A Revised Mobile KLM for Interaction with Multiple NFC-Tags

Paul Holleis[1], Maximilian Scherr[2], Gregor Broll[1]

[1] DOCOMO Euro-Labs, Landsberger Str. 312, 80687 Munich, Germany
{holleis, broll}@docomolab-euro.com
[2] Ludwig-Maximilians-Universität München, Amalienstr. 17, 80333 Munich, Germany
scherr@cip.ifi.lmu.de

Abstract. The Keystroke-Level Model (KLM) is a model for predicting the execution time of routine tasks. Initially, it had been devised for standard keyboard-desktop settings but an extension of this model for interactions with mobile phones has been described by Holleis et al. [10]. We propose a considerable update of this KLM focusing on NFC-based applications and interactions which are continuously gaining interest. Insufficiencies within the previous model regarding operators for Pointing, Mental Acts, and System Response Time are treated. We present the results of several studies conducted in order to update the values of these operators. A specific focus is put on the differences between static (NFC tags behind a printed poster or object) and dynamic interfaces (tagged displays or projections). Finally, we validate our results by modeling two applications with the former and the proposed model. The latter performed consistently better when compared with measurements from real user interaction data.

Keywords: Keystroke-level model (KLM), mobile device interaction, Near Field Communication (NFC), static / dynamic multi-tag interaction

1 Introduction

The Keystroke-Level Model (KLM) [5] is an established tool for assessing the performance of application designs in early phases of development before prototyping or implementation. It can predict the average execution time for specific tasks by modeling how much time expert users need to accomplish them on the level of basic input events, e.g. moving the mouse or pressing a key. For that purpose, KLM defines a set of *operators*, e.g. pointing, keystrokes or mental preparation, as well as average execution times for them that have been derived from experiments. The added execution times for KLM operators of a task can then be used to assess or compare different application designs for the same task.

Originally, KLM has been developed to assess the performance of desktop applications that get input from a keyboard and a mouse. With the advancement of Mobile Computing, it has also been applied and adapted to mobile devices and applications. Mobile devices enable new kinds of interactions that use Bluetooth, cameras, GPS, Near Field Communication (NFC) [25] or other technologies to interact with the real world and its objects. In order to incorporate these and other aspects of mobile interaction, Holleis et al. [10] have introduced the Mobile KLM with updated operators and execution times of the original KLM, e.g. Keystrokes or Pointing. It also includes new operators to model attention shifts, distraction or interactions with objects that are tagged with NFC or visual markers [21].

In this paper, we revisit the Mobile KLM in order to bring it up to date with current technology and knowledge for mobile interactions with NFC-tags. NFC is a wireless technology for data exchange over short distances. Passive tags can be attached to almost arbitrary objects and can store digital data which can be read by touching them with NFC-enabled mobile devices. In the Mobile KLM, Holleis et al. [10] (see the detailed description below) only regard simple interactions with *single tags*. Although this is very convenient for mobile payment, ticketing or information retrieval, mobile interaction with NFC can be more elaborate. Everyday objects such as posters or advertising columns can comprise *multiple NFC-tags* and serve as physical user interfaces (UI) that adopt features of mobile applications, complement mobile UIs and support new interaction techniques. Additionally, we focus on novel types of applications that use *dynamic interfaces*. In contrast to *static interfaces* such as tagged objects or posters, dynamic applications use tagged displays or tagged screens on which an interface is projected.

In the following section, we summarize related work about (mobile) KLM and mobile interaction with NFC, focusing on multi-tag interactions. Based on this overview, we point out shortcomings of the Mobile KLM regarding system response time, pointing and mental preparation for mobile interaction with multiple NFC-tags. Next, we describe a comprehensive user study whose different parts investigate these shortcomings one by one. The results are combined into an update of the Mobile KLM for multi-tag interaction with NFC. Finally, we validate our findings in an additional study that compares estimated task execution times for multi-tag applications derived from the original Mobile KLM and its suggested update. The latter performed consistently better on all instances compared to measurements from real user interaction data.

2 Related Work

The Keystroke-Level Model [5] can be seen as a lightweight version of the GOMS (Goals, Operators, Methods and Selection rules) model that has been developed by Card et al. [6] to model error-free interaction with interfaces. KLM simplifies the original GOMS by providing a common set of operators and modeling only one method thus removing the need for selection rules. The resulting KLM can be used to predict the time it would take expert users to complete a task with the given interface. The basic KLM operators defined in [5] are *Pointing* (e.g. with a mouse or other pointing devices), *Keystrokes* (e.g. pressing keys or buttons), *Homing* (moving the hand between the mouse and the keyboard), *System Response Time* and *Mental Act* (i.e. mental preparation for a subtask). The KLM also provides average execution times for its operators that have been derived from several experiments. Those have been applied and adapted for different kinds of applications and interactions in order to provide reliable performance assessments for them.

Over the last 25 years, the KLM has proven to be effective in a great variety of application areas such as email organization [1], map digitization [9], navigation systems [16], or interfaces for people with disabilities [14]. Mobile devices, applications and interaction have also become an area of application for KLM taking into account the specific properties and constraints of this platform. Small screens and keyboards, context-awareness, and mobility make changes and additions in the existing KLM essential. One example can be found in Luo and Siewiorek [15] who extend it to also predict the energy consumption of interactions on mobile devices. Still, most of the early works on models for mobile devices and applications only assess simple interactions like navigating mobile phone menus [17]. Still, the model and its operators have already been extensively used and validated, see, e.g., Holleis [11] for an overview on existing KLMs for text entry systems on mobile devices.

In order to be able to describe and examine more advanced interactions on mobile phones, Holleis et al. [10] performed an extensive set of studies and provide an updated Mobile KLM. It revisits operators of the original KLM and adds new operators modeling novel types of interaction such as gestures or using tagged objects. The operators set in bold are of particular interest for the paper at hand. NFC interaction has been considered with the operators R (using 2.58s as response time for NFC) and A (used to model steps such as focusing the camera; it is set to zero for NFC interactions).

Table 1. Overview on KLM operators from Holleis et al. [10]. The ones set in bold or their uses are updated within this paper.

