KLM Form Analyzer: Automated Evaluation of Web Form Filling Tasks Using Human Performance Models

Christos Katsanos[1], Nikos Karousos[1], Nikolaos Tselios[2], Michalis Xenos[1], and Nikolaos Avouris[2]

[1] Hellenic Open University, Parodos Aristotelous 18, Patras, Achaia 26335, Greece
{ckatsanos,nkarousos,xenos}@eap.gr
[2] University of Patras, Rio, Achaia 26500, Greece
nitse@ece.upatras.gr, avouris@upatras.gr

Abstract. Filling forms is a common and frequent task in web interaction. Therefore, designing web forms that enhance users' efficiency is an important task. This paper presents a tool entitled KLM Form Analyzer (KLM-FA) that enables effortless predictions of execution times of web form filling tasks. To this end, the tool employs established models of human performance, namely the Keystroke Level Model and optionally the Fitts' law. KLM-FA can support various evaluation scenarios, both in a formative and summative context, and according to different interaction strategies or modeled users' characteristics. A study investigated the accuracy of KLM-FA predictions by comparing them to participants' execution times for six form filling tasks in popular social networking websites. The tool produced highly accurate predictions (89.1% agreement with user data) in an efficient manner.

Keywords: Web form design, task efficiency, user performance time, automated tool, human performance models.

1 Introduction

Usability of interactive web forms is a critical aspect of the overall user experience. Form filling is a data entry task, and thus user efficiency is of particular importance in the design of web forms. Current design practices are mostly empirical and rely on guidelines derived from experimental studies comparing alternative designs and usability experts' experience or observations. For instance, the type of form elements as well as their positioning in the form layout significantly affect users' performance [1].

One may argue that theoretically-based approaches have had a limited impact on web form design practices. Unlike desktop [2] or mobile interfaces [3-4], GOMS [5] and its simplified version Keystroke-Level Model (KLM) [5-6], have been rarely used to guide web form design or evaluation. In addition, if field size and position on the form layout are not taken into account in such model-based techniques, superficial results may arise. For instance, interaction with a dropdown menu theoretically takes

P. Kotzé et al. (Eds.): INTERACT 2013, Part II, LNCS 8118, pp. 530–537, 2013.

longer than interaction with radio buttons. This is due to an additional point and click needed to open the dropdown menu. However, in one study the latter hypothesis was confirmed [7] and in another it was rejected [8].

As a result, there is a need to bridge HCI models, such as KLM, with design and evaluation practices. Previous research [9] resulted in the development of CogTool, a tool that can produce quantitative, model-based predictions of skilled performance time from tasks demonstrated on storyboard mockups of a user interface. CogTool-Explorer [10] builds upon CogTool to predict a user's goal-directed exploratory interaction with a website. Currently available model-based tools require non-trivial manual work to examine forms. In addition, if a large scale summative evaluation is needed, the evaluator has to repeat the same process without any particular assistance. Furthemore, the plethora of available functions and generic modeling nature of existing tools can overwhelm and discourage practitioners who, in most cases, need a simple tool focused on the problem at hand.

To tackle the aforementioned problems, in this paper a novel tool entitled KLM Form Analyzer (KLM-FA) is presented. KLM-FA extends the capabilities of existing modeling tools for practitioners by focusing specifically on automating the analysis of web forms. The paper is organized as follows: The tool functionality and usage is delineated in the next section, along with its internal architecture and reasoning. Finally, a validation study comparing KLM-FA results to human performance data is presented and discussed.

2 The KLM Form Analyzer Tool

The main objective of KLM-FA (available at http://klmformanalyzer.weebly.com) is to support design and evaluation of web forms in an effective and efficient manner. The tool employs web parsing algorithms, coupled with KLM and Fitts' modeling to estimate the time required to fill a web form according to different interaction strategies (e.g. using tab to move across the elements) or users' characteristics (e.g. age and typing expertise). Figure 1 presents the main interface and functionality of KLM-FA.

2.1 KLM-FA Typical Usage Scenario

First, the evaluator inputs the URL of the web form to be evaluated or selects a previously evaluated form. Next, the evaluator selects a set of analysis preferences related to the modeled user profile (typing ability, age), usage (or not) of Fitts' law in the calculations, and hypotheses about the interaction, such as initial cursor position and whether the user moves across form elements using the mouse or the keyboard. The evaluator can also assign a predefined field type to text elements (e.g. username, email) to easily specify their number of keystrokes. The tool provides an editable list of field types that covers most of the elements used. The default typical field entry lengths rely on empirical data available in the literature (e.g. mean password length [11]) and a dataset of our own with 839 registered web users' personal data.

