has been broadly explored. However, most works focus in long interaction sessions, which makes it difficult to tell how individual interface components influence usability. In contrast, this work aims to compare how different widgets perform for the same task, in the context of evaluating alternative designs for small components, implemented as *refactorings*. For this purpose, we propose a unified score to compare the widgets involved in each refactoring by the *level of effort* required by users to interact with them. This score is based on micro-measures automatically captured from interaction logs, so it can be automatically predicted. We show the results of predicting such score using a decision tree.

CHI'19 Extended Abstracts, May 4-9, 2019, Glasgow, Scotland, UK.

ACM ISBN 978-1-4503-5971-9/19/05.
DOI: https://doi.org/10.1145/3290607.XXXXXXX

CCS CONCEPTS

• **Human-centered computing → HCI design and evaluation methods**
• **Human-centered computing → Web-based interaction**

KEYWORDS

Web usability; interactivity; usability refactoring; A/B testing; user interaction metrics

ACM Reference format:
Julián Grigera, Juan Cruz Gardey, Andrés Rodriguez, Alejandra Garrido, Gustavo Rossi. 2019. ~ One Metric for All: Calculating Interaction Effort of Individual Widgets. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI'19 Extended Abstracts), May 4–9, 2019, Glagsow, Scotland, UK.* ACM, New York, NY, USA. 6 pages. https://doi.org/10.1145/3290607.XXXXXXX

1 INTRODUCTION

While addressing web usability has proven to provide a large value to customers, it still does not get enough resources like other core practices in the development process [6]. To cut down the cost of usability evaluation and repair, many automatic tools have emerged [2], as well as processes that do not depend so much on usability experts [7]. However, while most of these tools may diagnose general usability issues, fewer approaches focus on more localized interaction problems, which we consider to be more tractable, cheaper and easier to automate. This also enables incremental improvement and thus, the combination of agile and UCD approaches [6].

In previous works we have developed tools to automate the detection of usability problems of user interaction in web interfaces that we catalogued as *usability smells* [3], and to suggest and apply fixes to those smells in the form of *Client-Side Web Refactorings (CSWRs)* [4]. CSWRs perform small changes over one or a small set of widgets at the client-side with the purpose of improving usability while preserving functionality. Table 1 provides some examples and shows that there may be more than one CSWR that solves a given smell, i.e., alternative combination of widgets that allow users to fulfill the same task by different ways of interaction.

In this context, our goal is to automatically evaluate the effectiveness of alternative CSWRs, by comparing the behavior of final users in the wild without constrained tasks or lab conditions. This goal drove us to find a unified score that can be assigned to any type of widget based on its usage, and that allows to compare CSWRs, much as conversion-rate works for A/B testing. Our proposal for this unified score is the level of effort required by users to interact with a widget; we call it *interaction effort*. We based this score on the notion of interactivity proposed by Janlert and Stolterman [5]. According to this work, it makes sense to measure interactivity over individual artefacts, separately from the overall combined interactivity of an environment or context. Moreover, the dimensions of interactivity (e.g. time expenditure, interaction pressure) correspond to the idea of interaction cost, known as a direct measure of usability [1].

The interaction effort metric is based on micro-measures that can be automatically captured from user interaction events. The micro-measures involved were selected during a preliminary study where UX experts scored real user behavior on specific widgets and reported the interaction events that impacted in their decision. After that, a decision tree was created to predict the interaction effort of individual widgets as perceived by a human rater in a specific "interactive situation" [5]. As a result, we have an automated way to compare CSWRs, averaging for all users, the level of interaction effort of the widgets that each one involves. This work presents a first approach applied in two widgets: **text inputs** and **selects** (drop down lists). Since text input widgets can be refactored into selects (e.g., when the accepted values belong in a narrow set), this gives us the chance of comparing the baseline design against the refactored one.