Operator	Function	Value	New?
A, Action	Certain interactions require additional actions	0.00s-1.23s	●
F, Finger	Finger movement on a small screen	0.23s	●
G, Gestures	Gesture interaction with finger, hand, etc.	0.80s	●
H, Homing	Moving phone to/from ears	0.95s	
I, Initial Act	Find and start interaction with device	1.18s-5.32s	●
K, Keystroke	Press a key / hotkey on the device	0.39s/0.16s	
M, Mental Act	Mental preparation for a subtask	1.35s	
P, Pointing	Moving the mobile device	1.00s	
R, System Response	Waiting for the system to respond	variable	
S_{Macro}	Attention shift world↔device	0.36s	●
S_{Micro}	Attention shift keypad↔display on device	0.14s	●
X, Distraction	Distraction by environment	6-21% slowdown	●

Since 2007, when the Mobile KLM was introduced, mobile interaction with tagged objects has considerably evolved. In particular, Near Field Communication (NFC) [25] has become a popular technology for mobile payment, ticketing, information retrieval or service interaction [20]. Apart from these simple interactions with single tags, NFC can also be used for more elaborate interactions with multi-tagged objects that have not been considered in the Mobile KLM. Applications can map features and options to multiple NFC-tags on physical objects that serve as physical UIs and thus complement mobile UIs. Examples are posters for mobile ticketing [2], tagged maps [19] or control panels for multimedia players [22].

In the next step, physical UIs are covered with a grid of NFC-tags to provide a completely interactive surface for the manipulation of dynamic application UIs on displays or projected interfaces. Opposite to static posters, these interactive surfaces are more flexible and can dynamically map their tags to the UI elements of different applications. Vetter et al. [24] and Hardy et al. [8] have created and refined the first dynamic NFC-displays that use a grid of NFC-tags for the interaction with an application UI that is projected onto this interactive surface. Similarly, Ramírez-González et al. [18] have combined a grid of NFC-tags and a projected application UI to build an interactive NFC-panel. Seewoonauth et al. [23] use a grid of tags on the back of a laptop display to enable direct, touch-based interactions.

3 Shortcomings of the Current Mobile KLM

The Mobile KLM of Holleis et al. [10] has proven valuable for the analysis of several novel mobile interactions. However, there are several shortcomings that hinder the exact modeling of NFC-based interaction, especially with multiple tags. From our experience with modeling and studying multi-tag interfaces, we identified several aspects that need to be improved:

1. System response time reading NFC tags with current mobile devices
2. Model errors caused by mismatches between tag-matrix and UI widgets
3. Pointing time for close and far objects
4. Chunking of action sequences needs to be re-evaluated

Item (1) is a common issue of models such as KLM. Although the general type and technique of interaction may remain the same across device models, hardware and system response time are bound to change. The time to complete a task can significantly change (in most cases decrease) when upgrading to a newer type of device or hardware. For NFC-based applications, the time for reading the content of a tag has a significant impact on the total task completion time and needs to be updated from the previous Mobile KLM.

Issues (2) to (4) arise from the fact that mobile interaction with NFC has changed compared to the applications envisioned in the previous Mobile KLM. The switch from single-tag to multi-tag interactions means that additional unit tasks appear and that the properties of some types of interaction changed. Item (2) has not been encountered before as there was always one UI element matched onto one tag. Some multi-tag applications use a grid of tags that covers the whole UI area. This allows for a more flexible use of the tagged area as UI widgets can be displayed everywhere and be rearranged dynamically. This also means that the direct mapping between tags and UI items is lost. Thus, a displayed button can span two or more tags and the user might touch the gap between adjacent tags, failing to read a tag and thus to press the button.

Item (3) dwells on the fact that the previous Mobile KLM suggests using an average of about one second to model any pointing operation. As proposed in the original KLM by Card et al. [5], this should be replaced by a dynamic model such as Fitts' Law [7] to make the model more precise. However, in practice this is often too cumbersome if it cannot be automated. Thus, constant approximations are often used. In the case of multi-tag interaction, there are often many occurrences of a pointing action and smaller errors can add up more quickly than in single-tag interactions.

Item (4) refers to the placement of a special operator within the KLM: the Mental Act operator M is used to model some mental preparation that a user needs to perform before engaging into a physical action. However, certain sequences of actions can be anticipated and are not interrupted by a mental act. Guidelines for finding such sequences have been given by Kieras [13] and Holleis et al. [10]. These guidelines need to be updated for multi-tag interactions that have not yet been considered.

4 Studies to Update the Mobile KLM

In order to update the Mobile KLM for multi-tag interaction with NFC, we have conducted a comprehensive user study whose different parts focus on the previously elicited shortcomings.

1. Familiarization: At the beginning of the study, the subjects were introduced to mobile interaction with NFC. In order to become more familiar with this new technology, we let them play a game that uses a dynamic setup with a grid of NFC-tags and a projector that displays the game area on top of it.

2. Pointing: Moving the mobile device close to a tag is one of the most frequent actions for multi-tag interfaces. In the Mobile KLM, this is modeled with the Pointing operator P. This study mimics a Fitts' Law study by comparing pointing between targets of varying sizes and distances, both on static and dynamic UIs.

3. Sequences: The effect of consecutive touching of different tags is the subject of the second study. More precisely, the placement of Mental Act operators – describing brief pauses in the interaction when users need to remember or look for something etc. – is examined.

4. Measurement: The system response time for the most current NFC-enabled Nokia phone (6212) is measured and the respective value from the Mobile KLM is updated.

4.1 Study Setup and User Familiarization

An important aspect of working with models such as KLM is that only expert, routine behavior can be modeled with high accuracy. Although some extensions can also treat learning effects, see e.g. [12], investigators should only observe interactions where users are sufficiently familiar with the respective application. Errors should be kept to a minimum and the principle sequence of actions should be clear to the subjects without needing to check instructions, ask, or spend significant time thinking about a solutions strategy (unless, of course this should be part of the study or model).

