Modelling Learning of New Keyboard Layouts

Jussi P.P. Jokinen[1], **Sayan Sarcar**[2], **Antti Oulasvirta**[1]
Chaklam Silpasuwanchai[2], **Zhenxin Wang**[2], **Xiangshi Ren**[2]
[1]Aalto University, Finland [2]Kochi University of Technology, Japan

ABSTRACT

Predicting how users learn new or changed interfaces is a longstanding objective in HCI research. This paper contributes to understanding of visual search and learning in text entry. With a goal of explaining variance in novices' typing performance that is attributable to visual search, a model was designed to predict how users learn to locate keys on a keyboard: initially relying on visual short-term memory but then transitioning to recall-based search. This allows predicting search times and visual search patterns for completely and partially new layouts. The model complements models of motor performance and learning in text entry by predicting change in visual search patterns over time. Practitioners can use it for estimating how long it takes to reach the desired level of performance with a given layout.

ACM Classification Keywords
H.5. Information Interfaces and Presentation (e.g. HCI): Theory and methods

Author Keywords
Visual search; keyboard layouts; models of learning

INTRODUCTION

Understanding learnability and learning of user interfaces (UIs) has long been a core objective in HCI research. Encountering new, updated, or redesigned interfaces is characteristic of contemporary computer use. For a user, this directs resources toward constant learning and relearning. For design and innovation, it is crucial to understand how users weigh the expected benefits of a UI against costs such as learning effort [62]. History is replete with interfaces rejected on account of high perceived learning effort, such as attempts to tweak or more radically depart from the Qwerty layout [22, 20].

The objective with this paper is to advance research on accurate yet practical HCI models (e.g., [7, 12, 14, 15, 30, 36, 35]). Appropriate models of learning would not only inform design and decision-making but be suited to direct implementation in design algorithms and tools [8, 59, 60]. However, learning is a complex phenomenon involving perceptual, attentive, motor, and cognitive reorganisation. While there are plenty of empirical studies of learning in HCI (e.g., [19, 23, 47, 48, 53]), scholars have made relatively limited progress toward synthesising the findings from these into models that can drive practical endeavours.

This paper presents a model of *positional learning for keyboards*, examining how users learn to locate keys on a keyboard. Keyboards are the primary method of data entry for computer platforms and have been a central topic in HCI research for decades. Computer users frequently encounter new or changed keyboard designs, and in any such encounter, visual search performance is a critical factor in how well the user completes tasks involving the layout. Accordingly, how users learn a new layout is a well-recognised issue in the study of keyboard layouts [21, 22, 33, 49, 64].

The model presented here takes two software keyboard layouts and associated user histories as input. It predicts changes over time in both visual search patterns and search times with the new layout. For each task, the model simulates a series of fixation points and saccades until the target is found. The model predicts how gaze and total search time depend on various factors, such as the number of keys, their organisation into rows and columns, whitespace among them, and viewing distance. Figure 1 demonstrates the model's predictions in a scenario wherein a few keys in the familiar Qwerty layout are swapped. Firstly, the model is trained to be expert in finding keys in the Qwerty layout; this is visible in decreasing keysearch times. After the swap, there is an impact on immediate search times and a subsequent relearning of the new layout, again manifested in decreasing search times.

The model's structure is rooted in modern understanding of positional learning. In particular, it assumes that positional learning is a 'race' between two systems of human memory to control the oculomotor system. Two systems, visual shortterm memory (VSTM) and long-term memory (LTM), are needed to explain how learning changes and how the eyes move over a keyboard [4, 6, 17, 44, 34]. Firstly, when one looks for a key for the first time, search is not random but governed by **1)** VSTM, which inhibits the revisiting of previous locations. Without this assumption in the model, search times are unrealistically long. Secondly, with repeated use of a key, **2)** LTM starts to construct a representation of its spatial location. The strength of the association between a key and its location affects how quickly and reliably this location can be retrieved in the future. Eventually, the user will find the key by gazing at it directly with one glance [13]. At this level of

Copyright is held by the owner/author(s).
CHI 2017, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4655-9/17/05.
http://dx.doi.org/10.1145/3025453.3025580

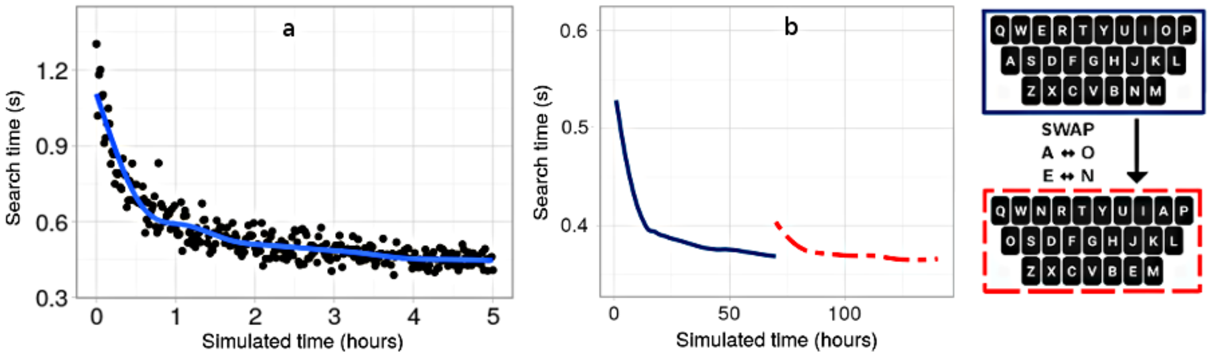

Figure 1. The model predicts how users learn the positions of keys on a new keyboard or one that has changed partially. Pane a shows how average key search times decrease as the model learns the layout. In pane b, the model gradually relearns a layout with two swapped pairs of keys. Pane a contains a subset of the whole data, presented in pane b.

skill, most task time is spent in motor performance, selecting the keys. The model described here does not cover this aspect; it focuses instead on the change in visual search times in the early stages of learning. To explain how changes in a layout affect learning, the model employs **3)** utility learning, which allows adapting to changes in the model's environment. With utility learning, the model can store multiple locations for the same key in LTM and still, in accordance with the layout, retrieve the correct location.

This model complements models of learning in HCI. Existing mathematical models are straightforward to apply but reveal fewer details about learning. In particular, our model goes beyond the most widely used mathematical model, the power law of learning [7, 14, 18, 15, 46], by modelling how search patterns change and by predicting the effects of specific changes to the layout on search times. To account for the time- and content-contingent phenomena involved, it includes a production system, which controls the oculomotor system by means of input from the two memory systems. This assumption is similar to those in ACT-R and EPIC [3, 12, 36, 35] but tailored to the case of 2D layouts. Hence, we apply not a universal model of the user but a simplified task-specific model used for predicting visual search. The benefits are that 1) complexity is reduced, 2) explanations for observed behaviour can be better identified, and 3) the model is easier to use.

For development and evaluation of the model, two datasets were collected. The first comprised visual search patterns and search times among users encountering a completely new keyboard layout and learning to search its keys over four days. The second dataset was from a more naturalistic task, wherein the participants, already familiar with the Qwerty layout, were given a modified Qwerty layout and asked to type words with it on a smartphone. This dataset was used for fitting the relearning parameters in the model. Results from several evaluations of the model are presented.

The model can serve as a tool for anticipating users' learning experience when practitioners tackle design problems such as:

- **Comparison:** Which of the given layout alternatives has the lowest learning costs?
- **Immediate cost:** What is the impact of layout changes on visual search performance immediately after the change?
- **Learning time:** How long does it take the user to learn a new layout to some desired level of visual search?

RELATED WORK AND GOALS

Whilst there is extensive work on modelling of learning in skilled activities [1, 3, 5, 4, 45, 61, 29] and visual search [31, 55, 34], models directly applicable to the domain of keyboards are lacking. Here we present a review of empirical findings on the effects of changing a keyboard, conducted to compile a list of phenomena that were taken as goals for the modelling effort and for discussing how existing models address these.