The first results we obtained from our experimentation with end users suggest that it is possible to automatically predict this metric, relying only on mouse and keyboard interaction data. Moreover, this work may contribute to a broader field outside refactoring, since it allows obtaining interaction effort scores on individual widgets not necessarily generated with a CSWR. Finally,

Table 1: Usability Smells & Refactorings

Usability Smell	*CSWR*
Free Input for Limited Values	Add Autocomplete
	Text Field into Select
	Text Field into Radio Buttons Set
Unformatted Date Input	Add Date Picker
	Date Input into Select
Undescriptive Element	Rename Element
	Add Tooltip
	Change Font
Unresponsive Element	Turn Attribute into Link
	Add Tooltip

used in the context of agile processes, the automatic assessment of interactivity may considerably speed up the analysis of alternative designs for the same task.

2 RELATED WORK

Our work is related with different endeavors. On the one side, we try to measure the user interaction effort in the wild. On the other, to automate the process of usability evaluation and repair. Janlert and Stolterman define interactivity as *the activity of interacting*, considering *activity* as an ongoing process [5]. It can be measured in several dimensions, in particular on the basis of time expenditure, the pace or frequency of interaction, or the pressure of the interaction (number of actions or operations per unit of time). Moreover, the authors propose to first consider artifacts and systems in isolation, apart from other artifacts.

Different tools for automatic UI analysis have been presented, e.g., AIM [8]. AIM uses empirically validated models and metrics to evaluate four aspects of an interface: color perception, perceptual fluency, visual guidance and accessibility. After an automatic analysis of the web interface, it reports metrics for designers' use. Among other goals, Oulasvirta et al. aim to give designers and developers some automatic tools to improve their processes. While we share this goal, our focus is on the interactivity, the dynamic process of interaction with the interface and its elements, rather than on the static issues related to visual clutterness, colors, etc.

Speicher et al. [9] present a tool to support usability-based split testing with an approach partially similar to ours. Like us, they follow a component-based approach to get metrics by an automatic tracking and analysis of user interactions. However, those automatic metrics are mapped to predefined usability heuristics to achieve a global score on the pages under analysis, unlike our individual widget score. Moreover, they focus on Search Engine Results Pages while in this work we focus on form widgets. Finally, they have found heuristics to be dependable of the user context (like screen size and user intention) while we did not find this limitation during the experiment, presumably because we focus on small designs rather than entire pages. A similar work, W3Touch [7], also gathers component-level metrics and proposes improvements in touch interfaces of mobile devices. Besides the difference in the kind of interaction and device, our work is aimed at rating and comparing widgets, instead of finding specific problems in responsiveness.

3 METHOD

In this paper we propose a unified metric to evaluate single interface widgets by its *interaction effort*, i.e., the level of effort required by the user to interact with a single widget, as perceived by an expert. This would allow for comparing alternative widgets that serve the same purpose. We aim at a fully automated rating process, so the metric is based on measures (micro behaviors) that can be automatically captured. Given an interaction sample over a single widget, we should be able to feed the captured measures into a process capable of instantly predicting the effort rating.

Table 2: Selected Micro-Measures

Name	*Description*
	Text Input
Typing latency	Time from focus to start typing
Total Typing Time	Time from first keypress to last keypress
Total Time	Time from focus to blur
Typing Speed	Total typing time in proportion to number of chars typed
Typing Speed Variance	Intra-keypress time variance
Corrections	Number of deleted characters
	Select
Clicks	Number of clicks on the widget, except those within the open list.
Keystrokes	Number of keystrokes, except within the open list.
Focus Time	Total focus time.
Option Changes	Number of times the selection is changed
Options Display Time	Total time the options list is open

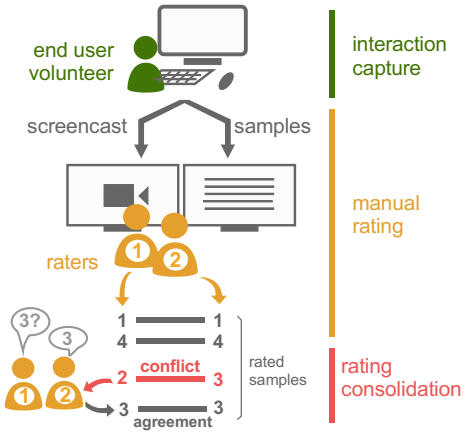

Figure 1: Process for capturing training data.