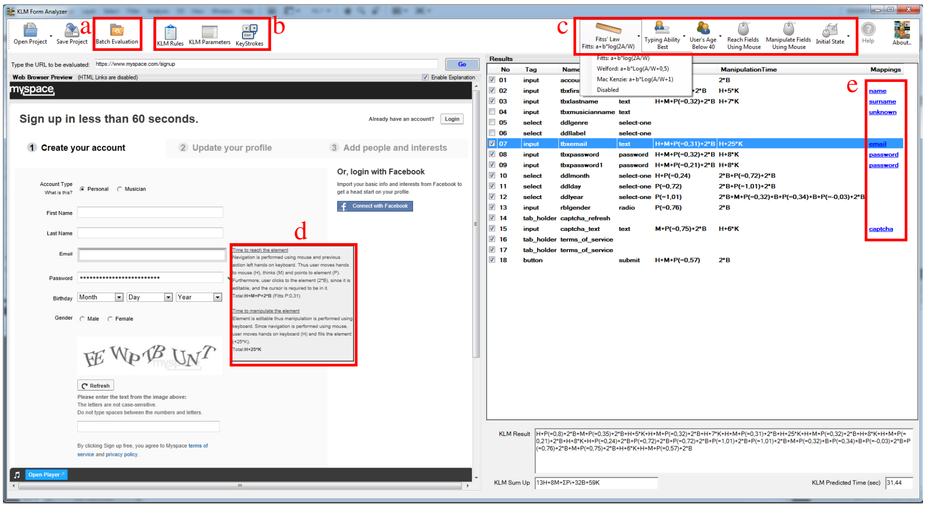

Fig. 1. Overview of the KLM-FA interface and functionality: (a) Mass scale evaluation, (b) Analysis rules and parameters, (c) Analysis preferences, (d) Tooltip explaining KLM modeling for the selected field, (e) Semantic mapping of a text-entry element to a number of keystrokes

Next, KLM-FA runs an algorithm which parses the evaluated form, produces the sequence of predicted user actions (KLM operators) based on the evaluator's selected analysis preferences and estimates task completion time based on a set of analysis parameters related to KLM and Fitts' law calculations. Based on empirical data [12-13], the tool provides a set of default values for the analysis parameters, which can be easily modified through appropriate dialogues. The internal architecture and algorithms employed by the tool are delineated in the next section.

The output of the tool is an interactive web form preview synchronized with a results list: when an element is selected in the web form preview it is highlighted in the results list and vice versa. Depending on the evaluation scenario, mental operators can be added or any element can be excluded from the analysis by simply unchecking it from the results list. In all cases, the tool updates the results in real time. Furthermore, KLM-FA provides an option that elaborates the underlying KLM modeling for each element in a tooltip. In this way, one can trace step-by-step the KLM modeling analysis by simply selecting the sequence of the form elements either in the web preview or in the list. Each evaluated form can be saved and/or subsequently modified. In addition, KLM-FA can employ mass scale summative evaluations by selecting a set of saved projects. Then, the tool runs an analysis of all the selected forms using the same settings for all projects and saves the results in an XML file.

2.2 Internal Architecture and Reasoning

The KLM-FA internal architecture comprises two conceptual layers: the user interface layer, responsible for the interaction with the tools' user, and the KLM analysis layer (named *KLMKernel*). The latter handles the elements identification through webpage parsing and the KLM modeling calculations. The fundamental data structure of the KLM-FA is the *Element* class, which is used to represent each element. Table 1 presents its main data members. The *KLMKernel* first parses a given form to produce a structured list of *Elements* (named *ElementList*) and then performs the KLM analysis and updates the *ElementList* with the results.