Empirical Studies of Layout Learning

Prior work has established that learning accounts for a large proportion of user performance in typing. In a study where the keyboard was randomised after each keypress, the attained performance (5.5 words per minute; WPM) was inferior to that in a condition wherein an unchanged Qwerty layout was used (20.5 WPM) [42]. This finding suggests that efforts to model visual search are important for understanding the development of typing performance.

The effect of changing a keyboard has constituted a major issue in the development of optimised layouts. Results are mixed. When words are learnt repetitively, an optimised keyboard may perform better than Qwerty [11, 10]. However, if full phrases are used, relearning takes longer and it may not even surpass Qwerty in the course of an experiment. Dunlop et al. [21] reported that *Sath* – a layout optimised for familiarity and word disambiguation – did not outperform the Qwerty layout in learning during their study. Upon change, speed dropped from 21 WPM to 13 WPM, though it rose to 18 WPM after about four hours. Such results highlight the importance of understanding how relearning is conditioned to the previous layout and how quickly a set desired level can be reached.

In attempts to explain the negative impact of layout change on typing performance, several causes have been pointed out: visual search for keys [42, 57, 66], unfamiliar arrangement of the keys (an alphabetical arrangement yields shorter search times than Qwerty) [65, 41], phrase set [66], and the number of keys affecting choice [63] (according to the Hick–Hyman law [25, 28]. See the work of Keele [32] for a review. Such effects are also influenced by top-down learning strategies that users may adopt. In a study wherein participants were taught relative spatial locations of keys before typing with a novel keyboard [38], this top-down approach improved performance (to 14.9 WPM) relative to that of a control group typing without top-down information (12.0 WPM).

Models of Visual Search and Learning

Models of learning in HCI can be classified into two types: mathematical and simulation. *Mathematical models* of learning describe change in a dependent variable as a function of repetitions or time. The classic power law of learning predicts how performance improves as a power function of repetitions [46]. While this forms the predominant mathematical model of learning in typing [18, 16, 41, 66] and menu interaction [7], it does not inform us about how users search or the way their performance is affected when a layout changes [50].

Our model builds instead on ideas from *simulation models*, step-wise executable architectures describing both process and infrastructural aspects of cognition [3, 12, 36, 45]. The former element is defined via production rules or users' declarative knowledge, while the latter is composed of processing units with unique properties.

EPIC (Executive Process Interactive Control) [36, 35] models visual search in menus by means of a production-rule system and perceptual-motor peripherals. The assumption is that execution is constrained by certain temporal and capacity limitations. Four search strategies for menus are covered: serial search, parallel search, random search, and systematic search. A blend of random and systematic search was found to offer the best match. A recent paper extended the general model with an active vision component that can deal with visual cues such as colours and shape [35]. Another production-rule architecture is *ACT-R*, which uses a different visual search process. For menu interaction [12], it assumes that eye fixations operate from top to bottom on items that share features with the target.

The original goal for EPIC was to predict the effect of perceptual, motor, and cognitive constraints on human action; therefore, it simulates expert performance rather than learning [36]. Conversely, the roots of ACT-R are in list learning [1, 5], and this architecture integrates various learning mechanisms, such as LTM, utility learning, and production compilation [3, 4]. The model presented here uses learning mechanisms similar to ACT-R's but in a more context- and task-constrained way, and the other components and assumptions from ACT-R theory are not applied. More specifically, we utilise the LTM and utility components from ACT-R but streamline them to the context of learning keyboard layouts. Further, we use a model of eye movements [55] that, while compatible with ACT-R, is not part of its core theory. A system for persistent visual storage for visual search tasks is adapted from EPIC [34].

Goals for This Paper

For the purpose of better supporting practical work in designing layouts for learnability, the goal here is to strike a balance between mathematical and simulation models. The former have a more restricted scope with regard in learning but are simpler to employ. These models typically have free parameters that need to be fixed with case-specific empirical data. Models of the latter type have a broader scope but are more complex and demand filling in many details in advance.

With the model presented here, our aim is to cover key phenomena pinpointed by empirical studies and previous models: 1) change in visual search performance and patterns with experience, 2) the effect of a previous layout, and 3) the effect of keyboard design and the number of keys. The decision to focus on these aspects leaves some others to future work. For example, there may be many search strategies that novices use when learning a new layout, such as fixating first on the corners of the layout or progressing from one key to another in a systematic fashion. Also, layouts may involve visual cues not covered here, such as key highlighting [43].

MODEL OVERVIEW

The input to the model is a keyboard layout, defined as (x,y) positions of keys, and a viewing distance. It outputs a sequence of (x,y) positions of fixations with timestamps, along with the total search time for a cued key, given a starting location for the search task and the target key. Each search task leaves a memory trace, which the model attempts to utilise in the next tasks. The keyboard layout can be changed 'on the fly', in which case the memory from the earlier layout is retained and hence affects subsequent relearning times.

An overview of the model's components is given in Figure 2, with the parameters listed in Table 1. The overarching assumption is that the total search time is determined by the interplay of four components:

- **Vision:** Given a command of where to attend, the model attends to that location, possibly moving the eyes towards the requested position via saccades. The model assumes no peripheral vision for shapes and colours, but it does simulate the fact that items close to the current fixation can be encoded without a saccade.
- **Visual short-term memory:** Each encoded key location is placed in the VSTM for a short time, inhibiting revisiting of this key as long as it is in the VSTM.
- **Long-term memory:** After the key has been found, an entry associating that key with its spatial location is made or, if one exists already, refreshed. With each new task, the model can attempt retrieval of the location of the key.
- **Controller:** The controller requests attention shifts from the visual system, and it also attempts to retrieve the target's location from LTM. If there are multiple locations associated with the target, as may be seen in the case of a keyboard switch, *utility calculation* is used to determine which memory entry to try to recall.

Vision

The model's visual search component follows the EMMA integrated model of eye movements and visual encoding [55]. Time to encode a key T_e is

$$T_e = K \cdot [-log(f)] \cdot e^{k \cdot \varepsilon}, \tag{1}$$

where K and k are constants, f is the frequency of the object (e.g., monogram frequency), and ε is the eccentricity, measured as the distance of the target from the current fixation

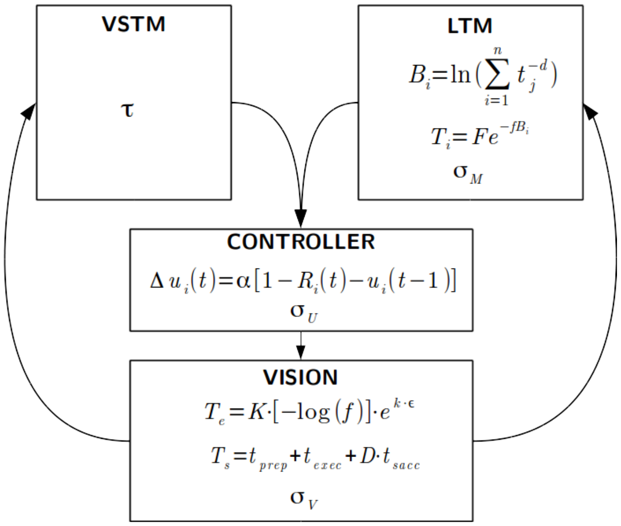
VSTM
τ
LTM
$B_i = \ln(\sum_{i=1}^{n} t_j^{-d})$
$T_i = Fe^{-fB_i}$
σ_M
CONTROLLER
$\Delta u_i(t) = \alpha[1 - R_i(t) - u_i(t-1)]$
σ_U
VISION
$T_e = K \cdot [-\log(f)] \cdot e^{k \cdot \epsilon}$
$T_s = t_{prep} + t_{exec} + D \cdot t_{sacc}$
σ_V

Figure 2. Guided by visual short-term memory and long-term memory systems, the controller requests attention shifts from the vision module. The positions visited are stored in the VSTM, which inhibits the controller from revisiting them. Target position is stored in the LTM for future retrieval attempts. In the event of conflicting positions in LTM, utility learning is used to simulate learning of new locations.