3.1 Obtaining the Relevant Micro Measures

The first step for designing the metric was to find the measures that compose it, i.e., the behavior traits that an UX expert considers relevant at the time of judging the effort that is being demanded from a user. We recorded and studied a first set of interaction samples with real users on 2 types of widgets: text input and select. We then asked 3 UX experts to rate each sample with a number between 1 and 4. This scale was chosen to avoid a neutral value - i.e., in its simplest form, an interaction can be either *effortless* (1 / 2) or *demanding* (3 / 4). After the rating task, we asked the experts to indicate what aspects of user behavior they had considered relevant for the chosen rating, so we could use them as candidate micro-measures. We started with a suggested set of micro-measures, but the experts were encouraged to add new ones, or remove the ones they deemed irrelevant. This way we obtained a consolidated list of micro-measures that could be automatically captured from mouse & keyboard interaction. Since each widget involves different kinds of interactions, there is a different set of micro-measures for each type of widget. The measures for each type of widget are listed in Table 2. It is worth to note that at the time of implementing the client-side script that captures the measures, we had to settle on a variation to the initial proposal with respect to the **select** widget, because we were not able to capture any events over an opened drop-down list. Experts agreed on using the measure of **Focus Time** measure instead.

3.2 Experiment

Once the client-side script was able to capture all the selected micro-measures, we ran several sessions with end users. We recruited 23 volunteers (13 m / 10 f, ages $\bar{x}$=42,56 s=19,5) with diverse expertise on the use of web applications, and asked each one of them to fill 3 to 5 forms from websites of different domains, like plane tickets, governmental paperwork or e-commerce account registration, simulating the form submissions behavior when necessary, so the volunteers could enter their own data without real consequences. We recorded each session as a screencast, capturing at the same time each interaction sample with their associated micro-measures. After the sessions, 4 UX experts used a custom web tool to rate each sample while watching the screencasts, in such a way that each sample got 2 independent ratings. Since the original intra-rater agreement was moderate (average Weighted Cohen's Kappa was 0.549 for text inputs and 0.57 for selects), all diverging ratings were consolidated by both raters in a second round. The 4 raters covered all the samples, paired in 2 different ways to mitigate bias. Using this procedure, we obtained a final set of 404 interaction samples for text inputs and 148 for selects, all of them with consolidated ratings. The process for capturing the rated samples is outlined in Fig. 1.

With the obtained datasets, we developed a decision tree classifier for each widget type in order to automatically predict the level of effort (from 1 to 4) for a given sample. In both cases, the dataset was split using 70% for training the decision tree and the remaining 30% for testing it. We decided to use a decision tree classifier at this early stage because it clearly shows how each micro-measure influences the classification, as opposed to other machine learning techniques.

Table 3: Precision/Recall analysis result.

	Precision	*Recall*	*F1*
Text Input	0.71	0.74	0.71
Select	0.77	0.76	0.75

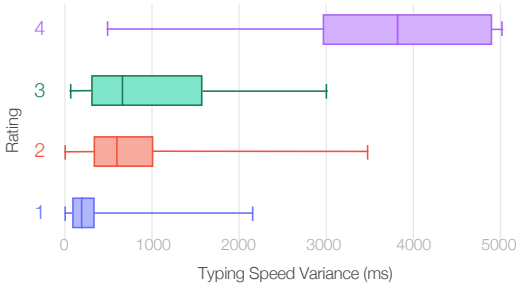

Figure 2: Typing Speed Variance micro-measure distribution across effort ratings in text inputs.

