Visualizing the Keystroke Level Model in Web Form Filling Tasks through Adaptive Animation towards Learning

Nikos Karousos
[1]Hellenic Open University,
Patras, Greece
[2]Technological Educational Institute of Western Greece,
Patras, Greece
karousos@eap.gr

Dionisis Karousos
[1]IMIS Lab, MEAD, University of Patras,
Patras, Greece
[2]Technological Educational Institute of Western Greece,
Patras, Greece
dkarousos@gmail.com

ABSTRACT

The KLM Form Analyzer (KLM-FA) application automatically estimates the time that a user needs to fill out a web form. The time estimation mechanism is based on the combination of two established methods on the Human-Computer Interaction (HCI) domain, the Keystroke Level Model (KLM) and the Fitts' Law. Primarily, the KLM-FA aims to support HCI researchers and practitioners during the design and the evaluation of a web form. However, recent studies have revealed the educational potentials of KLM-FA. In order to further facilitate its educational purposes, a new innovative KLM-FA feature was developed. In particular, an animation-based technique entitled 'Watch and Learn' that provides a step-by-step visual explanation for each form filling step was integrated into the tool.

CCS CONCEPTS

• Applied computing~Computer-assisted instruction • Humancentered computing~Human computer interaction (HCI) • Information systems~World Wide Web

KEYWORDS

Human−computer interaction; educational tool; web form design; Keystroke Level Model; animation, Fitts' law.

ACM Reference format:

Nikos Karousos and Dionisis Karousos. 2018. Visualizing the Keystroke Level Model in Web Form Filling Tasks through Adaptive Animation towards Learning In *XXXXXXX*

1 Introduction

The Keystroke Level Model [2] supports the modeling of a user's behavior while performing well-known CHI tasks. In this way, the estimation of the time required to complete a task is feasible. Furthermore, the combination of KLM with Fitts' Law [5] can be more accurate than using only KLM in desktop environments. This is due to the impressive capability of Fitts' Law that allows to estimate the required time of a mouse movement according a) to the distance between mouse pointer and target and b) to the width of the target.

In HCI these two techniques are often part of educational material in several HCI related courses. Since KLM is considered as a fundamental formal modeling technique, teaching KLM in the classroom is almost mandatory. Plenty web resources and scientific publications can be used as additional educational material [12, 17, 18]. Moreover applications that provides KLM modeling for either estimation or education purposes are available as well. Some representative of them are listed below:

- The CogTool project [3, 6]
- SANLab-CM [16]
- Cogulator: Cognitive calculator for GOMS modeling [4]
- Simple KLM_calculator [10] and Simple KLM calculator (Windows app) [14]
- The KLM Form Analyzer (KLM-FA)_[7, 8, 11]

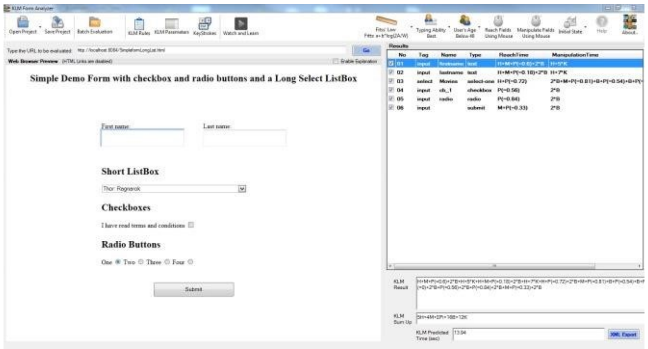
Among them, the KLM-FA has been used in several HCI courses and recently two case studies concerning its learning impact were conducted [9, 15]. The results of these studies emphasized the usefulness of the particular tool as an educational means towards KLM learning. Furthermore, a need for a more comprehensive KLM-FA User Interface (UI), that could aim practitioners at better understanding the KLM, has been also highlighted.

Having in mind that a carefully designed animation may facilitate learning when it provides micro-step information and time-specific details [13], we tried to build a mechanism that animates the KLM analysis produced by the KLM-FA estimation algorithm. To this end, this paper presents a new feature of the KLM Form Analyzer (KLM-FA) application that comes to facilitate learning through the provision of a KLM analysis presentation via an adaptive animation way. This mechanism focuses on each step (KML operator) that is required to be performed in order to fill a web form and provides visualized details as well.