(in degrees). Because encoding time increases exponentially as a function of eccentricity, the visual system may initiate a saccade to get closer to the target. Saccade duration T_s is

$$T_s = t_{prep} + t_{exec} + D \cdot t_{sacc}, \tag{2}$$

where t_{prep}, t_{exec}, and t_{sacc} are constants related to the human visual system and D is the distance to be covered by the saccade, in degrees. The landing point of the fixation is the target location with noise added from a normal distribution with a standard deviation of σ_V times the distance between the saccade's starting point and the intended landing points [55]. If the encoding time (see Equation 1) is less than t_{prep}, the target is encoded fully without a saccade; if not, the remaining encoding is done after the saccade.

The EMMA model has been successfully used for simulating gaze patterns in various tasks, such as reading [55], menu search [55], and driving [56, 37]. Its main benefit is that it can encode targets without necessarily having to make a saccade. Because not all keys that are encoded during visual search need their own fixation, it becomes more economical to search through the keys in a given area first before moving via a saccade to another section of the layout than it is to jump randomly between keys across the entire layout. Future versions should be able to use the utility learning of the controller component to predict this behaviour; currently, the controller only requests local attention shifts, unless inhibited by the VSTM.

Visual Short-Term Memory

VSTM attempts to direct the eyes to search neighbouring keys by keeping track of previously visited locations and inhibiting shifts to them. In a normal keyboard layout, the number of immediate neighbours is 4 for keys in the central area, 3 for those at the sides, and 2 for those in the corners. Inhibition is implemented as a time rule: τ indicates how long, in seconds, an item will be stored in the VSTM [34]. If the model is forbidden to move into any direction for reason of VSTM inhibition, it requests an attention shift towards a random non-inhibition-influenced key.

VSTM is necessary because it allows the model to build an overview of the layout as it surveys it for the first time [27, 35, 34]. Research on visually intensive search tasks has shown that an assumption about a short-term memory for previously visited objects is necessary for accurately predicting visual search [27, 34]; assuming that visual search is random would lead to unrealistically high search time. Here, the external features of the object are not considered in the simulation, as the keyboard's keys are not distinguishable from each other by shape or colour.

Long-Term Memory

In parallel with VSTM-guided visual search, the model attempts to retrieve the location of a target from LTM. After the target key has been attended to and encoded, either by VSTM-guided search or on account of retrieval from LTM, the location of the target key is refreshed in LTM for future retrieval attempts. When the model has found the cued key, its visual location is stored in LTM or, if the memory for the key's location already exists, the memory entry is refreshed. Each time this entry i is revisited, upon either recall or fixating again on the key, its activation B_i is updated [5]:

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d}), \tag{3}$$

where t_j is the time since the jth visitation of i and d is a decay parameter [5]. In the beginning of the task, the model tries to retrieve the visual location of the cued key from its long-term storage. If $B_i > 0$, with added noise from the logistic distribution with a standard deviation of σ_M, the location is retrieved and the controller requests an attention shift to that point (any ongoing visual processes, such as saccades, need to be completed first). This essentially completes the task as soon as the visual process, as described above, is finished. If a noisy fixation lands far from the target, a correcting second fixation is required. The time to retrieve location T_i depends on its activation and is obtained as

$$T_i = Fe^{-fB_i}, \tag{4}$$

where F and f are individual scaling constants [5].

LTM predicts the way in which repeatedly associating a given key with a spatial location increases the likelihood of recalling the location. Furthermore, the model predicts higher recall latencies early in learning and very short recall times in the expert phase, when the user has had extensive training in the layout.

The model for LTM presented here has been studied extensively for decades [1, 5] and shows a good fit in list-learning experiments [1] and learning the positions of visual objects [51]. For memory tasks, it produces learning curves similar to those yielded by the power law of learning while allowing the modeller to investigate the effect of the number of task objects on learning. This is a necessity for a model of visual search of keyboard layouts, which should be able to deal with different layouts without requiring refitting of parameters.

Table 1. Parameters of the model and their descriptions, where values adopted from the literature are denoted with 'a', and empirically fitted values are in boldface (LTM parameters were fitted in study 1, and controller parameters in study 2).

Parameter	Component	Elements controlled	Value	Ref(s).
τ	VSTM	decay (forgetting)	4.0^a	[34]
d	LTM	decay (forgetting)	0.5^a	[5]
F	LTM	a scaling factor for retrieval times	**1.06**	[5]
f	LTM	scaling of the effect of activation on retrieval time	**1.53**	[5]
σ_M	LTM	noise for retrieval probability	**0.60**	[5]
α	controller	a scaling factor for learning partially changed layouts	**0.02**	[3, 52, 61]
σ_U	controller	noise for choosing between conflicting memory retrievals	**0.25**	[3, 52, 61]
K	vision	a scaling factor for encoding speed	0.4^a	[55]
k	vision	a scaling factor for the effect of object eccentricity on encoding speed	0.006^a	[55]
t_{prep}	vision	visual pre-processing time	0.200^a	[55]
t_{exec}	vision	saccade execution time (baseline)	0.070^a	[55]
t_{sacc}	vision	a scaling factor for saccade execution time (seconds/deg.)	0.002^a	[55]
σ_V	vision	saccade landing-point noise	0.1^a	[55]

Relearning Partially Changed Layouts

After a layout is learnt well, any change negatively affects the performance of the model, as it still looks for the cued key in the old location. Relearning, therefore, is in part a separate phenomenon from learning a completely new layout. In relearning and encoding of a key in a location, LTM records a memory entry that associates the key with its location. If such an entry already exists, this entry is activated again, as indicated in Equation 3. However, if the key layout has changed, the new location is added as a new entry. After this, for each retrieval request there are two conflicting entries, of which the one with more activations is returned. Thus, after a key has been relocated, the new location starts slowly gaining association, while the old one begins to decay in memory. The model uses *utility learning* to help it choose between the old and the new memory retrieval for a given key. After the model has recalled a position and eventually (sooner or later) found the key, the utility (u_i) of that recall is updated via *the delta learning rule* [52, 3, 61]:

$$\Delta u_i(t) = \alpha[1 - R_i(t) - u_i(t-1)], \tag{5}$$

where α is a parameter to be estimated; $R_i(t)$ is a reward, which is calculated as the temporal distance between the recall and finding of the target (shorter temporal distances being more rewarding); and $u_i(t-1)$ is the utility in the previous step. The recall with the greatest utility, after adding of noise from a logistic distribution with standard deviation σ_U, gets retrieved.

Utility learning is necessary if the model is to adapt to changing layouts. The delta rule has been shown to offer a good model of rational action [3, 29]. It is reasonable to assume that, also in relearning, users try to maximise the efficiency of their actions by being critical of their memory of the old layout and adjusting their retrievals accordingly. However, it should be noted that the model does not provide a top-down framework for shifting between multiple familiar layouts. A user with knowledge of both Qwerty and Dvorak will be able to shift between the layouts without too much memory interference. Our model does not have such mechanism.

Parameter Acquisition

A key goal in the work described here was to ensure reasonable predictive power without having to fit too many parameters empirically, a goal seen with other simulator-based models too [2, 36]. All parameters of the model and their descriptions are shown in Table 1. Although the model appears to have numerous parameters, most values are fixed *a priori* in line with previous literature. In particular, all values for VSTM and the vision component are adopted from the literature, as they reflect relatively stable aspects of cognition and perception.