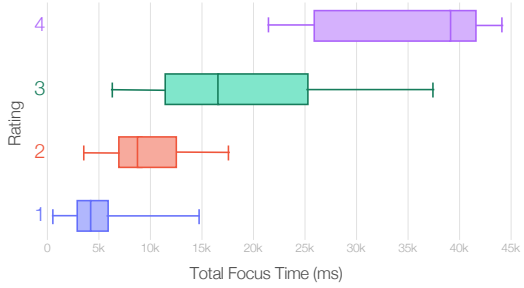

Figure 3: Focus Time micro-measure distribution across effort ratings in selects.

4 RESULTS AND ANALYSIS

The decision trees were able to predict the score with 0.71 precision, 0.74 recall in text inputs, and 0.77 precision, 0.76 recall in selects. F1 scores (harmonic average of precision and recall) were 0.71 and 0.75 respectively (see Table 3). All results in the table are averages of 20 runs (note that the averaged F1s in the table do not result from the averaged precision/recall scores).

The analysis of how micro-measures where used in the decision trees provided some interesting insights. For instance, we learned that the most dominant micro-measure in text inputs (estimated with the *Gini Importance*) was the **Typing Speed Variance** (TSV). The meaning we bestowed to this finding is that the largest interaction effort required by text inputs is when the user cannot find a steady typing pace. Observing the TSV distribution across ratings in the Box Plot of Fig. 2, we may clearly see distinctive values for each rating, ascending from 1 to 4. Ratings 3 and 4 show more diverse values (i.e. higher variance *within* the TSV itself), possibly due to the lower number of samples with these ratings. In the case of selects, the most important micro-measure was **Focus Time**. Analyzing this micro-measure, we can see that the possible values for each rating are clearly different (see related Box Plot in Fig. 3). This shows that settling for Focus Time when other measures were not possible was a good choice, since it was representative of the interaction effort.

Another analysis performed was the comparison between different pairs of micro-measures with the purpose of studying their relationship and observe whether a measure became irrelevant or could be subsumed by another. For example, Fig. 4 shows the comparison between Typing Latency and Total Typing Time in text inputs. While there is an apparent relationship between both measures, this is not constant. This result was similar for the other pairs of measures, so we were not able to discard any. Nevertheless, in the case of the select widget, two micro-measures were not being used to build the decision tree and hence discarded in subsequent experiments: Keystrokes and Option Changes. We think that Keystrokes was not used because very few users interact with select widgets using only the keyboard, i.e., without opening the options list. For this reason, in most samples, this micro-measure got a zero value. Something similar occurred with Option Changes, which only took three values that got repeated for the different ratings, thus obfuscating the separation of the samples for each rating within this micro-measure.

5 CONCLUSIONS AND FUTURE WORK

The first results show that it is possible to emulate an UX expert's opinion on interaction effort, in the constrained context of mouse-and-keyboard behavior around a single widget. The sample we generated for training and testing, although limited in number (due to the manual labor required), was diverse in both the user profiles and website domains. This reinforces the presumption that, unlike other comparable works, our approach appears suitable for any context.

From the perspective of our original goal, i.e., to automatically compare alternative designs generated via CSWR, we are now able to compare the performance of the CSWR "Text Input into Select" against a plain text input widget. We may do so by averaging predicted interaction effort ratings for several interactions from users in the wild.

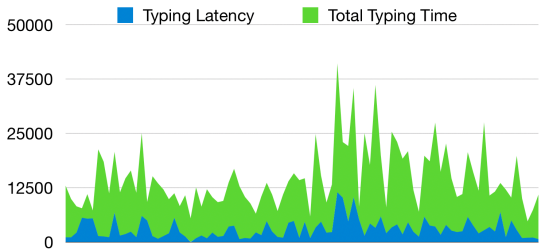

Figure 4: Comparison between Typing Latency and Total Typing Time in text inputs for samples rated with "2".