The rest of the paper is structured as follows. Section 2 presents the KLM-FA functionality. In section 3, the results of the studies about the learning impact of the KLM-FA are described. Next, section 4 introduces the KLM-FA new feature and describes its functionality together with implementation info. Finally, we discuss some new challenges and future work that were raised after the production of this new feature and we conclude.

2 The KLM-FA application

KLM-FA is an application that estimates the required time for filling a web form. It uses both the Keystroke Level Model and Fitts' Law to simulate a typical user who fills a web form and to calculate the result in terms of required steps and total time. The usage of this windows-based application focuses on supporting both web form design and evaluation. However, recently the KLMFA has been used for educational purposes (in HCI related courses) with very positive results [8, 14].



Type the URL to be evaluated: http://localhost:8084/SimpleformLongList.html

No	Tag	Name	Type	ReachTime	ManipulationTime
01	input	firstname	text	H+M+P(=0.6)+2*B	H+5*K
02	input	lastname	text	H+M+P(=0.18)+2*B	H+7*K
03	select	Movies	select-one	H+P(=0.72)	2*B+M+P(=0.81)+B+P(=0.54)+B+P(
04	input	cb_1	checkbox	P(=0.56)	2*B
05	input	radio	radio	P(=0.84)	2*B
06	input		submit	M+P(=0.33)	2*B

KLM Result: H+M+P(=0.6)+2*B+H+5*K+H+M+P(=0.18)+2*B+H+7*K+H+P(=0.72)+2*B+M+P(=0.81)+B+P(=0.54)+B+P (=0)+2*B+P(=0.56)+2*B+P(=0.84)+2*B+M+P(=0.33)+2*B

KLM Sum Up: 5H+4M+7P+16B+12K

KLM Predicted Time (sec): 13.94

Figure 1: The KLM-FA application while parsing and analyzing a web form. On the left an internal browser displays the loaded web form and on the right a list of the detected form elements and the required operators for reaching and filling them is presented in a grid. On the bottom right corner of the application the final result is calculated.

A KLMF-FA user usually starts with a web form selection by typing a URL in the internal web browser at the left of the screen (see Fig. 1). Then, the tool loads this web form, parses it, and detects the form elements. After that, the tool applies the KLM reasoning to the selected form elements: it calculates the required KLM operators first to reach and then to manipulate (fill) each one of the elements. The tool supports the following KLM operators: move hands from keyboard to mouse and vice versa (H), perform a mental activity (think) (M), move mouse (point) (P), press/release mouse button and click mouse (B and 2*B) and keypress (K). Finally, it sums up the operators and calculates the total time by replacing each operator with a particular predefined numeric value. Moreover, by enabling Fitts' Law, the pointing operator (P) is recalculated based a) on the distance between mouse pointer and target and b) on the width of the target. This leads to a more precise numeric result.

User is able to configure the KLM-FA analysis by:

- enabling/disabling Fitts' law or by selecting the appropriate Fitts' law model formulation
- selecting simulated user's age (this impacts the KLM operators values)
- selecting simulated user's typing ability (this impacts the KLM operators values)
- declaring mouse or keyboard as the main means for reaching or manipulating the elements
- setting up the initial state (and position) of both the cursor and the hands of the simulated user.

Furthermore, the user can declare when a KLM rule should be applied (for example when a mental operator (M) is needed), and can change the time values of the KLM operators and Fitts' law constants as well. In order to estimate the filling time for each text element using particular text values, KLM-FA supports the mapping of each text element to a pre-declared element name that comes with a predefined required number of characters to be filled. Hence, KLM-FA manages 'keystroke profiles' that consist of a *name-value* list of pairs in which *name* is the pre-declared element name and *value* is the number of the required characters to be filled. It's worth mentioning that the KLM-FA allows to open/save a project and to perform massive time estimations of a set of web forms for comparison purposes.

3 Recent case study results

In the past, researches have examined the KLM-FA learning impact when the application is used as an educational means: students found that the KLM-FA tool helped them to understand the KLM and Fitts' law. As a result, it was argued that KLM-FA can support teaching and learning of HCI modeling in the context of distance education [9].