In total, 16 subjects (13 male) with an average age of 30.3 years took part in the study. Three subjects did not have a background in computer science or related technical fields. On a Likert-scale from 1 ("none") to 5 ("expert"), the subjects rated their overall technical expertise and their expertise with mobile device as rather high (3.94 and 3.81). Although 9 of the 16 subjects had heard of NFC before the study, only 3 of them had actually used it.

In order to bring all subjects to roughly the same ‘expert’ level for the study, we explained its overall procedure and different parts. At the beginning of each part, we explained which aspect of mobile interaction with NFC it tested and let the subjects gain hands-on experience with the used NFC-enabled mobile phone and multi-tag UI. The subjects also performed each particular task of the study various times until they felt comfortable and understood the goal of the task and the method to reach it.

Before we started the actual studies, we had them play a prototype of the popular Whack-a-Mole game that we had previously adopted for dynamic NFC-displays [3]. This prototype projects the gaming area onto a grid of NFC-tags. Players have to touch its tags with an NFC-enabled mobile phone in order to hit the moles that appear on the gaming area and to win credits. The technical setup comprises a wall mounted grid of 48 by 20 Mifare NFC-tags, and a laptop running a Java SE game server that projects the game interface onto the tag grid via a ceiling-mounted short-throw projector. Players use a J2ME client application running on NFC-enabled Nokia 6212 phones. Each subject played two times alone for preparation and one multi-player round with one of the investigators. At this point, we conjecture that all subjects were familiar with the general concept of NFC tags, interaction techniques using a mobile phone and such tags, as well as the technology's shortcomings: reading errors due to gaps and overlapping tags in the tag mesh.

4.2 NFC System Response Time

The Mobile KLM [10] has already incorporated time estimates for detecting and reading an NFC-tag with an NFC-enabled mobile phone. However, this measurement has been done some years ago and only with the NFC-enabled phones that were available at that time – the Nokia 3220 with an additional NFC-shell.

It is nearly impossible to programmatically measure the time for reading an NFC-tag. First, it is difficult to detect the start of the NFC connection with the existing APIs. Second, from an interaction point of view, it is less interesting to identify the time elapsed between powering the tag and finishing the extraction of information. Rather, the time lost because of that tag reading process is of interest. The difference is that part of the reading can already be done while the user is still in the process of pointing. Therefore, in order to measure the real system response time, we designed the following experiment: First, using the Nokia 6212, a series of 50 consecutive readings were performed as fast as possible. The user was notified at each successful read with a beep and a short vibration. Second, the same person performed the same actions the same number of times but did not touch a tag and did not wait for any response. This means, that the difference of both runs leaves only the NFC reading time. Dividing this difference by the number of repetitions results in an acceptable value for system response time of 0.40 seconds for a static interface, e.g. a poster.

In addition to the direct response time between the touching of a tag and the phone’s response, we repeated the experiment using a dynamic setup. This means that the notification needed to be sent via Bluetooth from the phone to the server and processed there. This resulted in a value for R_{NFC} for dynamic interfaces and the Nokia 6212 phone of 0.62 seconds. This value was also successfully validated in the video recordings from the other user studies reported later in this paper. For the previous Nokia 6131 phone, we measured R_{NFC}(Nokia 6131) to be 1.20s and for the only recently released Google Nexus S phone R_{NFC}(Nexus S, static) = 0.34s, R_{NFC}(Nexus S, dynamic) = 0.56 seconds.

4.3 Pointing / Fitts' Law Test

After the introduction to NFC and multi-tag UIs, we conducted the first part of the study to investigate pointing and selection speed. The basic setup mimicked similar studies that are used to derive the relationship between selection time and distance as well as size of the target, known as Fitts' Law [7]. The basic idea of this part of the study is to have users consecutively point to and touch specific areas. By varying the size and distance of these areas, the effect of those parameters can be measured, see [7]. We implemented both a static and a dynamic version.

Static Fitts' Law Study and Results

The static version is a direct copy of the original Fitts' Law experiment and uses three different sizes, respectively widths, of targets (1, 2 and 4 tags) and distances between them (30, 60 and 90 cm). Fig. 1 (left) shows some example configurations. In order to quickly change from one configuration to another, we created modular tiles (see Fig. 1, right) that can be easily attached and moved on a wall or whiteboard. The order of use was counterbalanced with a Latin Square design. During the study, an NFC-enabled Nokia 6212 phone was used as pointing and tag reading device. Each subject contributed with 15 measurements per setting.

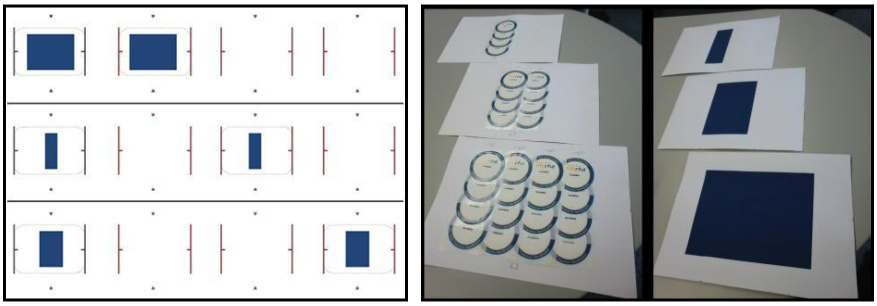

Fig. 1. Left: Three sample configurations for the Fitts' Law study with three different distances between targets (30, 60 and 90 cm). Right: Tiles for the Fitts' Law study (back side with tags and front side) with different widths (1, 2 and 4 tags).

The results of the average measurements can be seen in Table 2. Using extensive video analysis, we created two datasets, one which included runs where reading the NFC tag failed (with errors) and one that excluded those (w/o errors). Two interesting aspects can be seen: first, the effect of such errors is not visible for UI elements of width 1 tag. Second, without considering errors the times do not seem dependent on the width of the elements. Both observations can be grounded statistically: the differences between measurements including and excluding errors are not significant for a width of 1 tag ($p>0.05$) but are significant for all runs with 2 or 4 tags ($p<0.05$).

Further, a linear regression analysis of the measurements with respect to Fitts' Law parameters reveals a very low correlation factor of only 0.29.