ACKNOWLEDGMENTS

The authors acknowledge the support from the Argentinian National Agency for Scientific and Technical Promotion (ANPCyT), grant number PICT-2015-3000.

Having completed this initial research on text inputs and selects, we are currently experimenting with other HTML widgets for which we have already selected the relevant micro-measures. Having the ability to score a larger catalog of widgets will allow us to reach the original goal of comparing all CSWRs, i.e., alternative ways of interaction for the same task, in the wild.

We are also working on streamlining the process for manual rating, which is currently somewhat cumbersome, even with the help of our tool. This is necessary to maximize the number of rated samples on which our approach relies.

Regarding the prediction stage with decision trees, we have not discarded other techniques for predicting the ratings for interaction effort, like neural networks. We did run early tests using this technique and the results, while not as satisfactory as those of the decision trees, could improve as the sample set grows. Also, the insights we gathered from using the decision trees with respect to the micro-measures may help us to design more efficient neural networks.

Finally, future work includes studying other applications of the interaction effort score and its performance outside the context of refactoring. Moreover, we'd like to transfer our research to touchscreen devices, analyzing the relevant micro-measures for interaction effort in the mobile context.

REFERENCES

[1] Raluca Budiu. Interaction cost: definition. 2013. Nielsen Norman Group. https://nngroup.com/articles/interaction-cost-definition/ Accessed 2018-12-21

[2] Paolo Burzacca and Fabio Paternò. 2013. Remote usability evaluation of mobile web applications. In *Proceeding of the 15th International Conference on Human-Computer Interaction (HCI 2013)*. Lecture Notes in Computer Science, vol 8004. Springer, Berlin, Heidelberg, 241–248. DOI: https://doi.org/10.1007/978-3-642-39232-0_27

[3] J. Grigera, A. Garrido, J. M. Rivero, and G. Rossi. 2017. Automatic detection of usability smells in web applications. International Journal of Human-Computer Studies. (97), 129-148. DOI: https://doi.org/10.1016/j.ijhcs.2016.09.009

[4] J. Grigera, A. Garrido, and G. Rossi. Kobold: web usability as a service. In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2017)*. IEEE Press, 990-995. DOI: https://doi.org/10.1109/ASE.2017.8115717

[5] Lars-Erik Janlert and Erik Stolterman. 2017. The Meaning of Interactivity – Some Proposals for Definitions and Measures. Human-Computer Interaction, 32:3, 103-138, DOI: https://doi.org/10.1080/07370024.2016.1226139

[6] M. Larusdottir, J. Gulliksen, Å. Cajander. 2017. A license to kill – Improving UCSD in Agile development. Journal of Systems and Software, 123, 214-222. DOI: https://doi.org/10.1016/j.jss.2016.01.024

[7] M. Nebeling, M. Speicher, and M. Norrie. 2013. W3touch: metrics-based web page adaptation for touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2013)*. ACM, New York, NY, USA, 2311-2320. DOI: https://doi.org/10.1145/2470654.2481319

[8] A. Oulasvirta, S. De Pascale, J. Koch, T. Langerak, J. Jokinen, K. Todi, M. Laine, M. Kristhombuge, Y. Zhu, A. Miniukovich, G. Palmas, and T. Weinkauf. 2018. Aalto Interface Metrics (AIM): A Service and Codebase for Computational GUI Evaluation. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. ACM, 2018: 16-19. DOI: https://doi.org/10.1145/3266037.3266087

[9] M. Speicher, A. Both, and M. Gaedke. 2015. S.O.S.: Does Your Search Engine Results Page (SERP) Need Help? In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing System (CHI 2015). ACM, New York, NY, USA, 1005-1014. DOI: https://doi.org/10.1145/2702123.2702568