Recently, a relative study was conducted in the context of a traditional campus-based education correspondingly. Findings of the second research were in agreement with the first one [15]. Moreover, during the second research, some issues to be improved were tracked down. Among them, the enrichment in modeling explanation was underlined. In addition, the most frequentlymentioned students' general suggestion was to wide its availability through different operating systems.

Considering the need for better KLM explanation, the KLM-FA was revised with the intention of upgrading the KLM reasoning explanation to a satisfactory level. A very interesting approach to this target is the usage of animation technique. Although the question "do animations help learning?" had not received always positive answers, recent studies indicated that when animation is used to represent sequential dynamic activity, learners may actually benefit [1].

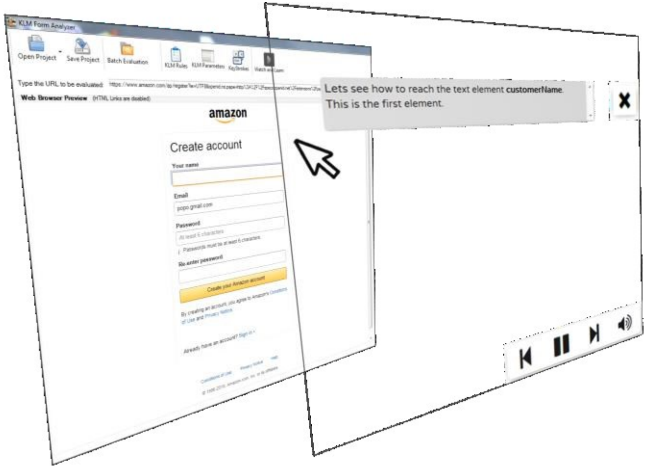
Furthermore, the idea to use web technologies in order to achieve this will be helpful in a future step towards full web based provision of the KLM-FA.

4 The KLM-FA new feature: the 'Watch and Learn' option

The main goal of the KLM-FA 'Watch and Learn' feature is to visualize, step-by-step, the KLM analysis of a web form filling task.

This feature tries to visually explain the algorithm that KLMFA uses in order to analyze the web form and to estimate the required time to fill it.

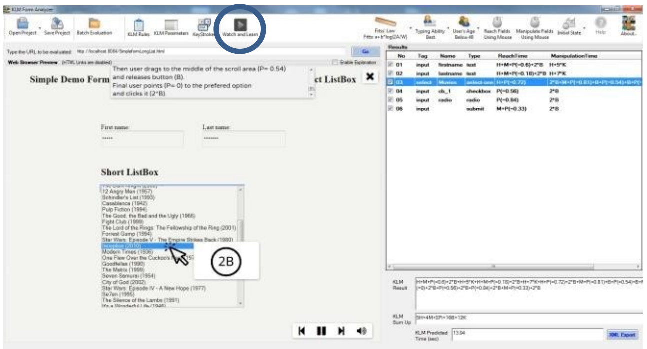
The KLM-FA simulates the entire web form filling process using animation in a transparent layer over the web page preview panel (see Fig 2.) In this layer, KLM-FA uses a virtual (fake) mouse pointer, a message box that displays the explanation messages, a navigation bar with the ability to move step(s) forward or backward and a set of animated gifs that depict the KLM basic operators (H,M,P,B and K).



Lets see how to reach the text element customerName. This is the first element.

Figure 2: KLM-FA builds a transparent layer over the web form and then uses it as a background layer to depict all the graphic items that participate to the animation scene.

As a result, KLM-FA allows a user to simple watch the entire process of a web form filling task and assist him/her in order to understand the application reasoning during the KLM analysis. Furthermore, the combination of a message speaker that explains all the performed actions using audio together with subtitles gets him/her free of reading the messages in the message box, and allows him/her to focus on the simulated steps. In simple words, this KLM-FA add-on tries to provide a movie-like interactive experience to the user. Moreover, navigation buttons allows the user to mute/unmute sound, pause/play the animation and move forward or backward in the animated form filling steps (see Fig. 3). It is obvious that the difficult part of the task was not the provision of just a movie/tutorial over a specific web form but the provision of an adapted animated tutorial on every different form that the user wishes to study. Hence, the KLM-FA builds the entire 'Watch and Learn' scenario on runtime: each time KLM-FA parses a web form and runs the internal KLM estimation algorithm, a new 'Watch and Learn' scenario is created. Accordingly, the scenario may be rebuilt in cases of changing one or more user preferences or KLM algorithm configuration options.