These results can be explained by the very nature of these errors. Although the NFC-tags have been attached so close to each other that they overlap, there still remains some noticeable space between two tags where the phone is not able to read any of those tags. This directly explains the first finding. The second finding can be explained similarly as the users quickly adapted to these errors and started trying to touch exactly the spot that they found out to be working. Although it should be easier to hit a wider target, users voluntarily restricted their target width to 1 tag.

Since Fitts' Law does not seem to be a good approximation, we propose to distinguish two cases: pointing to a close target (distances of about 30cm or less) from far targets. Using the overall average time for 30cm and subtracting the NFC reading time, this yields a value of P(close) = 0.58s. For other distances P = 0.83s.

Table 2. Measurements for pointing (static UI) in seconds without & with technical errors.

Tile width	**1 tag**			**2 tags**			**4 tags**		
W/o errors	0.98s	1.05s	1.23s	0.99s	1.07s	1.26s	0.99s	1.05s	1.23s
W errors	1.00s	1.07s	1.25s	1.07s	1.12s	1.35s	1.16s	1.10s	1.34s

Dynamic Fitts' Law Study and Results

The dynamic version could have directly copied the setup from the static version. However, in order to reduce the effect of subject adaptation to errors in hitting between NFC tags, we modified it to a more dynamic version. A sequence of boxes of varying sizes appeared at varying locations on the dynamic NFC-displays, one appearing after the previous one had been touched. The sequence of positions and dimensions was randomly generated once and was the same for all subjects and all runs. Each subject was asked to do two runs, each with 21 rectangle touches, resulting in 20 measurements per run.

As the sizes and distances were random on a continuous scale, one cannot simply create a table as in the static case. When applying the necessary calculation to compute the index of difficulty (ID) of Fitts' Law from the distance and width values, the result should reveal a roughly linear relationship between ID and time. However, it turns out that the correlation factor is only 0.21. Therefore, it is difficult to justify a linear relation as postulated by Fitts' Law in the dynamic case. In line with the argumentations from the static case, we propose to apply the pointing times from the static case also in dynamic tag-based applications.

4.4 Sequences of Touch-Interactions

The second part of the study tries to find improvements to the current rules for placing Mental Act (M) operators. This is an important aspect as the number of Mental Act operators within a KLM can largely influence the overall time estimation. Set as 1.35 seconds [5], this operator is one of the largest ones. The study will shed light on when and where to place this important operator. More precisely, the effect of the following independent variables on Mental Act operator placement was measured:

1. **Sequence Length (1 / 3 / 5)**: the number of tags to be read consecutively
2. **Tag Distance (close / far)**: the physical distance between consecutive tags within a sequence
3. **Interaction Type (simple touching / confirmation)**: presence of attention shifts between physical UI and mobile device
4. **Interface Type (static / dynamic)**: the type of physical interface

We designed a set of physical UIs in order to test all parameters. The combination of the variables "Interaction" and "Interface Type" results in 4 different prototypes. First, a static and a dynamic UI have been built with the same visual setup (see Fig. 2). Second, for each UI, two mobile prototypes have been created: for the first one, touching a tag is sufficient for the tasks while for the second one, the phone displays a dialog after each touch that needs to be confirmed by pressing one of the softkeys (left or right) on the mobile phone. The mapping of the "OK" button to either of them was randomized for all tasks. This ensures that users definitely have to look at the screen. In addition, the combination of the variables "Sequence Length" and "Tag Distance" results in 6 tasks. We chose sequence lengths of 2, 4, and 6 tags (i.e. 1, 3, and 5 pointing actions from one tag to the next). Tag distance only distinguished 'close' and 'far' distances, representing distances of less and more than 40cm.

Each subject carried out each task with each prototype (counterbalanced). Again, the tasks were executed with a Nokia 6212 phone. Time measurements were done either on the phone itself (static versions) or on the server (dynamic versions). In addition, we taped all subjects in order to spot errors that occurred during the tasks.

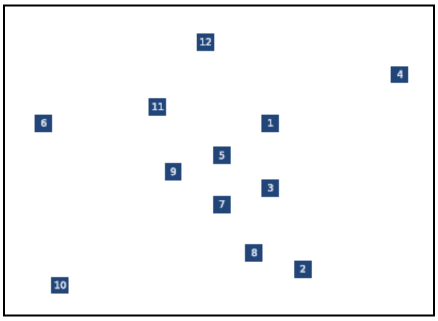

Fig. 2. NFC tag layout for the touch sequence study. Odd numbers are close together.

During the analysis of the retrieved data, all erroneous runs (user errors as well as technical errors) were discarded. The amount of errors in general was very low. Since single tag sized spots were used especially technical errors hardly ever occurred as, mentioned in the pointing experiment, gaps are the main issue for technical errors.

Interaction Type "simple touching"

This section covers the Interaction Type without forced attention shift, i.e. without needing to confirm the tag reading. Table 3 shows the average values for all values of the other 3 independent variables. It is not completely clear why there is a (in 4 of 6 cases even significant, on average 16%) slow-down of the interaction in the dynamic version compared to the static system. Some comments from the subjects indicate that a slightly more glaring and slightly less clear dynamic display in contrast to the matt and high resolution printout might have made the interaction slightly more difficult. However, we do not have strong evidence for that and a more detailed inspection of this effect will remain future work.

Two further observations can be made: first, as expected, measured times for the Tag Distance “far” experiments are higher than for Tag Distance “close”. Second, and perhaps more interesting, interaction times increase linearly when increasing the number of tags to be read. The 4 combinations far/dynamic, far/static, close/dynamic, and close/static reveal a very good linearity with a R^2 of 0.89, 0.93, 0.85, and 0.97, respectively. This strongly indicates that there is no need to place any additional Mental Act operators for longer interaction sequences. Of course, this result is limited to sequences of no more than 6 tags, which however, in our experience, seems reasonable with most interaction scenarios. Thus, we conjecture: **Sequence Length and Tag Distance do not influence M placement.**

Table 3. Total performance times for Interaction Type “simple touching”.

