WKLM-FA: A Web Application for Automated Human-Computer Interaction Modeling of Web Form Filling Tasks

Nikos Karousos
Hellenic Open University, Patras, Greece
karousos@eap.gr

Fotis Tatsis
Hellenic Open University, Patras, Greece
karousos@eap.gr

Dionisis Karousos
IMIS Lab, Department of Mechanical Engineering and Aeronautics, University of Patras, Greece
karousos@gmail.com

Christos Katsanos
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
ckatsanos@csd.auth.gr

Nikolaos Tselios
Educational Sciences and Early Childhood Education Department, University of Patras, Greece
nitse@ece.upatras.gr

Michalis Xenos
Computer Engineering and Informatics Department, University of Patras, Patras, Greece
xenos@ceid.upatras.gr

ABSTRACT

This paper presents the web-based KLM-Form Analyzer (WKLM-FA), a web application that allows the automated estimation of time that a user needs to fill out a web form. Based on feedback from previous studies, this implementation comes to fulfill the need for a cross-platform application of human-computer interaction models, namely the Keystroke Level Model (KLM) and Fitts's Law. WKLM-FA is a new web-based implementation of the KLM-Form Analyzer (KLM-FA), a desktop application that we have developed in our previous work. WKLM-FA also provides additional functionality compared to KLM-FA, such as a user account mechanism, an evaluation tracking feature, and an admin panel to support multiuser functionality. WKLM-FA is the first online KLM-based time estimation tool for web forms and it can be easily used not only by HCI evaluators but also for educational purposes teaching the KLM and the Fitts's law.

CCS CONCEPTS

• **Human-centered computing** → Human computer interaction (HCI); • **Information systems** → World Wide Web.

KEYWORDS

Human–computer interaction, educational tool, web form design, Keystroke Level Model, Fitts's law

ACM Reference Format:
Nikos Karousos, Fotis Tatsis, Dionisis Karousos, Christos Katsanos, Nikolaos Tselios, and Michalis Xenos. 2020. WKLM-FA: A Web Application for Automated Human-Computer Interaction Modeling of Web Form Filling Tasks. In *24th Pan-Hellenic Conference on Informatics (PCI 2020), November 20–22, 2020, Athens, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3437120.3437303

1 INTRODUCTION

The Keystroke Level Model (KLM) [1] is one of the most representative predictive models in the scientific field of Human-Computer Interaction (HCI). KLM simulates low-level input actions of a user, such as keystrokes and pointing to on-screen targets, and estimates the required time for an expert user to perform an interaction task. Fitts's Law [2] predicts the time required to rapidly move to a target area and if used to model the act of pointing in KLM then the results are more accurate.

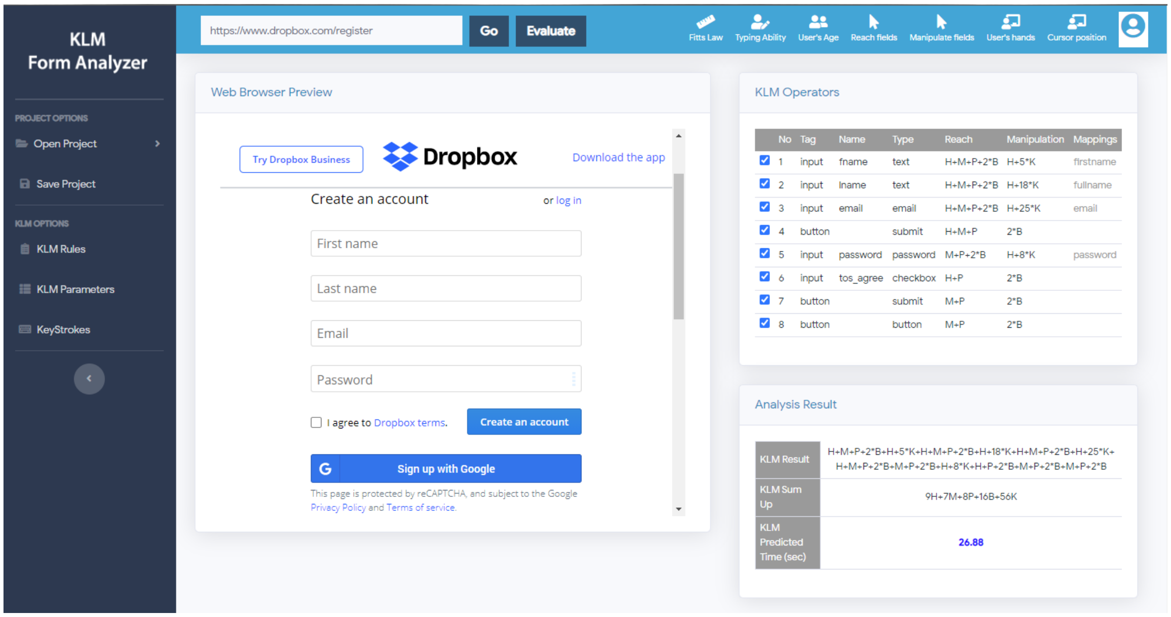
However, using these HCI models is rather challenging in today's software products due to their size and complexity. One must perform a low-level simulation of each user interaction task and then calculate the predicted time by summing up the analyzed operations. The inclusion of Fitts's Law to accurately model mouse movement operation makes the modeling task even more complicated and time-consuming. Thus, various applications that provide KLM modeling for either practitioners' or educators' purposes have been proposed in the literature, such as the CogTool [3], the SANLab-CM [9], the Cogulator [10], the Simple KLM calculator [11] and the KLM Form Analyzer (KLM-FA) [4, 5].