Then user drags to the middle of the scroll area (P= 0.54) and releases button (B).
Final user points (P= 0) to the prefered option and clicks it (2*B).

No	Tag	Name	Type	ReachTime	ManipulationTime
01	input	firstname	text	H+M+P(=0.6)+2*B	H+5*K
02	input	lastname	text	H+M+P(=0.18)+2*B	H+7*K
03	select	Movies	select-one	H+P(=0.72)	2*B+M+P(=0.81)+B+P(=0.54)+B+P(
04	input	cb_1	checkbox	P(=0.56)	2*B
05	input	radio	radio	P(=0.84)	2*B
06	input		submit	M+P(=0.33)	2*B

KLM Result: H+M+P(=0.6)+2*B+H+5*K+H+M+P(=0.18)+2*B+H+7*K+H+P(=0.72)+2*B+M+P(=0.81)+B+P(=0.54)+B+P(=0)+2*B+P(=0.56)+2*B+P(=0.84)+2*B+M+P(=0.33)+2*B

KLM Sum Up: 5H+4M+7P+16B+12K

KLM Predicted Time (sec): 13.04

Figure 3: Circled on the top of the screen is the new 'Watch and Learn' feature in the KLM-FA. When it is selected, the entire internal browser covered by a transparent layer (DIV) in which a navigation bar, a message box and a big mouse pointer are placed. Afterward, the animation starts and the user watches how the Keystroke Level Model is applied in the particular web form.

Although the tutorial is an animation that takes place over the web form preview (in the left side of the application), all the critical events that affect the form elements are executed in real by calling the appropriate commands. In this way, events like focus, keypresses and clicks are triggered and the web form is actually manipulated by the 'Watch and Learn' player. When a form element is getting focused, the correspondence analyzed element in the grid on the right side of the application is highlighted. Therefore, the user is able to see the sum-up of the klm operators of the element that he/she is watching at the same time in the animation on the left side of the screen.

When the animation finishes, the user can replay it as many times as he/she wants. Furthermore, the user can select different KLM properties such us 'navigation/manipulation of the elements using mouse or keyboard', 'exclude or include elements in the estimation algorithm', 'set the initial state of hands and mouse pointer and then replay the 'Watch and Learn' and see the actual differences of the web filling task process by comparing current and previous animated actions.

Some representative screen capture videos that have been produced for demonstration reasons are available at a temporary Web page: http://klm.quickweb.gr. In this page, visitors can also download a copy of the new beta version of KLM-FA (v2.0). The new stable version of KLM-FA (v2.0) together with the 'Watch and Learn' feature is going to be officially published after the completion of some improvements that are in an ongoing process.

5 The 'Watch and Learn' background mechanism

5.1 The General Architecture

The KLM-FA 'Watch and Learn' feature consists of two major modules: **the KLM scenario builder** and the **KLM animation player**. The builder's target is to generate the KLM scenario during the KLM analysis of a web form. Depending of the elements' type,

Please cite this document as: Nikos Karousos and Dionisis Karousos. 2018. Visualizing the keystroke level model in web form filling tasks through adaptive animation towards learning. In Proceedings of the 22nd Pan-Hellenic Conference on Informatics (PCI '18). Association for Computing Machinery, New York, NY, USA, 239–242. https://doi.org/10.1145/3291533.3291587

KLM builder produces all the appropriate events that are needed to be included in the animation. The whole set of events comprises the KLM animation scenario. These events concern KLM operators for reaching or manipulating each element. They are formed according to a data model that will be explained below and they are passed to the KLM animation player through a JSON format.

After the ‘Watch and Learn’ scenario is produced, the animation player module is called in order to visualize the scenario. This module has been developed on Javascript and is injected to the web form through the Application Programming Interface (API) of the internal web browser of the KLM-FA. Apart from the Javascript injection, a stylesheet declaration together with a small set of HTML DIV areas are injected as well. The above code creates a transparent DIV area over the web form preview and places on it all the appropriate DIV elements (pointer, message area and navigation area) that will be used in the animation. Finally, the start and the exit command for the animation are triggered through the execution of the corresponding Javascript functions by the KLMFA application through the API of the Web Browser control.