Task	Close, 1	Close, 3	Close, 5	Far, 1	Far, 3	Far, 5
Static	1.00s	3.19s	5.31s	1.29s	3.87s	6.32s
std. dev.	0.10s	0.32s	0.38s	0.33s	0.61s	0.66s
Dynamic	1.29s	3.31s	6.58s	1.47s	4.41s	7.16s
std. dev.	0.32s	0.38s	1.39s	0.62s	0.81s	0.99s
Significance	**p<0.01**	**ns**	**p<0.01**	**ns**	**p<0.05**	**p<0.05**

Interaction Type “confirmation”

This Interaction Type uses forced attention shift to model mobile interactions with physical UIs that require users to switch their attention to the mobile device, e.g. for confirmation or input from the keyboard. Table 3 shows the averages for all values of the other 3 independent variables. The same effects as with the Interaction Type “simple touch” can be observed. The slow-down of dynamic interface interaction with respect to the static version is again measurable (significant for all cases, on average 26%).

Table 4. Total performance times for Interaction Type “confirmation”.

Task	Close, 1	Close, 3	Close, 5	Far, 1	Far, 3	Far, 5
Static	1.76s	5.50s	10.58s	2.28s	7.51s	12.10s
std. dev.	0.17s	0.36s	1.11s	0.41s	0.98s	1.22s
Dynamic	2.42s	7.47s	12.89s	2.75s	8.89s	14.94s
std. dev.	0.43s	1.35s	1.32s	0.69s	1.55s	2.82s
Significance	**p<0.01**	**p<0.01**	**p<0.01**	**p<0.05**	**p<0.01**	**p<0.01**

Furthermore, the two previous observations can again be made: first, measured times for the Tag Distance (far) experiments are higher than for Tag Distance (close). Second, interaction times increase linearly when increasing the number of tags to be read. The 4 combinations far/dynamic, far/static, close/dynamic, and close/static also reveal a very good linearity with a R^2 of 0.88, 0.95, 0.93, and 0.96, respectively. This strongly supports the conjecture from the “simple touch” experiment.

In addition, a video analysis showed that attention shifts were in fact made, but that subjects became used to this very quickly and became able to habitually glance at the display and press the correct button, in most cases even without retracting the mobile device from the physical UI. Pauses of any kind before attention shifts were not observed which indicates that in such repetitive sequences mental preparations play a very minor role. Thus, we conjecture: **Interaction Type (attention shifts) does not influence M placement.** Also, even though there was a significant difference between dynamic and static interfaces (Interface Type), it most likely results from aspects inherent to the projection setup but definitely cannot be circumvented by adding Ms. **Interface Type (static / dynamic) does not influence M placement.**

Recommendation for Mental Act Operator Placement

In our experience, hardly any mental preparation happens during touch sequences, as the rather accurate modeling of the measured times without Ms show. It was however observed, that before a sequence subjects did actually pause for a brief time which could be assumed as mental preparation. Thus, we propose to generalize the placement rules of M and our recommendation can be summarized into: **Add Mental Act (M) operators before cognitive chunks.**

Based on their concept in cognitive psychology, Card et al. [5] define chunks as “highly integrated sub-methods that show up over and over again in different methods." These chunks “usually reflect syntactic constituents of the system's command language." Although the identification of such chunks can be difficult, this recommendation matches more with the guidelines presented by Kieras [13]. The most important advice is to identify chunks and thus place Ms consistently, especially when pitting different UI designs against each other. The assumption is, that making the same placement ‘mistakes’ in all alternative designs still yields a good comparison although the estimated times might highly deviate from actual expert execution times.

As a brief illustration of the impact of the new model proposal, take the following example as described above, touching and confirming the touch on 5 tags that are located close to each other on a static interface (i.e. Sequence Length “5”, Tag Distance “close”, Interaction Type “confirmation”, Interface Type “static”). Table 4 shows an average measured value of 10.6 seconds.

Original KLM: For each of the 5 NFC reads, the user switches attention to the display S_{macro}, points to the tag P, and waits for the read to finish R(NFC). Additionally, each time, the confirmation needs a Mental Act M, an attention shift to the phone S_{macro}, and a key press K(hotkey). This adds up to 18.6 seconds.

Proposed KLM: according to our new model, the Mental Act is not included, the pointing operation uses the P(close) operator, and the value for R(NFC) is updated. This adds up to 10.5 seconds which is much closer to the measured time of 10.6s.

4.5 An Updated Mobile KLM – Summarizing the Results

Based on the studies and their results, this section summarizes the updated Mobile KLM for NFC-based mobile applications. As described in the section about shortcomings of the original Mobile KLM, four major aspects have been investigated. We now present the results of the studies with respect to each of them:

1. System response time reading NFC tags with current mobile devices
2. Model errors caused by mismatches between tag-matrix and UI widgets
3. Pointing time for close and far objects
4. Chunking of action sequences needs to be reevaluated

Item (1), see Section 4.2: System response time will always be a parameterized operator that will have to be measured for each specific setup. However, it helps future modelers to provide current values for specific devices. First, they do not have to perform exhaustive studies themselves; second, it can help to know the different values for different devices to decide which ones should be targeted at. In combination with previous work (Nokia 3220 from [10]), we now have as value for reading an NFC tag:

$$\mathbf{R_{NFC}(Nokia\ 3220{+}NFCShell) = 2.58s}$$

$$\mathbf{R_{NFC}(Nokia\ 6131) = 1.20s}$$

$$\mathbf{R_{NFC}(Nokia\ 6212,\ static) = 0.40s,\ R_{NFC}(Nokia\ 6212,\ dynamic) = 0.62s}$$

$$\mathbf{R_{NFC}(Nexus\ S,\ static) = 0.34s,\ R_{NFC}(Nexus\ S,\ dynamic) = 0.56s}$$