What remains is a set of three LTM-related parameters (F, f, and σ_M) and two controller-related ones (α and σ_U). These parameters are context-dependent, so experimentation is needed for finding plausible values for them. We fitted the values in the context of learning visual search with touchscreen layouts: two empirical studies are reported on below. The first entailed fitting the two LTM parameters, which we then assumed to be the same for the second study, wherein we fitted the controller-related parameters. Finally, we tested the generalisability of the model, with the fitted parameters, against results of an independent previous study [21].

In summary, our aim is a model that can be used directly, after parameter fitting, for soft keyboards on touchscreen devices. However, the model can be calibrated to other contexts involving visual search of 2D surfaces. The procedure for parameter fitting and model validation is:

- **Forward modelling:** Set parameters of the vision and VSTM components based on literature.
- **Inverse modelling:** Obtain the rest of the parameters in two controlled studies (see below).
- **Validation:** Validate generalisability against results obtained previously.

STUDY 1: LEARNING A NEW LAYOUT

The first experiment dealt with learning a completely new layout. The aim was to fit parameters of the LTM model to data and provide initial validation of the model. To this end, the experiment was simple, involving only visual search tasks with new keyboards, without motor action such as typing, to produce data on learning-related changes in visual search times and fixation patterns. The experiment followed the idea of using cued reaction times to measure visual search time on keyboard layouts [57], and it used randomised keyboard layouts. The experiment used two user groups (young and older adults) and spanned four days in order to provide a more extensive dataset for parameter acquisition. Because previous work has shown that cognitive performance as well as typing speed is affected by age, calibrating and validating the study only with young university students would be limited [24, 54]. For this reason, we chose a more representative sample.

Method

Participants: In total, $N = 33$ people were recruited for the experiment. Half of the participants (17) were older adults (61–71 years old, $M = 68$, $SD = 3$), and half (16) were young adults (19–35 years old, $M = 25$, $SD = 6$).

Procedure and Task: The experiment spanned five consecutive days and employed the structure shown in Table 2. On days 1–4, the participants conducted visual search tasks with the layout assigned to them on the first day, and on day 5 they transcribed text by using a smartphone. On the first day, each participant was assigned an individual layout, with the 26 English Qwerty keys randomly shuffled. This layout was used for all visual search tasks from days 1 to 4 except when, for the first of the two blocks on day 1, the subjects performed tasks with layouts that were randomly shuffled between all trials (however, data from the latter searches were not used in the model fitting reported on here). There were, in all, 10 blocks (one on the first day and three on each of the following three days), of 15 minutes each, in which the participants conducted visual search tasks with their assigned layouts.

A visual search task block, which was 15 minutes long, involved searching for cued keys on a picture of a keyboard. Firstly, the participant was shown a cue, which was one of the 26 letters. The cue was shown on an otherwise blank computer screen; the duration for displaying it was randomised between 2.5 and 3.5 s so that participants could not anticipate the search task. In addition, they were required to look directly at the cue: if the gaze, as tracked by an eye tracker, strayed from the cue, the timer was reset.

After the cue letter disappeared, a picture of a keyboard, always in the same location, was shown. The participant's task was to locate the cued key as quickly as possible and press a reaction time (RT) button. After pressing, a new cue was presented immediately. The cue was positioned either on top of the area where the keyboard would appear (33% of the time) or in a random place within the area where the keyboard would appear (the rest of the time). The reason for this was to simulate two cases: the user's gaze entering the keyboard from a typing area above it and the gaze arriving from within the keyboard space. The participants completed as many trials as they could during the 15 minutes allotted for each block. They were told not to cheat by pressing the RT button before actually visually locating the key, and that the eye-tracking equipment would be used to ascertain whether they had. No clear signs of such behaviour were evident in the data.

Table 2. The experimental design in study 1, with motor-calibration tasks (M), visual search with a random keyboard (R) and an assigned keyboard (A), and transcription tasks with a touchscreen mobile phone (T) (each block, apart from M, was 15 minutes long).

Day	Blocks
1	M + R + A + M
2–4	M + A + A + A + M
5	T + T

Motor calibration task: At the start and the end of each day encompassing visual search, participants performed a brief motor-calibration task. The subject was shown a large circle on the monitor for a random time within the range 5–10 seconds. The task, repeated five times, was to press the RT button as soon as the circle disappeared. The resulting motor-calibration data were used to adjust the search times recorded in the visual search experiment: each visual search time was reduced by the mean of the median RT from the two motor-calibration tasks carried out on the day in question.

Typing speed assessment: On the final, fifth day, the participants completed a transcription task. A Samsung Galaxy 6 smartphone was used for the experiment (screen size 5.1 inches (70.7% screen-to-body ratio), with resolution 1440 × 2560 pixels (577 ppi); orientation: portrait; placement in a case positioned at 45° elevation from the table surface on which the case was rested). The typing software presented the participants with random words from a corpus, or one or two random letters from that corpus (composed of the 200 most commonly used English words), as shown in Figure 5. The participants completed as many trials as they could in two 15-minute blocks with a short rest in between. Half of the participants, balanced between the age groups, conducted the transcription task with the keyboard layout that had been assigned to them on the first day of the experiment, while the other half were given a new randomised layout. The typing-speed task did not involve eye tracking, because it was performed on a smartphone, with which using the available tracker was not possible.

Apparatus and Analysis

Search times: We tested the hypothesis that search time decreased by task block by performing multilevel regression analysis via the LME4 package in R [9], with RT as a dependent gamma-distributed variable, task block as the fixed term, and participant as a random term. The benefit of using a multilevel regression is that it works better with nested data, such as data from repeated trials [26]. All negative search times observed were discarded as accidental keypresses with associated very short search times, which became negative after the motor-calibration subtraction. The number of these was very small (0.5%), however, and we consider these observations to be distributed at random.

Gaze tracking: The eye movements of the participants were recorded during all visual search tasks using Tobii X120 eye tracker with a sampling rate of 120 Hz. The resolution was set to 1680 × 1050 pixels (17 inch LCD screen). All eye tracking analyses were conducted to data, that begun when the cue letter disappeared and the keyboard appeared, and ended at the RT of that trial, with the motor calibration subtraction, as explained above. Thus, we deleted all fixations after the RT button press, and those right before that press for the duration of the RT button press time. In addition, all trials that were excluded from the analysis of the RT data were excluded from the analysis of the eye-tracking data also. We analysed the fixation data similarly to search times, with the number of fixations as a dependent gamma-distributed variable and with the independent and random terms as in the search-time model.

Typing performance: Finally, fifth-day typing performance was assessed in characters per second (CPS) [40]. A regression with the between-subjects factor of whether the participant was using a newly randomised or the earlier assigned layout was taken as the independent variable in the model, with CPS as the dependent variable. We used this procedure to answer the question of how great the role of knowing the key locations in a keyboard layout is for typing performance.

Parameter Fitting

For model fitting and estimation, the search times from the experimental data were first median aggregated within individual participants and task blocks, then mean aggregated within task blocks. The grand mean block values thus obtained were then used to fit the three LTM parameters for the model (F, f, and σ_M). For a fitness score, we calculated root-mean-square error ($RMSE$) between the predicted search times and RT data for each 100 random parameter combinations. Here, $RMSE$ represents the accuracy of the predictions when the model was fitted with the data in terms of how much, on average, the predictions deviate from the data Random search with researcher-provided seeds and Gaussian process modelling (with the package GPfit in R [39]) yielded very similar parameter values. The final values are reported in Table 1.

Results

Search times: All participants except one demonstrably learned the layout of their assigned keyboards over the four days (one participant grew worse each day but was not excluded from the data analysis). A multilevel regression with search time as the dependent variable and block as the independent variable showed a statistically significant main effect for block, $F(1,294) = 103.6$, with $p < .001$. The change in search times between blocks is displayed in Figure 3, which also shows the model's predictions with the best-fitting parameters ($F = 1.06, f = 1.53$, and $\sigma_M = 0.60$). The model's fit was $R^2 = 0.97, RMSE = 0.03$. For the variance in search times produced by the model, and predictions up to five hours of learning, see Figure 1a.