KLM-FA, the desktop version of the WKLM-FA application presented in this paper, has been shown [4, 5] to effectively support automated modeling of web form filling tasks. It has also been used as a learning tool in several HCI courses and has been found [6-8] to have a significant positive effect on students' learning gain in campus-based, blended, and distance education. However, these KLM-FA studies also showed that the current implementation of the tool poses a critical obstacle for non-Windows users. Moreover, the monolithic architecture of the KLM-FA was not able to deal with multi-instances and multi-user-issues.

Motivated by these limitations, this paper presents the development of a new KLM-FA version that is provided as a web application. This web version of KLM-FA (WKLM-FA) is implemented to initially support the provision of the existing KLM-FA functionality. However, the novelty of this implementation is that WKLM-FA,

PCI 2020, November 20–22, 2020, Athens, Greece

ACM ISBN 978-1-4503-8897-9/20/11…$15.00
https://doi.org/10.1145/3437120.3437303



	No	Tag	Name	Type	Reach	Manipulation	Mappings
☑	1	input	fname	text	H+M+P+2*B	H+5*K	firstname
☑	2	input	lname	text	H+M+P+2*B	H+18*K	fullname
☑	3	input	email	email	H+M+P+2*B	H+25*K	email
☑	4	button		submit	H+M+P	2*B	
☑	5	input	password	password	M+P+2*B	H+8*K	password
☑	6	input	tos_agree	checkbox	H+P	2*B	
☑	7	button		submit	M+P	2*B	
☑	8	button		button	M+P	2*B	

KLM Result	H+M+P+2*B+H+5*K+H+M+P+2*B+H+18*K+H+M+P+2*B+H+25*K+ H+M+P+2*B+M+P+2*B+H+8*K+H+P+2*B+M+P+2*B+M+P+2*B
KLM Sum Up	9H+7M+8P+16B+56K
KLM Predicted Time (sec)	26.88

Figure 1: The WKLM-FA main page for modeling human-computer interaction with a web form. The Web Browser Preview (left) displays the evaluated web form. The "KLM Operators" (top right) presents a list of the detected form elements and the required operators for reaching and filling them. The "Analysis Result" (bottom right) displays the summed-up KLM operators and the predicted task completion time.