Item (2): Although errors are usually not modeled within a KLM, it does make sense to incorporate some treatment of those if they occur even despite following the expert user assumption. Mostly due to the fact that NFC devices have not been initially designed for multi-tag UIs, tag reading accuracy depends on several factors. One of the most prominent ones is the number of tags that render a displayed UI element interactive. If more than one tag is used for, e.g. a button, small gaps appear between those tags. If users touch with their phone on one of those gaps, the probability is high that the touch attempt will be missed by the system. The KLM does not include a mechanism to specify that, e.g., every 5th touch needs a higher value or an additional attempt. Therefore, we propose to model these issues by adding a small amount of time to the NFC Action operator. This models additional time that occurs with some low probability over time. As this applies mostly to multi-tag interaction and can be neglected for single-tag interfaces, the operator is zero for the single-tag case. The other values result from subtracting the measurements of all reads without technical errors from those with errors and dividing this by the number of interactions. As these are on average quite different for static and dynamic interfaces, we propose using different values for each of the action operators:

$$\mathbf{A_{NFC}(single\text{-}tag) = 0.00s}$$

$$\mathbf{A_{NFC}(multi\text{-}tag,\ static) = 0.07s,\ A_{NFC}(multi\text{-}tag,\ dynamic) = 0.32s}$$

Item (3), see section 4.3: If a fine-grained model for an interface that requires many pointing operations needs to be created, the pointing operator should be implemented using a specific model such as Fitts' Law (provided there is no indication that the interface does not adhere to it). However, in practice, this is rarely done as it complicates the generation of the model. Already in the earliest uses of the model (see [5]), it has proven to be sensible to use constant average values. However, we found that the original value of about 1 second [5] is overly high. In addition, it makes a significant difference whether two consecutive touches are close (within 30cm) or not. Thus, we propose two constant values depending on the distance between source and target of a pointing operation:

P(close) = 0.58s, P = 0.83s

Item (4), see section 4.4: The Mental Act operator is of high importance in most KLMs as it usually occurs quite often and, with 1.35 seconds, all occurrences together add up to a considerable amount of time. Our proposal for a placement rule is:

Add Mental Act (M) operators before cognitive chunks.

Although the accurate identification of such a chunk is difficult, it is clear that not the specific place but only the number of such operators counts [13]. In addition, especially when comparing different interfaces, it is more important to be consistent in operator placement than to be 100% in accordance with reality. With the studies described above, we were able to provide additional insights:

- A chunk is independent of the length of a touch sequence (if the sequence is known to the user beforehand)
- Additional mental preparations are not necessary for pointing longer distances with respect to shorter ones

5 Validation of the Updated Mobile KLM

In order to find out whether the observed results yield to a valid model and to test the superiority with respect to the existing model, we conducted two additional studies with example applications to validate the applicability and accuracy of the updated model. We recruited the same subjects from previous studies to perform the additional studies. Prior to the experiments, KLM models, i.e. the operator sequences were put together both with the existing Mobile KLM and the updated version so as to compare them later on with the actual execution times and each other. Distractions were not taken into consideration since these were not present in the lab setup. It should be noted that, when modeling a task with the original Mobile KLM, we already used the updated time value for the Nokia 6212. This is in line with the interpretation of the System Response Time operator R which is to be parameterized according to the hardware in use.

5.1 Validation Scenario 1: Munich Map Poster

The first validation used a prototype by Broll and Hausen [4] that was designed for different tasks on a map of sightseeing places in the city of Munich. Examples for such tasks include finding information about specific points of interest and getting route directions. A poster with NFC-tags was used as a static physical UI and again, a Nokia 6212 with a J2ME application was used as the input device. This application measured the times between the first touch of a tag and the last user action either by touch or key press. To avoid learning effects, the tasks of this study were counterbalanced with a Latin square design. Tasks always began with touching a “start” tag which started the measurement. Prior to the actual runs and measurements, subjects were instructed to familiarize themselves with the prototype by touching various tags and exploring the phone UI. The following tasks had to be performed:

- **Task 1** merely consisted of consecutively retrieving information from two points of interests, i.e. tags. The KLM for touching 2 POIs is 2*(P, R(NFC)) which amounts to 2.80s and 2.46s for the old and new model, respectively. The differences stems from the updated pointing time. In comparison, the average of the actual user data was 2.46s.
- **Task 2** consisted of selecting 3 nearby points of interest and having the system generate a route between them. As this task is mostly a subtask of Task 3, we refer to Task 3 for the models. The time estimations are 6.07s (old model) and 4.04s (new model) with the user data averaging to 4.75s.
- **Task 3** (see **Table 4**) is similar to Task 2 but adds one far pointing operation and increases the route size to 5 POIs. The results are summarized in Table 4 which shows the predictions for the third task as modeled and calculated with the original Mobile KLM, with our proposed updated KLM as well as the average from the performance time measured with the study. The targets of all pointing operations are quite close which is reflected in the new model. Also, the original KLM would force an M operator before a macro attention shift. The reason this is not done in the updated KLM is because it can be seen as part of a single chunk where selecting the points directly belongs to the route generating subsequence.

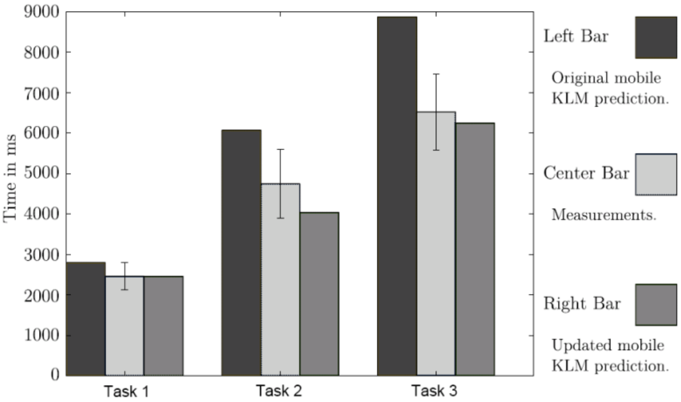

Fig. 3. Comparison of the existing model, the times measured within the study, and the new model for 3 tasks for the Munich Map Poster scenario.

Fig. 3 summarizes the modeled and measured values for the 3 tasks. For the short Task 1, the original KLM predicts rather precisely but is already outperformed by the updated model. With increasing task length, the prediction error of the original Mobile KLM increases. In general, it can be observed that the original KLM tends to predict longer times than are actually measured. This is largely alleviated by the new model which tends to predict shorter times, but is considerably more accurate. This result is also more in line with the original concept of KLM which is supposed to model a lower bound for execution time [5] as dictated by the expert user assumption.