Visual search patterns: The number of fixations per trial decreased each day: $F(1,287) = 64.8$, with $p < .001$. The model's fit for the number of fixations when normalised for the visual distance between the cue and the target was $R^2 = 0.85$. For a non-normalised fixation count, $RMSE = 2.2$, indicating that the model underestimated the number of fixations. The reason for this may be that the model makes a new fixation strictly only when necessary, while human beings perform smaller saccades even when not necessarily needing one. Figure 4 presents sample fixation locations in trials on days 1, 2, and 4 for both a human participant and the simulator. We picked similar tasks for qualitative illustration of human and model-produced search strategies

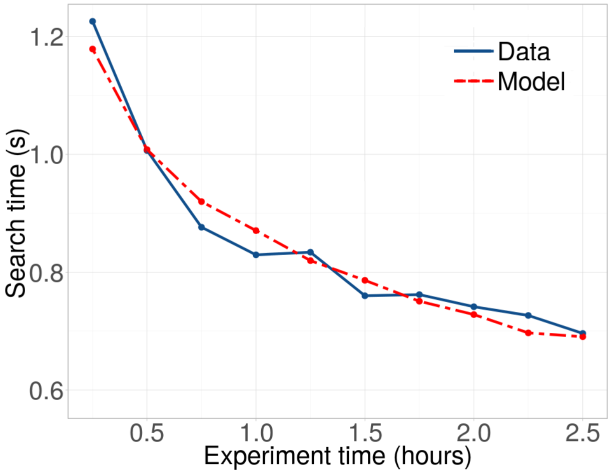

Figure 3. Development of search times in experiment 1 for the empirical data and model. The task blocks, each 15 minutes long, were distributed over four consecutive days. Model fit to the data was $R^2 = 0.97, RMSE = 0.03$.

Typing performance Although the model does not cover motor performance, it was still interesting to see how the participants performed when completing an actual typing task with the assigned layouts as compared to newly randomised ones. The mean CPS value for the group using the layout that was assigned to them on the first day was 0.58 (7 WPM), and that for the group using a random layout was 0.37 (4.4 WPM). The older subjects contributed somewhat to the low WPM figures [58]. Among elderly subjects, typing was clearly slower (0.34 CPS vs. 0.59 CPS). Also, the participants had not had a chance to develop motor performance with their assigned keyboards. That said, having positional knowledge of the layout has a large impact on typing performance; hence, models of positional learning are relevant for layout design.

Summary

The model's predictions of initial visual search times with new keyboard layouts, as well as of the subsequent learning curve, were a good fit with the data. The same was true of predicting decrease in the number of fixations per trial, although the model did underestimate the number of fixations per task. Furthermore, the simulated search patterns and their change due to learning were shown to be qualitatively similar to data from human subjects.

STUDY 2: LEARNING A CHANGED LAYOUT

The first study dealt with learning a completely new layout and did not involve the participants relearning with a partially different one. Data from such conditions are needed 1) for parameterising (obtaining the utility-learning parameters α and σ_U) and 2) for evaluating how well the model predicts learning and typing performance with changed layouts. To inform assessment of generalisation to a realistic context, study 2 involved typing instead of visual search alone. Here, we investigated the impact of partial layout changes (key swaps) on typing performance and the subsequent relearning.

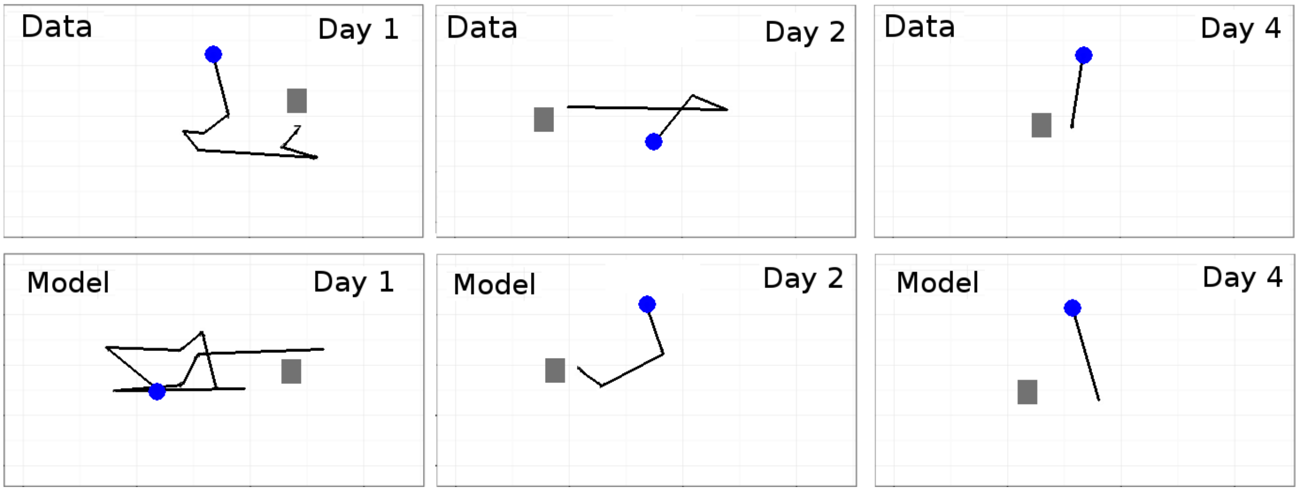

Figure 4. From study 1, example fixation patterns for three tasks, on days 1, 2 and 4, for a human participant (upper row) and the model (lower row). The blue circle is the cue location, where the task started, and the grey rectangle indicates the location of the target. The first-day search tasks resulted in multiple fixations around the task space, whereas on the last day, both participant and model fixated close to the target.

Method

Participants: We recruited 10 people to participate in the study (4 M, 6 F, ages 21 to 36 years, mean age 26, $SD = 4.4$). All were university students and experienced in smartphone use (3–4 hours/day) and text messaging in particular (1–2 hours/day) They rated themselves as being good or expert at touchscreen typing.

Procedure and Task: The experiment applied a within-subjects design wherein each participant completed transcription tasks with the dominant-hand index finger on two touchscreen keyboards, first a Qwerty layout, then a layout wherein the E and N and the A and O keys were swapped. These letters were chosen because they were the most frequent in the English-language corpus used: hence, the effect of the swap would be larger than with infrequently pressed keys.

Each of the two blocks consisted of 10 minutes of uninterrupted typing, with a five-minute rest between blocks. The model was likewise given cues for the four swapped keys more frequently than for others. After it was trained for 50 simulated hours, to have extensive knowledge of the Qwerty layout, the layout was changed, to the one the human participants had used (see Figure 5), and the model was given cues for keys for 10 simulated minutes. The cue frequencies corresponded to the letter frequencies in the corpus used in the experiment. For model fitting, both the full dataset and a subset with only one-letter words were used, the latter because this corresponds best to the task that the model handles: to find one key on the keyboard at a time. Further, fit analysis was conducted separately for full one-letter typing data and a subset with only the swapped keys.

Parameter Acquisition

Both human- and model-generated data were median aggregated into five two-minute blocks. From this point onward, the process of fitting α and σ_U was identical to the procedure in study 1. Parameter search resulted in the best α (0.02) and σ_U (0.25). The LTM parameters were as set in study 1.

Results

Typing performance: The participants' median typing performance with Qwerty was 0.37 seconds per key (32 WPM), and it was 0.44 seconds per key (27 WPM) with the key-swap layout. Median CPS values with the key-swap layout for the five blocks were 0.50 (24 WPM), 0.44, 0.41, and 0.40 (30 WPM). The model's performance after 50 hours of learning settled at 0.40 seconds per key (Figure 1 shows the model's simulation). It should be noted that while merely looking for a key is a task somewhat different from typing, the results are almost the same.