Table 1. Data members of the *Element* class, the fundamental data structure of KLM-FA

Data member	Brief Description
TagName	HTML form tag (e.g. <select>) or special KLM-FA tag (e.g. tabholder)
Type	HTML type of <input> tag (e.g. password, text, radio etc.)
Name	Value of the HTML name attribute
Choices	Number of choices for radio buttons and drop-down lists
MappedField	Semantic mapping of a text-entry element to a number of keystrokes
Position	X and Y coordinates of the element position in the form layout
Size	Width and height of the element
MentalExtras	Mental operators manually added or removed
Active	Flag to denote whether the element is included in the calculations
ReachTime	Predicted time required to reach the element
ManipulationTime	Predicted time required to manipulate the element
KLMexplanation	Explanatory text describing the rationale of the derived KLM operators

Parsing Module. It is responsible for parsing a provided webpage to identify existing forms and their elements. To this end, it employs two separate algorithms: a) *form identifier*, and b) *element identifier*. The *form identifier* parses the HTML DOM loaded in the internal browser and finds all forms. It filters out forms that cannot be eligible for analysis (hidden) and presents a "select form" dialogue if two or more forms are found. Then, the *element identifier* parses the selected form, identifies and stores visible fields in the internal *ElementList*. Currently, the tool cannot identify fields when either Flash or AJAX is used. However, KLM-FA provides support to manually add fields and specify their properties in a straightforward manner (e.g. clicking on unidentified field registers its position and size). The following pseudo-code sketches the *element identifier* algorithm which produces an updated *ElementList*.

```
GetFormElements(FormNode, ElementList){
  foreach Element in FormNode.Elements
      if (validate_element(Element))
        if (Element.Type == "radio")
          calculate_middle_Element(formNode, Element)
        else
          if (Element.Type == "select")
            calculate_select_options(formNode, Element)
      ElementList.Add(Element)
      Element.Active = isElementInsideHiddenDiv(Element)
}
```

Analysis Module. This module performs the KLM modeling and related calculations. It takes as input the *ElementList* along with the following parameters:

- evaluator-defined preferences concerning modeled users' typing proficiency and age, mouse or keyboard usage for navigation and manipulation of the elements, Fitts' Law activation, and initial position of the user's hands and cursor,
- predefined time values for KLM operators and Fitts' constants,
- paired list of *[fieldname-keystrokes]* that is used for text entry calculations, and
- set of KLM analysis rules regarding placement of mental operators, and other specific modeling assumptions (e.g. manipulation of dropdown lists with keyboard).

For each form element the algorithm produces the sequence of required actions (KLM operators) to first reach it (ReachTime) and then manipulate it (ManipulationTime). This distinction enables flexible modeling of various user interaction strategies (e.g. tab-based navigation). In addition, the algorithm creates an explanatory text of the KLM modeling rationale which can be displayed as a tooltip in the web preview form.

Concerning Fitts' law, the analysis module calculates the pointing operator by storing the previous position of a simulated mouse cursor and updating it whenever the modeling process requires its movement to a new position. The MacKenzie-Shannon formula and constants [2] for Fitts' law are the default selection for modeling pointing device movement time. However, given the lack of consensus on the Fitts' formula [14], the tool offers additional options (e.g. Welford's formulation [2]) and it is also easy to add further formulas or modify constants values.

Finally, KLM-FA sums up the results and produces a sequence of operators and the predicted form completion time for the provided analysis preferences and parameters. The entire form analysis concludes to an updated *ElementList* that can be saved, reanalyzed with a different set of parameters or exported to an XML file. In addition, the form analysis algorithm can be executed for a set of saved forms (*ElementLists*) to rapidly produce massive KLM modeling results for the same set of analysis parameters. The following pseudocode sketches the *form analysis* algorithm.

```
Analyze(ElementList){
  TypeElement prev_el;
  foreach element in ElementList
      if (is_active(element))
        prev_el = ElementList.GetPreviousActiveElement
                   (element, nav_using_mouse());
        if (Fitts_Law is_enabled() and nav_using_mouse())
            estimate_FittsP_Reach(element);
        analyze_element_reach(element, prev_el);
        analyze_element_manipulation (element);
}
```

3 Validation Study

The aim of the study was to investigate the accuracy of the results obtained by using KLM-FA. The study compared the KLM-FA predictions with user testing data for three signup forms of popular social networking websites: facebook, twitter, and myspace. For each form, two interaction strategies were investigated: a) *mouse-based*, in which a user is assumed to interact with the form using the mouse, except for input in text entry fields, b) *keyboard-based*, in which form fields are reached using the tab key and manipulated only through the keyboard. In both interaction strategies, users were assumed to fill the fields following the form layout. All in all, times for a total of six form-filling tasks (3 forms x 2 interaction strategies) as calculated by KLM-FA and measured through user testing were compared.