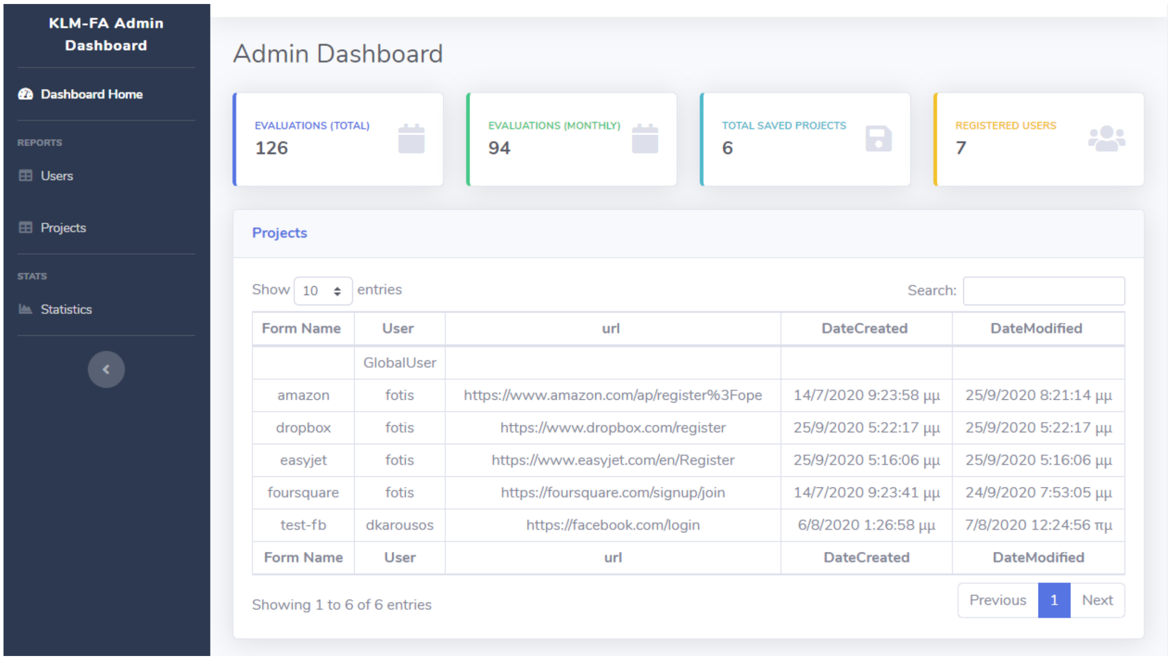
the first online KLM-based time estimation tool for web forms, is platform-independent since it now requires only a web browser to be deployed. Also, WKLM-FA may help researchers with administrator privileges to investigate the usage frequency and purpose (educational or evaluation) of the application and to extract useful statistics on how users evaluate web forms and optimize the design in terms of task completion time.

2 THE WKLM-FA APPLICATION: TOOL DESCRIPTION

WKLM-FA is a web-based application that predicts the required steps and time to fill a web form by employing human-computer interaction models, namely the KLM and Fitts's Law. The tool is addressed to both practitioners and educators who are involved in web form design and evaluation. A typical usage scenario of the WKLM-FA is presented in the following.

First, the WKLM-FA user types the URL of the evaluated web form (Figure 1, top left). Then, the tool employs an algorithm that parses the web form, identifies the form elements (Figure 1, top right), and renders the page in the "Web Browser Preview" area (Figure 1, left). Next, the tool employs an algorithm that applies the KLM modeling to the form elements selected in the corresponding list (Figure 1, top right). In specific, it calculates the required KLM operators first to reach and then to manipulate (fill) each one of the elements. The WKLM-FA supports all the operators of the original KLM [1] –K for keystrokes, B for mouse button press or release, P for pointing to a target on a display with a mouse, H for homing the hand(s) on the keyboard or other device, and M for mentally preparing for executing physical actions– except for the D (drawing) operator, which is not an action performed in the context of form filling tasks. Finally, it sums up the operators and calculates the total time by replacing each operator with a particular predefined and user-configurable numeric value. WKLM-FA can optionally employ the Fitts's Law to calculate the numeric value for the P operator more accurately, which in turn leads to a more accurate overall prediction.

The WKLM-FA supports user accounts, but it also allows non-registered use of the application for free. Registered users may save their projects and continue working on them at any time, while non-registered do not. An admin panel in WKLM-FA (Figure 2) depicts usage statistics of the evaluated web forms. Moreover, WKLM-FA provides a flexible way to define various parameters that impact the modeling results. First, the user of the tool can select which form elements should be used based on the simulated form filling task. The tool also provides functionality to map a text element of the evaluated form to a pre-declared element name (e.g., email, password) that comes with a research-based estimated number of characters to be filled. This number of characters can be also specified by the evaluator to fit a specific user interaction scenario. Also, the WKLM-FA enables defining the mouse or keyboard as the interaction device used for reaching or manipulating the form elements. The WKLM-FA also supports setting up the initial state for the simulated cursor and user's hands. Furthermore, the evaluator can change the simulated user's age and typing ability and the WKLM-FA updates the results in real-time. Finally, users of the WKLM-FA can manage and apply on demand various KLM rules, such as when a mental operator (M) is needed and they can also modify the numeric values of the KLM operators and Fitts's law