The model's fit to the full set of typing data was $R^2 = 0.49, RMSE = 0.15$, and for the one-letter typing data it was $R^2 = 0.76$, $RMSE = 0.24$. For one-letter typing data with only the swapped keys, the fit is $R^2 = 0.60$, $RMSE = 0.19$. Since visual search tasks and typing differ in nature, the fit obtained speaks to the model's value for analysing the effects of smaller changes in layout on typing performance.

Summary

The fit with the data suggests that the model can accurately capture changes in visual search times after a small change in

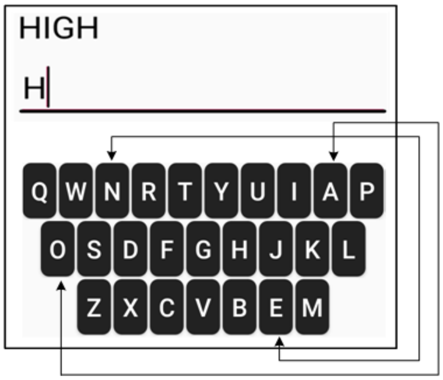

Figure 5. Study 2, the transcription task and keyboard with swapped keys marked with arrows.

layout. Further, the results show that a large proportion of typing performance may be attributable to visual search when the user is not an expert in the layout. This is because the model's predictions were plausible even though the experimental data included motor action whilst the model's operations considered only visual search.

SIMULATING LAYOUT CHANGES

With the parameters fitted, the model can be used to evaluate the learning and relearning of various layout designs for touchscreen keyboards. The last section of the paper explores the practical value of the model's ability to anticipate how new designs affect immediate performance and relearning.

Case 1: Testing Generalisability with Dunlop et al. 2012

We conducted a further test of the model's ability to predict the impact of partial layout changes on performance and the subsequent relearning, by using the data presented by Dunlop et al. [21]. The layout they used, *Sath*, was optimised to balance faster-than-Qwerty typing with familiarity to Qwerty users. Hence, it offers a suitable test for the model's fit. The impact of a keyboard switch from Qwerty to Sath was reported as percentage speed relative to Qwerty [21], so the results with our model were converted to this form. Although details of how long the participants ($N = 10$) typed with the Sath layout (on a regular smartphone) are not available, some estimates can be made. There were four sessions, each of less than 45 minutes. In the first session, only half of the time was used in typing with Sath. Here, we assume a session to have included 15 minutes of typing, which puts the typing time in the first session at 15 minutes of typing with Sath, and that in each of the other three sessions at 30 minutes.

The model was again trained to be expert in Qwerty via a time span of 50 simulated hours. The typing speed for the last 60 minutes was recorded for comparison with the performance after the switch to Sath. The reported human typing performance with Sath, as a fraction of Qwerty performance, was 0.64, 0.65, 0.72, and 0.85 [21]. The model's predictions with all parameters set as above and not fitted for these data specifically were 0.43, 0.48, 0.52, and 0.55. The fit was $r^2 = 0.80, RMSE = 0.23$, a result that supports the model's validity. The impact of the switch was more dramatic with the model, but one factor in this is probably that, whereas the model was expert in visual search on Qwerty (30 WPM), the human participants were not as fast at Qwerty typing (21.3 WPM) [21]. Nevertheless, the slope of the predicted relearning times is comparable to the human data without adjustment of the model's parameters.

Case 2: Comparison of Three Designs

In the second case, our interest lay in a scenario wherein a designer wants to learn about the effects of changing a familiar layout. Obviously, the model should predict that the closer the new layout is to the familiar one, the easier the transition will be. It should also predict the length of the transition period.

Compared layouts: To this end, we examined three keyboard layouts from HCI literature, each suggested to be faster to type with than Qwerty for an experienced user: *Dvorak*, *Sath*, and *IJQwerty* (see Figure 6). Dvorak (in pane a) has little similarity with the Qwerty layout. Sath, as a layout optimised to be familiar to Qwerty users but at the same time yield higher typing speeds [21], served as a middle level between hard and easy layout reconfiguration (see pane b). Finally, IJQwerty represents a very small layout change, consisting of swapping Qwerty's I and J (see pane c) [11]. The small and local swap in IJQwerty should have a smaller impact on relearning costs.

Dependent Variables: We analysed two values of interest from the model's simulations: 1) the immediate impact of a keyboard switch on visual search time and 2) how long it takes to achieve an acceptable level of performance. Here, 'acceptable' was deemed to be 0.6 s search time per key, which corresponds to 20 WPM (with five-character words).

Results: For all three cases, the model was firstly allowed to reach expert level with Qwerty by conducting visual searches for 50 simulated hours. We set all of the model's parameters in line with the two empirical studies. Figure 6 depicts the impact of layout change on search times and also shows how search times start to decrease as relearning with the new layout occurs. As predicted, the negative impact of the new layout on visual search times is the biggest with Dvorak, then with Sath, and only slightly with IJQwerty. Likewise, relearning with the Dvorak layout takes a long time (6.0 hours), whereas search times for the Sath (4.2 h) and especially for IJQwerty recover more quickly (for IJQwerty, the switch did not raise search times above 0.6 seconds). Figure 6 also shows that the model acquires similar search times and projected learning curve with both Dvorak and Sath with about 15 hours of practice.

Case 3: Changes in Keyboard Form Factor

In the second and final case, we examine the effect of switching from a full keyboard to one with fewer keys. While the parameter fitting was conducted for a keyboard with 26 keys and normal dimensions (three rows of $7-10$ keys), it is possible to simulate learning with other row–column designs. The model can predict learning times to the required level of expertise.

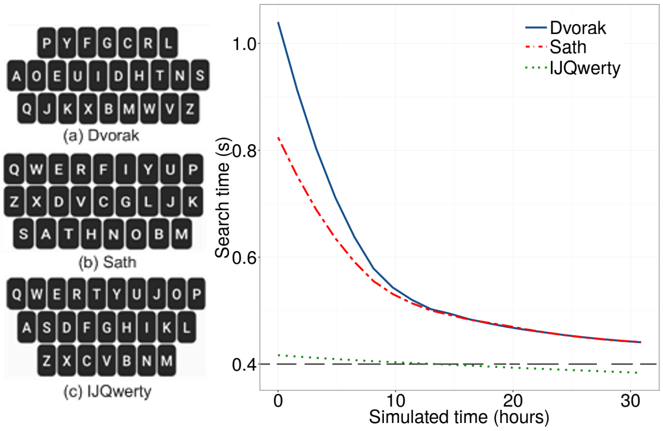

Figure 6. Left: Three investigated keyboard layouts. Right: The simulated negative impact on visual search times of a layout change from Qwerty to one of these three example layouts (the dashed horizontal line shows performance before the switch). The model was expert in Qwerty visual search and was then forced to switch to Dvorak (a), Sath (b), or IJQwerty (c).

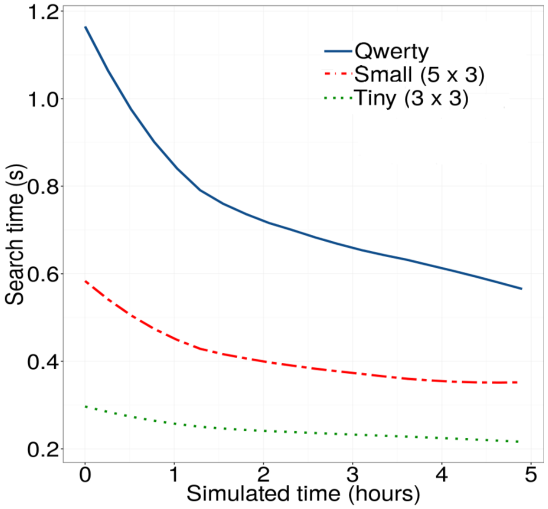

Figure 7. The model's simulation of the learning curve for a tiny (9-key), small (15-key), and Qwerty (26-key) layout.