Fifteen University students, 12 male, with a mean age of 27 (sd=5.8), a mean of 14 years of QWERTY keyboard usage (sd=4.2) and a mean typing speed of 42 corrected words per minute (sd=16) took part in the study. First, participants completed a short online demographics questionnaire and a typing speed test. Next, they were asked to perform 10 trials for each of the six tasks and their behavior was monitored by an in-house web-based software developed for the needs of the study. Ten trials have been used in similar studies [15] to allow users' to reach skilled performance. In this study, participants were allowed to perform additional trials if their tenth trial was not error-free (max number of trials observed = 12). Task execution times were derived from participants' last error-free trial.

In each trial, participants were first presented with an instructions webpage, followed by the actual form which appeared when they clicked on a link. In the instructions page, they were asked to familiarize themselves with the form registration data and were instructed to strictly employ a specific interaction strategy (i.e. mouse-based or keyboard-based) in order to fill the form as fast and correct as possible. In the form webpage, participants were first required to press a start button located in the top-left of the screen, which started logging of actions and ensured the same starting cursor position for all. The presentation order of both the forms and interaction strategies were counterbalanced to avoid serial order effects. Participants used an HP standard keyboard, an HP 3-button optical mouse and a TFT 17" screen with a resolution of 1280x1024. User sessions lasted about 75 minutes.

In KLM-FA, the following assumptions were used: a) the user was a poor typist (40 wpm) and aged below 40, b) system response time was negligible, c) the cursor's initial position was at the top–left corner of the page, d) tool defaults for all analysis parameters were used, apart from field entry lengths that were appropriately adjusted for each task, e) the user's hand began on the main device of each interaction strategy, and f) Fitts' law calculations were enabled in KLM-FA. KLM-FA analyses were also conducted on a TFT 17" screen with a resolution of 1280x1024. The process to evaluate all six tasks using KLM-FA required approximately 10 minutes.

Table 2 presents participants' task execution times and KLM-FA calculated times for each form and interaction strategy combination, along with the KLM-FA error rate. The error rate was calculated as the participants' mean task time minus the KLM-FA predicted time, and this difference divided by the participants' mean task time. Results show that the mean error of KLM-FA predictions was 10.9% (sd=6.4%), which is well within the 20% margin of error reported in the literature for KLM predictions in other contexts [6], [16]. The lowest and highest KLM-FA error rate values were 4.5% and 17.6% respectively. In general, KLM-FA tended to slightly overestimate (16.7% on average) and underestimate (5.1% on average) task time in the mouse-based and keyboard-based interaction strategies respectively.

Table 2. Study results showing means and, in parenetheses, standard deviations

Signup form	Interaction strategy	Participants' task time (ms)	KLM-FA predicted time (ms)	Error rates of KLM-FA predictions (%)
Facebook	Mouse-based	30739 (6742)	35320	14.9%
Facebook	Keyboard-based	27306 (7752)	25640	6.1%
Myspace	Mouse-based	33201 (6341)	39050	17.6%
MySpace	Keyboard-based	29641 (9310)	28320	4.5%
Twitter	Mouse-based	22478 (5146)	26420	17.5%
Twitter	Keyboard-based	23144 (6108)	24240	4.7%

4 Conclusions

This paper presents KLM-FA, a tool that employs predictive models of human performance to estimate execution times of web form filling tasks. In addition, a study is presented that demonstrates the accuracy of KLM-FA predictions by comparing them to human execution times for the same six form filling tasks.

KLM-FA extends the capabilities of existing general modeling tools for practitioners, such as CogTool [9], by focusing specifically on web form interaction. In this way, KLM-FA increases automation of evaluation tasks, minimizes the required effort and achieves increased simplicity and flexibility, thus increasing the chances of its adoption in actual practice. As a result, practitioners can rapidly evaluate alternative web form design approaches using a variety of scenarios. In addition, the ability of KLM-FA to evaluate keyboard-based interaction with web forms can be valuable in automated accessibility testing. KLM-FA can also be used to produce benchmark data of form completion times for specific web domains, such as social networking or e-commerce. Finally, the tools' step-by-step tracing of the KLM modeling supports learning through examples and thus can be valuable for both educators and students.