5.2 The Data Model

The ‘Watch and Learn’ data model is based on a list of animation events. Each event contains information that will be visualized in the animation by the KLM animation player module. Apart from a unique ID that is needed, each event has a step ID that is been used when the user wants to move one or more steps forward or backward. In other words, events are been grouped to steps: each step consists of a set of sequentially placed events that are required to reach or to manipulate an element.

In a more detailed view, an event consists of the following attributes

- ID: a unique event identifier
- Type: a string that defines the type of the event. Each type is considered from KLM animation player as a separate action that must be visualized in the web browser. Allowed types are: ‘stepinfo’, ‘initialmessage’, ‘simplemessage’, ‘showimage’, ‘tabbpress’, ‘scrolldrag’, ‘click’, ‘mousedown’, ‘mouseup’, ‘radio_check_buttonClick’, ‘listboxOptionclick’, ‘listboxclick’, ‘keypress’, ‘spacepress’ and ‘tabpress’. The ‘stepinfo’ type is used only to determine the step initial state in order to support moving forward and backward.
- Duration: the allowed max duration that an event may occur (displayed).
- Params: a set of name-value paired attributes that are type related and that usually contain the element’s name, the element’s position, the message that has to be displayed etc.

5.3 Implementation Issues

The KLM-FA application has been built on C# programming language. Consequently, this animation feature was also built in C#. However, the decision for the animation module was to be developed using Javascript. In this way, we will have the ability to exploit the web browser capabilities and to be prepared to a future full web based provision of the entire KLM-FA application. On the other hand, this decision lead us to face some critical problems regarding web security issues and web browser emulator capabilities: since Visual Studio uses Internet Explorer V7.0 as an internal browser, we faced many difficulties when web forms were using JQuery and other contemporary web technologies. Fortunately, in order to ensure the support of the KLM animation to a newly created web form, the web developer could include a special HEADER declaration that sets the browser engine to Microsoft Edge or Internet Explorer.

Furthermore, due to security issues, the manipulation of a web page elements from Javascript is not always permitted. For example, expanding of a drop down select list or playing a local sound in a web page are not allowed. Hence, in order to bypass such problems, the inclusion of a hidden local web server to the KLM-FA was a successful solution, although it was not the optimal one.

6 Challenges and future work

The current version of the KLM-FA has now a new interesting functionality that allows users to watch an animation based tutorial of KLM reasoning. However, three tasks have to be considered in the future in order to reach a satisfactory level of the application quality.

Prove the usefulness of the new feature: Even though we strongly believe that the particular feature facilitates KLM learning, this should be scientifically proved. Therefore, the next step is to conduct an appropriate study aiming at the examination of the learning impact of the particular ‘Watch and Learn’ functionality over real students in HCI courses under specific learning activities.

Improve some technical issues regarding animation: There are several limitations of the animation player. The most important of them are:

- Animation operates only when the web page does not require scrolling. There is a need of modifying internal code in both scenario, builder and animation player modules in order to support web pages with scrollbar.
- Several web pages do not visualize the animation panel over the web page due to some CSS or Javascripts issues. In the navigation panel, some animation settings such as speed configuration, subtitling, language etc., will be reconsidered.
- Fitts’ law is not actually visualized in the animation (when Fitts’ law is enabled the KLM-FA automatically calculates the required time of a mouse movement and displays the result close to the P operator). The next stable version should present how Fitts’ law is applied, probably by depicting trigonometry (lines, target width etc.) based on a visual trace of the mouse movement for each pointing operation (or at least the first one).

Make the KLM-FA functionality available through web: Previous research showed that many users were not satisfied with the fact that KLM-FA requires only Windows operating system and

Please cite this document as: Nikos Karousos and Dionisis Karousos. 2018. Visualizing the keystroke level model in web form filling tasks through adaptive animation towards learning. In Proceedings of the 22nd Pan-Hellenic Conference on Informatics (PCI '18). Association for Computing Machinery, New York, NY, USA, 239–242. https://doi.org/10.1145/3291533.3291587

cannot provide its functionality through other systems or through the web. Thus, we believe that moving KLM-FA functionality to the web will extend the number of potential users of it. As a first step, we are planning the provision of the 'Watch and Learn' player functionality through a web page in order to be accessible from anyone. Next, we will reconsider the re-implementation of the entire KLM-FA as a web application.