Using the parameters and training reported above, the model predicts that learning to visually search keys consistently in less than 0.6 s, which would correspond to WPM > 20, takes only 0.02 hours for a layout consisting of nine keys arranged in a 3 × 3 grid (see Figure 7). In a layout with three rows having five keys each, the search time is still only 0.08 hours. In study 1, with 26 keys the participants did not achieve search times this low (see Figure 3), and the model predicts that it would take an average participant 4.5 hours to reach this level.

DISCUSSION

The present work advances understanding of how users learn to locate keys on a keyboard. The proposed simulation-based model predicts how quickly they learn a layout, how their search times and patterns change over time, the way in which layout changes set them back, and how factors such as layout design and the number of keys affect all of this. Two experiments were used to fit the model's parameters and confirm that the model-simulated search times both correspond to human search times and can be used to predict the impact of layout change on performance in typing with touchscreen keyboards.

The model assumes that two memory systems 'compete' for control of gaze. The short-term component is necessary to ensure that search times are not too long in the beginning, the long-term component to address learning and forgetting of positions. The modelling assumptions and most parameters are rooted in previous literature on cognitive models. Various types of evidence supporting the model have been presented above. The model shows a good fit with experimental data pertaining to search times and patterns in how users learn new layouts. Via associated parameters, the model was further fitted to predict experimental data related to performance changes when the user is relearning, typing with a slightly changed layout. Without changes to the parameters obtained in these two studies, the model's predictions proved an acceptable fit to data from an external study on the impact of a partially new layout on performance and relearning time.

There are many possibilities for improving on the model presented here, with the long-term goal being to offer a generic model of learning different layouts, not only keyboards. The model should be extendable to account for objects' shape and colour while also adjusting to different work flows [35] – that is, not just searching for randomly cued keys but adapting to changing task conditions. However, such extensions might better find applications outside ordinary keyboards, which are two-dimensional grids composed of elements with very small variations in size, colour, orientation, and shape.

While the model does not predict motor performance, it offers an estimated lower bound to typing performance, which is particularly pronounced in the beginning during learning of a new layout. Nevertheless, an obvious step is to integrate the visual search model with a finger-movement model, utilising, for example, Fitt's law. In this modelling, a critical factor will lie in analysing how much the eye and the finger move in parallel when the user types on a mobile phone. Further components for this integrated model of touchscreen typing should include, for instance, dividing attention between the keyboard and the text area, proofreading the text typed, and using multiple fingers to type. These extensions forecast an integrated model, which considers how learning and performance are affected by various UI elements. Such a model is crucial for understanding individual differences in skill with various daily computer tasks. As no user is the same, this is an important step in creating accessible and usable UIs.

CONCLUSION

Today's user interacts constantly with multiple and changing interfaces, and consistency is thus a critical challenge for UI design. The probability of accepting a new interface is associated with how quickly familiarity can be achieved. As evident in the case of keyboard layouts, new, more efficient designs are rarely accepted due to the difficulty of relearning. Our model clarifies this problem by operationalising layout learning and relearning as a dynamic system consisting of visual search, visual short-term memory, long-term memory, and utility learning. The model can assist designers and decision-makers by predicting **1)** how changes in layouts influence relearning times; **2)** that large changes in layouts are possible when some maximum acceptable relearning cost is assumed; and **3)** how to change one layout into another subtly.

Formal models of user, such as presented in this paper, are a necessary instrument in design, especially when the design space is large. Whenever the user faces a new design, visual search will be a limiting factor to its use until the positions of the elements have been learned. The model presented here provides a psychologically grounded and empirically tested tool for analysing this phenomenon, and helps the designers attain sufficient UI consistency when implementing new designs.

ACKNOWLEDGMENTS

This work has received funding from the joint JST–AoF project "User Interface Design for the Ageing Population" (AoF grant 291556) as an activity of FY2014 Strategic International Collaborative Research Program (SICORP), and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 637991). Dongcai Wen supported designing the text entry interface used in the experiments.

REFERENCES

1. John R Anderson. 1976. Language, memory and thought Hillsdale. *NJ: LEA* (1976).
2. John R Anderson. 1995. *Learning and memory*. Wiley, New York.
3. John R Anderson. 2007. *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press, New York.
4. John R Anderson. 2013. *The architecture of cognition*. Psychology Press.
5. John R Anderson, D Bothell, C Lebiere, and M Matessa. 1998. An integrated theory of list memory. *Journal Of Memory And Language* 38, 4 (1998), 341–380. DOI: http://dx.doi.org/10.1006/jmla.1997.2553
6. Alan Baddeley. 2012. Working memory: theories, models, and controversies. *Annual review of psychology* 63 (2012), 1–29.
7. Gilles Bailly, Antti Oulasvirta, Duncan P Brumby, and Andrew Howes. 2014. Model of visual search and selection time in linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3865–3874. DOI: http://dx.doi.org/10.1145/2556288.2557093
8. Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. Menuoptimizer: Interactive optimization of menu systems. In *Proceedings of the ACM symposium on User interface software and technology*. ACM, 331–342. DOI: http://dx.doi.org/10.1145/2501988.2502024
9. Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software* 67, 1 (2015), 1–48. DOI:http://dx.doi.org/10.18637/jss.v067.i01
10. Xiaojun Bi, Barton A. Smith, and Shumin Zhai. 2010. Quasi-qwerty Soft Keyboard Optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 283–286. DOI: http://dx.doi.org/1753326.1753367
11. Xiaojun Bi and Shumin Zhai. 2016. IJQwerty: What Difference Does One Key Change Make? Gesture Typing Keyboard Optimization Bounded by One Key Position Change from Qwerty. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 49–58. DOI: http://dx.doi.org/10.1145/2858036.2858421
12. Michael D Byrne. 2001. ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies* 55, 1 (2001), 41–84.
13. Stuart K Card, Thomas P Moran, and Allen Newell. 1980. Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive psychology* 12, 1 (1980), 32–74.
14. Stuart K Card, Allen Newell, and Thomas P Moran. 1983. The psychology of human-computer interaction. (1983).
15. Andy Cockburn, Carl Gutwin, and Saul Greenberg. 2007. A predictive model of menu performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 627–636. DOI: http://dx.doi.org/10.1145/1240624.1240723
16. Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2015. Supporting novice to expert transitions in user interfaces. *ACM Computing Surveys (CSUR)* 47, 2 (2015), 31:1–31:36.
17. Nelson Cowan. 2012. *Working memory capacity*. Psychology press.
18. Justin Cuaresma and I Scott MacKenzie. 2013. A study of variations of Qwerty soft keyboards for mobile phones. In *Proceedings of the International Conference on Multimedia and Human-Computer Interaction-MHCI*. 126.1–126.8.
19. Arindam Das and Wolfgang Stuerzlinger. 2012. Comparing Cognitive Effort in Spatial Learning of Text Entry Keyboards and ShapeWriters. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, New York, NY, USA, 649–652. DOI: http://dx.doi.org/10.1145/2254556.2254676
20. Paul A David. 1985. Clio and the Economics of QWERTY. *The American economic review* 75, 2 (1985), 332–337.
21. Mark Dunlop and John Levine. 2012. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2669–2678. DOI: http://dx.doi.org/10.1145/2207676.2208659
22. August Dvorak. 1943. There is a better typewriter keyboard. *National Business Education Quarterly* 11, 51-58 (1943), 66.
23. Brian D Ehret. 2002. Learning where to look: Location learning in graphical user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 211–218. DOI: http://dx.doi.org/10.1145/503376.503414
24. Janine F. Hay and Larry L. Jacoby. 1999. Separating habit and recollection in young and older adults: effects of elaborative processing and distinctiveness. *Psychology and aging* 14, 1 (1999), 122–134. DOI: http://dx.doi.org/10.1037/0882-7974.14.1.122
25. William E Hick. 1952. On the rate of gain of information. *Quarterly Journal of Experimental Psychology* 4, 1 (1952), 11–26.
26. Joop Hox. 2010. *Multilevel analysis: Techniques and Applications* (2nd ed.). Routledge, Hove.
27. Johan Hulleman. 2009. No need for inhibitory tagging of locations in visual search. *Psychonomic Bulletin & Review* 16, 1 (2009), 116–120.