Investigating the effect of KLM-FA adoption on the learning outcome, while educating students in KLM, constitutes a future research goal. In addition, we plan to conduct additional studies that compare KLM-FA predictions with human performance data. An additional future research goal is to incorporate enriched models of KLM [17] in order to support design of web forms that enhance users' efficiency in mobile interaction contexts.

Despite the advantages of the presented automated approach, it only addresses task efficiency which is one aspect of the web user experience. Other tools that automate different aspects of web design are also available [18]. However, all such approaches should be used in conjunction with user-based methods.

Acknowledgements. This work has been partly funded by the European Territorial Cooperation Operational Programme "Greece-Italy 2007-2013" under the project Intersocial.

References

1. Bargas-Avila, J.A., Brenzikofer, O., Tuch, A.N., Roth, S.P., Opwis, K.: Working towards Usable Forms on the Worldwide Web: Optimizing Multiple Selection Interface Elements. Advances in Human-Computer Interaction 2011, Article ID 347171, 6 pages (2011)
2. Soukoreff, R.W., MacKenzie, I.S.: Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI. International Journal of Human-Computer Studies 61(6), 751–789 (2004)
3. Myung, R.: Keystroke-Level Analysis of Korean Text Entry Methods on Mobile Phones. International Journal of Human-Computer Studies 60(5-6), 545–563 (2004)
4. Maragoudakis, M., Tselios, N.K., Fakotakis, N., Avouris, N.M.: Improving SMS usability using Bayesian Networks. In: Vlahavas, I.P., Spyropoulos, C.D. (eds.) SETN 2002. LNCS (LNAI), vol. 2308, pp. 179–190. Springer, Heidelberg (2002)
5. Card, S.K., Moran, T.P., Newell, A.: The Psychology of Human Computer Interaction. Lawrence Erlbaum Associates, Hillsdale (1983)
6. Card, S.K., Moran, T.P., Newell, A.: The Keystroke-Level Model for User Performance Time with Interactive Systems. Communications of the ACM 23(7), 396–410 (1980)
7. Healey, B.: Drop Downs and Scrollmice: The Effect of Response Option Format and Input Mechanism Employed on Data Quality in Web Surveys. Social Science Computer Review 25(1), 111–128 (2007)
8. Hogg, A., Masztal, J.J.: Drop-down, Radio Buttons, or Fill-in-the-blank? Effects of Attribute Rating Scale Type on Web Survey Responses. In: Proceedings ESOMAR 2001 (2001)
9. John, B.E., Prevas, K., Salvucci, D.D., Koedinger, K.: Predictive Human Performance Modeling Made Easy. In: Proceedings CHI 2004, pp. 455–462. ACM Press (2004)
10. Teo, L., John, B.E.: CogTool-Explorer: towards a Tool for Predicting User Interaction. In: Ext. Abstracts CHI 2008, pp. 2793–2798. ACM Press (2008)
11. Cazier, J., Medlin, B.: Password Security: An Empirical Investigation into E-Commerce Passwords and Their Crack Times. Information Systems Security 15(6), 45–55 (2006)
12. Kieras, D.: Using the Keystroke-Level Model to Estimate Execution Times. Unpublished report (2001)
13. Olson, J.R., Olson, G.M.: The Growth of Cognitive Modeling in Human-Computer Interaction since GOMS. Human-Computer Interaction 5, 221–265 (1990)
14. Drewes, H.: Only One Fitts' Law Formula Please! In: Ext. Abstracts CHI 2010, pp. 2813–2822. ACM Press (2010)
15. Luo, L., John, B.E.: Predicting Task Execution Time on Handheld Devices Using the Keystroke-Level Model. In: Ext. Abstracts CHI 2005, pp. 1605–1608. ACM Press (2005)
16. Ivory, M.E., Hearst, M.A.: The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys 33(4), 470–516 (2001)
17. Holleis, P., Otto, F., Hussmann, H., Schmidt, A.: Keystroke-Level Model for Advanced Mobile Phone Interaction. In: Proceedings CHI 2007, pp. 1505–1514. ACM Press (2007)
18. Katsanos, C., Tselios, N., Avouris, N.: A Survey of Tools supporting Design and Evaluation of Websites based on Models of Human Information Interaction. International Journal on Artificial Intelligence Tools 19(6), 755–781 (2010)