Admin Dashboard

EVALUATIONS (TOTAL) 126

EVALUATIONS (MONTHLY) 94

TOTAL SAVED PROJECTS 6

REGISTERED USERS 7

Projects

Show 10 entries Search:

Form Name	User	url	DateCreated	DateModified
	GlobalUser			
amazon	fotis	https://www.amazon.com/ap/register%3Fope	14/7/2020 9:23:58 μμ	25/9/2020 8:21:14 μμ
dropbox	fotis	https://www.dropbox.com/register	25/9/2020 5:22:17 μμ	25/9/2020 5:22:17 μμ
easyjet	fotis	https://www.easyjet.com/en/Register	25/9/2020 5:16:06 μμ	25/9/2020 5:16:06 μμ
foursquare	fotis	https://foursquare.com/signup/join	14/7/2020 9:23:41 μμ	24/9/2020 7:53:05 μμ
test-fb	dkarousos	https://facebook.com/login	6/8/2020 1:26:58 μμ	7/8/2020 12:24:56 πμ
Form Name	User	url	DateCreated	DateModified

Showing 1 to 6 of 6 entries

Figure 2: The WKLM-FA admin dashboard page presenting tool usage statistics.

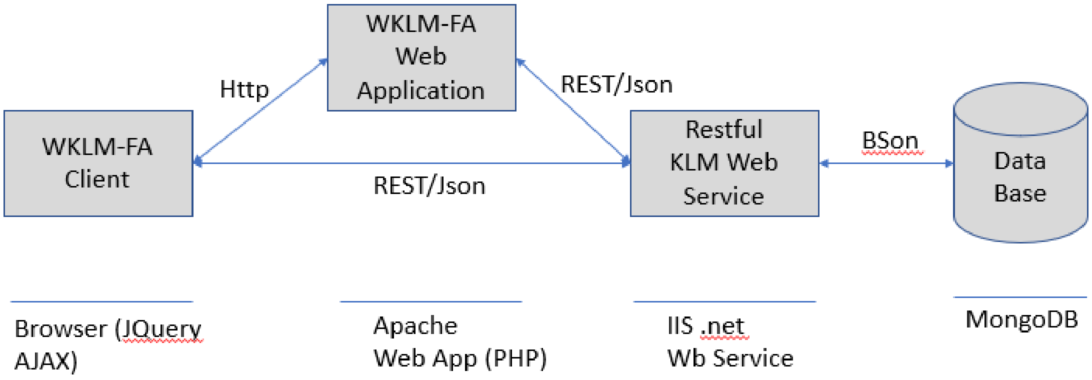


Figure 3: Software modules analysis of the WKLM-FA architecture

constants. The WKLM-FA is temporarily available online[1] and will soon be hosted in a dedicated server with a static domain name.

3 DESIGN AND IMPLEMENTATION ISSUES

Based on previous user studies [6-8], our goal was to move from the desktop application to a web-based one. Having in mind the complexity of the underlying data model of KLM-FA and the need to support a rich UI environment, we designed an n-tier architecture in which the application functionality is provided by three separated modules: a) the UI module, which handles the interaction with the user, b) the Form Analyzer module, which parses a Web form and identifies its KLM-related elements, and c) the KLM module, which applies both the KLM and Fitts's Law to a list of form elements and estimates the required time to fill them. Finally, we use a Storage Layer to store and retrieve all the appropriate information (Projects, Evaluations, Accounts, etc.).