Table 4. Validation “Munich Map Poster”, Task 3 long route generation.

Interaction	Original KLM		Updated KLM	
Touch POI_1	P, R(NFC)	1.40s	P, R(NFC)	1.23s
Touch POI_2	P, R(NFC)	1.40s	P(close), R(NFC)	0.98s
Touch POI_3	P, R(NFC)	1.40s	P(close), R(NFC)	0.98s
Touch POI_4	P, R(NFC)	1.40s	P(close), R(NFC)	0.98s
Touch POI_5	P, R(NFC)	1.40s	P(close), R(NFC)	0.98s
Look at phone	M, S(macro)	1.71s	P(short), S(macro)	0.94s
Press hotkey	K(hotkey)	0.16s	K(hotkey)	0.16s
Total		**8.87s**		**6.25s**
Study Average	**6.52s**			

5.2 Validation Scenario 2: London Map Projection

The second validation is based on a prototype that shows a map of London on a dynamic NFC-display, similar to the one that was used for the dynamic UI experiments and the introductory game before the experiments. With this prototype, users can touch the dynamic NFC-display with their mobile devices to see details about hotels and sights. Again, a Nokia 6212 and a J2ME client were used in conjunction with a J2SE server producing the dynamic UI. Task execution times were measured via video analysis from the point right after bringing the phone close to an active area on the physical UI, i.e. right before the first NFC tag reading. The active areas on the dynamic UI were all the size of one tag, so technical errors as described above did not have to be taken into consideration at all. To avoid learning effects the tasks were counterbalanced with a Latin square design. Again, subjects were instructed to familiarize themselves with the prototype by touching various tags and exploring the phone UI prior to the actual runs and measurements. The following tasks had to be performed:

- **Task 1** mainly consisted of touching a tag to show all hotels, select one hotel, and touch a tag to add it to the favorites. The models consist only of P, R(NFC), and system response times. The difference of 6.54s (old model) and 5.53s (new model) stems from the changed time for the Pointing operator P. The measured average time was 5.39s.
- **Task 2** merely increases the length of the task. The difference between the existing and the updated model becomes more pronounced as the time estimates for the pointing operations were estimated as too high by the existing model. Results are: 11.7s (old model), 9.69s (new model), and 10.2s for the study measurements.
- **Task 3** consisted of a longer task. First, a sight needed to be added to the favorites. Second, this sight should again be removed from the favorites; however, the phone menu should be used for this instead of the tag matrix interface. The results are summarized in Table 5 which shows the predictions for the third task as modeled and calculated with the original Mobile KLM, with our proposed updated KLM as well as the average from the performance time measured with the study subjects. The application dependent constants R_0=0.20s, R_1=0.60s, R_2=0.57s, R_3=0.57s, and R_4=1.97s are System Response Time operators that we measured beforehand.

Table 5. Validation “London Map Projection”, Task 3 using NFC and phone interface.

Interaction	**Original KLM**		**Updated KLM**	
Touch tag (show sights)	R(NFC), R_1	1.00s	R(NFC), R_1	1.00s
Touch POI	P, R(NFC), R_2	1.97s	P, R(NFC), R_2	1.80s
Touch tag (add to)	P, R(NFC), R_0	1.60s	P(close), R(NFC), R_0	1.18s
Touch POI	P, R(NFC), R_0	1.60s	P, R(NFC), R_0	1.18s
Look at phone	M, S(macro)	1.71s	P(short), S(macro)	0.94s
Navigate to favorites	3*(M, K(hotkey))	4.53s	M, 3*K(hotkey)	1.83s
Open favorites	K(hotkey)	0.16s	K(hotkey)	0.16s
Select POI	K(hotkey), R_4	2.13s	K(hotkey), R_4	2.13s
Remove POI	M, K(hotkey), R_3	2.08s	K(hotkey), R_3	0.73s
Total		**16.8s**		**11.0s**
Study Average	**13.2s**			

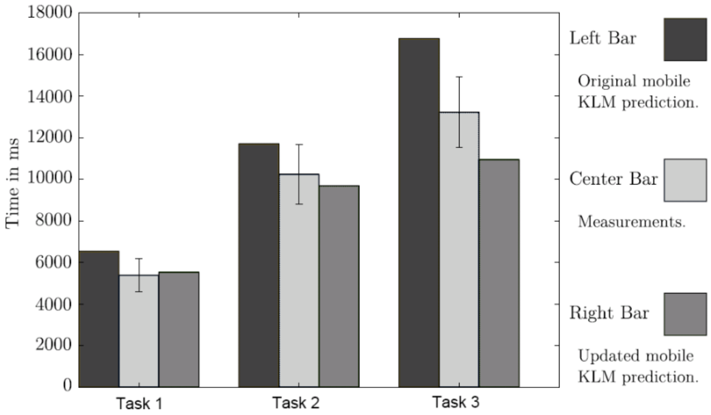

Fig. 4. Comparison of the existing model, the times measured within the study, and the new model for 3 tasks for the London Map Projection scenario.

For the models, we had to measure some system response times that occur when waiting for all hotels being displayed, show information on the phone, etc. These were the same for the old and the new model. Besides the reduced time for pointing operations, the largest difference stems from the difference in M placement. In contrast to the new model, the original Mobile KLM prediction forces an M in front of the Macro Attention Shift for looking at the phone UI. Also, when navigating the mobile menu, the original Mobile KLM would place an M before nearly each key press [10]. Fig.4 summarizes the modeled and measured values for the 3 tasks.

Though deviating, the original Mobile KLM seems to predict execution times quite accurately for Tasks 1 and 2. However, the prediction for task C is way off. It predicts a 27% longer time than the measurements show. Except for one case (the predicted time for Task 1 is about 3% longer than the measured time), the times predicted by the updated KLM are shorter than the measured times which is a good indication for the intended applicability of the KLM for predicting a lower bound for execution time [5]. In summary, the two validation experiments very well illustrated the increased accuracy of the four aspects that have been identified for a need to update.