7 Conclusions

The results of the previous research regarding the learning impact of KLM-FA application emphasized the need for user interface improvement towards learning enhancement. Having in mind that animation has already been successfully used as an educational means, we created a new KLM-FA functionality named 'Watch and Learn" which dynamically builds and visualizes a KLM analysis scenario in a both adapted and animated way. We strongly believe that this unique (in the area of KLM) feature will help students to learn how Keystroke Level Model works by explaining them visually and in details each step of the KLM analysis. Furthermore, the adopted implementation strategy of this feature will give in the future a boost for the provision of this capability to the Web.

ACKNOWLEDGMENTS

Apart from the initial KLM-FA researchers that are already mentioned into the referred papers, we would like to thank our colleague Mr. George Kontogeorgas for providing us with various test forms and the entire set of sounds for the KLM-FA speech functionality.

REFERENCES

[1] Ainsworth, S. 2008. How do animations influence learning. *Current perspectives on cognition, learning, and instruction: Recent innovations in educational technology that facilitate student learning*. (2008), 37–67.

[2] Card, S.K., Moran, T.P. and Newell, A. 1980. The Keystroke-level Model for User Performance Time with Interactive Systems. *Commun. ACM*. 23, 7 (Jul. 1980), 396–410. DOI:https://doi.org/10.1145/358886.358895.

[3] CogTool: *https://cogtool.wordpress.com/*. Accessed: 2018-10-04. [4] Cogulator: A Cognitive Calculator: *http://cogulator.io/*. Accessed: 201810-04.

[5] Fitts, P.M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. 47, 6 (1954), 381–391. DOI:https://doi.org/10.1037/h0055392.

[6] John, B.E., Prevas, K., Salvucci, D.D. and Koedinger, K. 2004. Predictive Human Performance Modeling Made Easy. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), 455–462. [7] Karousos, N., Katsanos, C., Tselios, N. and Xenos, M. 2013. Effortless tool-based evaluation of web form filling tasks using keystroke level model and fitts law. *CHI'13 Extended Abstracts on Human Factors in Computing Systems* (2013), 1851–1856.

[8] Katsanos, C., Karousos, N., Tselios, N., Xenos, M. and Avouris, N. 2013. KLM Form analyzer: automated evaluation of web form filling tasks using human performance models. *14th International Conference on Human-Computer Interaction (INTERACT)* (2013), 530–537.

[9] Katsanos, C., Tselios, N., Karousos, N. and Xenos, M. 2015. Learning Web Form Design by Using the KLM Form Analyzer: A Case Study. *Proceedings of the 19th Panhellenic Conference on Informatics* (New York, NY, USA, 2015), 44–49.

[10] Keystroke Level Model (KLM) Calculator: *http://courses.csail.mit.edu/6.831/2009/handouts/ac18-predictiveevaluation/klm.shtml*. Accessed: 2018-10-04.

[11] KLM Form Analyzer: *http://klmformanalyzer.weebly.com/*. Accessed: 2018-10-04.

[12] KLM-GOMS | Usability Body of Knowledge: *https://www.usabilitybok.org/klm-goms*. Accessed: 2018-10-04.

[13] Morrison, J.B., Tversky, B. and Betrancourt, M. Animation: Does It Facilitate Learning? 8.

[14] Syntagm - Keystroke Level Model Calculator: *http://www.syntagm.co.uk/design/klmcalc.shtml*. Accessed: 2018-10-04.

[15] Tool-Mediated HCI Modeling Instruction in a Campus-Based Software Quality Course | SpringerLink: *https://link.springer.com/chapter/10.1007/978-3-31991238-7_10*. Accessed: 2018-10-04.

[16] W Patton, E. and Gray, W. 2010. SANLaB-CM: A tool for incorporating stochastic operations into activity network modeling. *Behavior research methods*. 42, (Aug. 2010), 877–83. DOI:https://doi.org/10.3758/BRM.42.3.877.

[17] 2017. Keystroke-level model. *Wikipedia*.

[18] MeasuringU: Measuring Task Times Without Users.