28. Ray Hyman. 1953. Stimulus information as a determinant of reaction time. *Journal of experimental psychology* 45, 3 (1953), 188–196.

29. Christian P. Janssen and Wayne D. Gray. 2012. When, What, and How Much to Reward in Reinforcement Learning-Based Models of Cognition. *Cognitive Science* 36, 2 (2012), 333–358. DOI: http://dx.doi.org/10.1111/j.1551-6709.2011.01222.x

30. 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, 455–462. DOI: http://dx.doi.org/10.1145/985692.985750

31. Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: from eye fixations to comprehension. *Psychological review* 87, 4 (1980), 329–354.

32. S. W. Keele. 1986. Motor Control. In *Handbook of perception and human performance*, & J. P. Thomas K. R. Boff, L. Kaufman (Ed.). Wiley, New York, 30.1–30.60.

33. Nina Keith and K Anders Ericsson. 2007. A deliberate practice account of typing proficiency in everyday typists. *Journal of Experimental Psychology: Applied* 13, 3 (2007), 135–145.

34. David Kieras. 2011. The persistent visual store as the locus of fixation memory in visual search tasks. *Cognitive Systems Research* 12, 2 (2011), 102–112.

35. David E Kieras and Anthony J Hornof. 2014. Towards accurate and practical predictive models of active-vision-based visual search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3875–3884. DOI: http://dx.doi.org/10.1145/2556288.2557324

36. David E Kieras and David E Meyer. 1997. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-computer interaction* 12, 4 (1997), 391–438.

37. Tuomo Kujala and Dario D Salvucci. 2015. Modeling visual sampling on in-car displays: The challenge of predicting safety-critical lapses of control. *International Journal of Human-Computer Studies* 79 (2015), 66–78.

38. Paul Ung-Joon Lee and Shumin Zhai. 2004. Top-down learning strategies: can they facilitate stylus keyboard learning? *International journal of human-computer studies* 60, 5 (2004), 585–598.

39. Blake MacDonald, Pritam Ranjan, Hugh Chipman, and others. 2015. GPfit: An R package for fitting a gaussian process model to deterministic simulator outputs. *Journal of Statistical Software* 64, 12 (2015), 1–23.

40. I. Scott Mackenzie. 2002. A note on calculating text entry speed. (2002). Unpublished work. Available online at http://www.yorku.ca/mack/RN-TextEntrySpeed.html.

41. I. Scott MacKenzie and Shawn X. Zhang. 1999. The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 25–31. DOI: http://dx.doi.org/302979.302983

42. I. Scott MacKenzie and Shawn X Zhang. 2001. An empirical investigation of the novice experience with soft keyboards. *Behaviour & Information Technology* 20, 6 (2001), 411–418.

43. Laurent Magnien, Jean Leon Bouraoui, and Nadine Vigouroux. 2004. Mobile Devices: Soft Keyboard Text-entry Enhanced by Visual Cues. In *Proceedings of the 1st French-speaking Conference on Mobility and Ubiquity Computing*. ACM, New York, NY, USA, 158–165. DOI: http://dx.doi.org/10.1145/1050873.1050908

44. Akira Miyake and Priti Shah. 1999. *Models of working memory: Mechanisms of active maintenance and executive control*. Cambridge University Press.

45. Allen Newell. 1990. *Unified theories of cognition*. Harvard University Press, Cambridge, Mass.

46. Allen Newell and Paul S Rosenbloom. 1981. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition* 1 (1981), 1–55.

47. Marketta Niemelä and Pertti Saariluoma. 2003. Layout attributes and recall. *Behaviour & information technology* 22, 5 (2003), 353–363.

48. Antti Oulasvirta, Lari Kärkkäinen, and Jari Laarni. 2005. Expectations and memory in link search. *Computers in Human Behavior* 21, 5 (2005), 773–789.

49. Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, and Myroslav Bachynskyi. 2013. Improving Two - Thumb Text Entry on Touchscreen Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), 2765–2774. http://10.1145/2470654.2481383

50. Andriy Pavlovych and Wolfgang Stuerzlinger. 2004. Model for non-expert text entry speed on 12-button phone keypads. In *Proceedings of the Conference on Human Factors in Computing Systems*. 351–358. DOI: http://dx.doi.org/985692.985737

51. David Peebles and Corinna Jones. A model of object location memory. In *Proceedings of the Annual Conference of the Cognitive Science Society*.

52. Robert A Rescorla and Allan R Wagner. 1972. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory* 2 (1972), 64–99.

53. Ladislao Salmerón, José J Cañas, and Inmaculada Fajardo. 2005. Are expert users always better searchers? Interaction of expertise and semantic grouping in hypertext search tasks. *Behaviour & information technology* 24, 6 (2005), 471–475.

54. Timothy A Salthouse. 1984. Effects of age and skill in typing. *Journal of Experimental Psychology: General* 113, 3 (1984), 345–371.

55. Dario D Salvucci. 2001. An integrated model of eye movements and visual encoding. *Cognitive Systems Research* 1, 4 (2001), 201–220. DOI: http://dx.doi.org/10.1016/S1389-0417(00)00015-2

56. Dario D Salvucci and Kristen L Macuga. 2002. Predicting the Effects of Cell-Phone Dialing on Driver Performance. *Cognitive Systems Research* 3, 1 (2002), 95–102. DOI: http://dx.doi.org/10.1016/S1389-0417(01)00048-1

57. Andrew Sears, Julie A Jacko, Josey Chu, and Francisco Moro. 2001. The role of visual search in the design of effective soft keyboards. *Behaviour & Information Technology* 20, 3 (2001), 159–166.

58. Amanda L. Smith and B. S. Chaparro. 2015. Smartphone Text Input Method Performance, Usability, and Preference With Younger and Older Adults. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 57, 6 (2015), 1015–1028. DOI: http://dx.doi.org/10.1177/0018720815575644

59. Leong-hwee Teo, Bonnie E John, and Marilyn Hughes Blackmon. 2012. CogTool-Explorer: A Model of Goal-Directed User Exploration that Considers Information Layout. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), 2479–2488. DOI: http://dx.doi.org/10.1145/2207676.2208414

60. Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the ACM Conference on Designing Interactive Systems*. ACM, 543–555. DOI: http://dx.doi.org/2901790.2901817

61. Vladislav D. Veksler, Christopher W. Myers, and Kevin A. Gluck. 2014. SAwSu: An Integrated Model of Associative and Reinforcement Learning. *Cognitive Science* 38, 3 (2014), 580–598. DOI: http://dx.doi.org/10.1111/cogs.12103

62. Viswanath Venkatesh, James YL Thong, and Xin Xu. 2016. Unified Theory of Acceptance and Use of Technology: A Synthesis and the Road Ahead. *Journal of the Association for Information Systems* 17, 5 (2016), 328.

63. R William Soukoreff and I. Scott Scott Mackenzie. 1995. Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard. *Behaviour & Information Technology* 14, 6 (1995), 370–379.

64. Eldad Yechiam, Ido Erev, Vered Yehene, and Daniel Gopher. 2003. Melioration and the transition from touch-typing training to everyday use. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 45, 4 (2003), 671–684.

65. Shumin Zhai, Michael Hunter, and Barton A Smith. 2002a. Performance optimization of virtual keyboards. *Human–Computer Interaction* 17, 2-3 (2002), 229–269.

66. Shumin Zhai, Alison Sue, and Johnny Accot. 2002b. Movement model, hits distribution and learning in virtual keyboarding. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2002), 17–24. DOI:http://dx.doi.org/503380.503381