6 Summary and Conclusion

The Keystroke-Level Model (KLM, [5]) has already been used to model interactions and predict task execution times for a great variety of applications. For advanced interaction with mobile devices, an extension to the original, desktop-based KLM has been described in [10] (Mobile KLM). In this paper, we have revisited and revised the Mobile KLM in several aspects, especially focusing on interactions with interfaces featuring multiple NFC tags. These improvements have been made taking two important categories of such (multiply) tagged applications into account: static and dynamic interfaces. Main insights generated by our work include:

- Updated KLM values for the system response time of current phones for detecting and reading NFC tags
- Explanation and guideline to avoid errors using multiply tagged interfaces (use odd number of tags) and a way to model errors for those instances
- A more precise estimation of pointing time (distinguishing between 'close' and 'far' pointing, no difference between static and dynamic interfaces)
- Improved guidelines for placing Mental Act (M) operators

With these additions, it is possible for designers and developers to model interfaces that make use of advanced, multi-tagged UIs with the easy-to-use KLM. Several metrics for user interfaces such as user acceptance or fun are of course difficult to evaluate with user models. Still, the revised Mobile KLM allows a quick and simple estimation of end-user task performance times as well as a concise and meaningful comparison between interfaces. The main advantage of this approach is that such metrics can be generated quickly, cheaply, and without the prior implementation of any of the interfaces.

We thus hope that our work broadens the applicability of the KLM to novel and upcoming applications and improves the accuracy of the approach while keeping the simplicity of the model intact. Of course, this type of work will be ongoing as long as new types of interactions will be added. In general, including our previous and related work, we have thus far managed to cover a very broad array of interactions with mobile devices.

References

1. Bälter, O.: Keystroke Level Analysis of Email Message Organization. In Proc. CHI'00. 105-112 (2000)
2. Broll, G., Siorpaes, S., Rukzio, E., Paolucci, M., Hamard, J., Wagner, M., and Schmidt, A.: Supporting Mobile Service Usage through Physical Mobile Interaction. In Proc. PerCom'07. 262-271 (2007)
3. Broll, G., Graebsch, R., Holleis, P., and Wagner, M.: Touch to Play - Mobile Gaming with Dynamic, NFC-based Physical User Interfaces. In Proc. MobileHCI'10 (2020)
4. Broll, G. and Hausen, D.: Mobile and Physical User Interfaces for NFC-based Mobile Interaction with Multiple Tags. In Proc. MobileHCI'10 (2010)
5. Card, S.K., Moran, T.P., Newell, A.: The Keystroke-Level Model for User Performance Time with Interactive Systems. Communications of the ACM 23(7). 396-410 (1980)

6. Card, S.K., Newell, A., Moran, T.P.: The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Inc. (1983)
7. Fitts, P.M.: The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. Journal of Experimental Psychology 47(6). 381-391 (1954)
8. Hardy, R. and Rukzio, E.: Touch&Interact: Touch-based Interaction of Mobile Phones with Displays. In Proc. MobileHCI'08. 245-254 (2008)
9. Haunold, P. and Kuhn, W.: A Keystroke Level Analysis of a Graphics Application: Manual Map Digitizing. In Proc. CHI'94. 337-343 (1994)
10. Holleis, P., Otto, F., Hussmann, H., and Schmidt, A.: Keystroke-level Model for Advanced Mobile Phone Interaction. In Proc. CHI'07. 1505-1514 (2007)
11. Holleis, P.: Integrating Usability Models into Pervasive Application Development, PhD Thesis, Ludwig-Maximilians-Universität München, Germany (2009)
12. John, B.E. and Kieras, D.E.: The GOMS Family of Analysis Techniques: Tools for Design and Evaluation. Human-Computer Interaction Institute. Technical Report (1994)
13. Kieras, D.: Using the Keystroke-Level Model to Estimate Execution Times. The University of Michigan. Technical Report. http://www.pitt.edu/~cmlewis/KSM.pdf (1993)
14. Koester, H.H. and Levine, S.P.: Validation of a Keystroke-Level Model for a Text Entry System Used by People with Disabilities. In Proc. Assets'94. 115-122 (1994)
15. Luo, L. and Siewiorek, D.P.: KLEM: A Method for Predicting User Interaction Time and System Energy Consumption during Application Design. In Proc. ISWC'07 (2007)
16. Manes, D., Green, P., and Hunter, D.: Prediction of Destination Entry and Retrieval Times Using Keystroke-Level Models. Technical Report. University of Michigan (2007)
17. Mori, R., Matsunobe, T., Yamaoka, T.: A Task Operation Prediction Time Computation Based on GOMS-KLM Improved for the Cellular Phone and the Verification of that Validity, In Proc. ADC'03 (2003)
18. Ramírez-González, G., Muñoz-Organero, M., Kloos, C.D., and Astaiza, Á.C.: Exploring NFC Interactive Panel. In Proc. Mobiquitous'08 (2008)
19. Reilly, D., Welsman-Dinelle, M., Bate, C., and Inkpen, K.: Just Point and Click?: Using Handhelds to Interact with Paper Maps. In Proc. MobileHCI'05. 239-242 (2005)
20. Riekki, J., Salminen, T., and Alakärppä, I.: Requesting Pervasive Services by Touching RFID Tags. IEEE Pervasive Computing 5(1). 40-46 (2006)
21. Rohs, M., and Gfeller, B.: Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects. In Proc. Advances in Pervasive Computing. 265-271 (2004)
22. Sánchez, I., Riekki, J., and Pyykknen, M.: Touch&Control: Interacting with Services by Touching RFID Tags. In Proc. IWRT'08 (2008)
23. Seewoonauth, K., Rukzio, E., Hardy, R., and Holleis, P.: Touch&Connect and Touch&Select: Interacting with a Computer by Touching it with a Mobile Phone. In Proc. MobileHCI'09. 1-9 (2009)
24. Vetter, J., Hamard, J., Paolucci, M., Rukzio, E., and Schmidt, A.: Physical Mobile Interaction with Dynamic Physical Objects. In Proc. MobileHCI'07 (2007)
25. Want, R.: An Introduction to RFID Technology. IEEE Pervasive Computing 5(1). 25 (2006)