As it is depicted in Figure 3 and considering the data model layer, the WKLM-FA uses XML/Json to describe: a) the list of form elements, b) the list of application settings (i.e., modeling rules, user preferences and KLM algorithm parameters), and c) the information for storing and retrieving a project. All these data are encapsulated by a single XML document that is transformed to BSON and then

[1] http://hci.csd.auth.gr/wklmfa

Table 1: Comparison of functionality between two versions of KLM-FA

No	Functionality	KLM-FA (Desktop Version)	WKLM-FA (Web Version)
1	KLM Algorithm	YES	YES
2	Fitts's Law	YES	YES
3	KLM and Fitts's Law Parameters	ALL	ALL
4	Video Preview (Watch and Learn)	YES	NO, under Development
5	Project manipulation (store, retrieve)	YES	YES
6	Multiple instances of application	NO	YES, in Web Browser TABS
7	Supported platform	WINDOWS	ALL, through Web Browser
8	Supported Web Forms	MANY	MANY (fewer than KLM-FA)
9	Web Form manipulation (dynamically change elements, redesign/re-render web form, inject JavaScript, etc.)	SUPPORTED but not implemented	PARTIALLY SUPPORTED due to security issues
10	User accounts	NO	YES
11	Projects statistics	NO	YES
12	Reporting	NO	YES

stored to a NoSQL database. The process is reversed when retrieving a saved project.

The next layer is a RESTful web service developed in c# and .net framework and operates through an IIS web server. This kernel layer implements the business logic of the KLM model. First, it applies the KLM and Fitts's Law to a project document, and then it manipulates projects (store, receive, keep log, etc.) by interacting with the storage layer. This implementation allows us to add, in the future, new types of client applications to the system (e.g. mobile app, browser add-ons, etc.) and connect them to the kernel. Hence third-party apps may easily run the modeling algorithm.

The subsequent layer of the web app consists of a PHP and Javascript/JQuery project which provides the UI and the module for web page parsing. Finally, the source project is deployed by the apache web server. In this way, many REST requests to the business logic Web Service are taking place both from the web server side and from the web browser (AJAX calls) as well.

3.1 Dealing with the web page preview

Each time a user selects a webpage by typing its URL, the WKLM-FA must load it and then render the webpage which contains the form and finally embed it to the WKLM-FA main page. For this purpose, an iframe named "Web Browser Preview" is used. In HTML, the iframe element is used inline within a normal HTML body but during the last years, the use of frames has been made obsolete due to usability and accessibility concerns. The feature of framesets has been removed from the HTML5 standard. The iframe element, however, remains with several "sandboxing" options intended for sharing content between sites, such as blog comments.

Websites use techniques to avoid clickjacking attacks by ensuring that their content is not embedded into other sites. Furthermore, nowadays, almost all web servers are using by default the X-Frame-Options: DENY. So, the WKLM-FA cannot load external websites into an iframe. However, the tool needs to show a preview of the evaluated webpage, but it does not need to have any complicated interaction with it. To achieve this, the WKLM-FA parses the html source code of the evaluated webpage. In most cases, however, this collection of code would render in a browser without the associated CSS and thus would not look like the actual evaluated webpage. This is typically caused by using relative paths for CSS files on websites. To overcome this limitation, the WKLM-FA converts relative paths to absolute paths in the header section of the parsed webpage.

In sum, WKLM-FA shows the evaluated webpage by parsing the HTML source code, converting the absolute paths of CSS to relative ones, and rendering the result to the "Web Browser Preview" iframe. So, in most cases, the rendering of the webpage in WKLM-FA is identical to the actual webpage.

3.2 WKLM-FA evaluation

The WKLM-FA has been built based on the algorithmic logic and user interface design of KLM-FA, our previously developed desktop application. The KLM-FA application has been already evaluated in terms of its accuracy and validity [4, 5]. Thus, the WKLM-FA algorithmic logic was only tested against the KLM-FA one. All test results were identical in terms of KLM modeling and predicted form completion time.

As far as WKLM-FA's usability is concerned, it has not been evaluated until now. A quite notable issue that emerged with the WKLM-FA is the size limitation of the web browser preview. This preview is smaller (more often in width) in comparison to the preview in the desktop version. Thus, some explanations of the detected form elements and their operations may not appear with the appropriate clarity. A user testing study is required to answer if the latter has a detrimental effect on the overall usability of the tool. However, apart from the beforementioned issue, it is expected to be similar to KLM-FA (i.e., "Good to Excellent" [8]), given that WKLM-FA has been designed by the same team and it adopts the same user interface design principles.

4 DISCUSSION AND FUTURE WORK

The current version of the KLM-FA is now available through the Web, so it can be considered as a cross-platform version. Comparing it with the previous desktop application we ensured that this version produces accurate results and we expect it to be equally useful as previous studies have found [8]. Table 1 summarizes the comparison results for WKLM-FA and KLM-FA in terms of the provided functionality.

Most of the implementation challenges which we faced were related to bypassing security constraints enforced in the Web. Despite our efforts, there are still some webpages that cannot be parsed and manipulated by the WKLM-FA tool due to the complexity of the technologies used to dynamically produce them. Moreover, WKLM-FA renders the web forms inside the application page, whereas the KLM-FA tool used an internal web browser to do so. Hence, the KLM-FA was able to evaluate most of the existing web forms. As discussed previously, the size limitation of the web browser preview might negatively influence some aspects of the user experience, and it should be further examined.

We believe that this version of the WKLM-FA would be more suitable for educational reasons (e.g., for teaching KLM or Fitts's Law) -by selecting suitable web forms- than for large-scale web form evaluations. The latter might be better supported by developing the application as a browser add-on. In this way, we will be able to: a) exploit the entire web browser area for rendering the web form, b) better support form (re)design since we will have full access to the web form and thus be able to redesign it in run-time, manipulate its elements and calculate the optimal KLM time, and c) provide quick and easy time estimations by simply activating the addon in the browser, a web developer's native environment.

5 CONCLUSIONS

A new web-based version of a desktop tool, offering KLM functionality to both HCI experts and educators, was implemented. This version, named WKLM-FA, is a platform independent web-app and it maintains all the rich features of the previous desktop version, while it is enhanced with account control and usage analytics as well. The multi-layer architecture of this web version makes possible the implementation of third-party applications that use KLM modeling as a web service. Our future work focuses on developing the application as a browser add-on to alleviate security constraints enforced in the Web ecology.

REFERENCES

[1] Stuart K. Card, Thomas P. Moran, and Allen Newell. 1980. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM* 23 (7), 396–410.

[2] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47 (6), 381.

[3] Bonnie E. John, Konstantine Prevas, Dario D. Salvucci, and Ken Koedinger. 2004. Predictive human performance modeling made easy. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Vienna, Austria, 455–462.

[4] Nikolaos Karousos, Christos Katsanos, Nikolaos Tselios, and Michalis Xenos. 2013. Effortless tool-based evaluation of web form filling tasks using Keystroke Level Model and Fitts Law. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13), ACM, New York, NY, USA, 1851–1856.

[5] Christos Katsanos, Nikos Karousos, Nikolaos Tselios, Michalis Xenos, and Nikolaos Avouris. 2013. KLM form analyzer: automated evaluation of web form filling tasks using human performance models. In *Human-Computer Interaction – INTERACT 2013* (Lecture Notes in Computer Science), Springer Berlin Heidelberg, 530–537.

[6] Christos Katsanos, Nikolaos Tselios, Nikolaos Karousos, and Michalis Xenos. 2015. Learning web form design by using the KLM Form Analyzer: A case study. In *Proceedings of the 19th Panhellenic Conference on Informatics* (PCI '15), ACM, New York, NY, USA, 44–49.

[7] Christos Katsanos, Michalis Xenos, and Nikolaos Tselios. 2018. Tool-mediated HCI modeling instruction in a campus-based software quality course. In *Proceedings of the 20th International Conference on Human-Computer Interaction, HCI International 2018* (Lecture Notes in Computer Science), Springer International Publishing, 114–125.

[8] Christos Katsanos, Michalis Xenos, Nikolaos Tselios, and Nikos Karousos. 2020. Tool-mediated HCI modelling instruction: evidence from three studies. *Behaviour & Information Technology.*

[9] Evan W. Patton and Wayne D. Gray. 2010. SANLab-CM: A tool for incorporating stochastic operations into activity network modeling. *Behavior Research Methods* 42 (3), 877–883.

[10] Cogulator: A Cognitive Calculator. Retrieved September 26, 2020 from http://cogulator.io/index.html

[11] Keystroke Level Model (KLM) Calculator. Retrieved September 26, 2020 from http://courses.csail.mit.edu/6.831/2009/handouts/ac18-predictive-evaluation/klm.shtml