Article can be accessed online at http://www.publishingindia.com

Model-based Evaluation of Simulated Touch Screen Cell Phone Interface

Pantha Kanti Nath*

Abstract

This research compares two models for predicting the speed of users traversing through alternative menu hierarchies on a touch screen based cell phone simulated interface. Actual performance data from real users was compared with predictions of two alternative models - Fitts' and GOMS – and the GOMS model is shown more closely match the user data. The potential value of validated models is that, they can make it faster to compare alternative designs than with usability testing with end users. This validated models can become an efficient way for the UX designer and evaluator to predict ease of use. The efficacy of model-based evaluation is demonstrated through results of user studies.

Keywords: Cell Phones, Touch Screen, Menu Traversal, Cognitive Modeling, Model-Based Evaluation

1. Introduction

We are going through a period of rapid change as innovators seek to exploit new ways of interaction with mobile devices. Mobile user interface (UI) technology is in the midst of this evolutionary phase. Many popular mobile devices use touch as input. While touch input can be intuitive, it is relatively imprecise. Touch targets, such as icons, must be fairly large and widely spaced in comparison with the mouse and pointer style inputs. Phones that use Apple's iOS (iPhone), Google's Android architecture, Blackberry's operating system, HP's webOS, Samsung's Bada and Windows Phone 7 mobile operating systems, all offer diverse UI design approaches using touch screen interface technology. Touch screen phones are not only used for making and receiving calls or sending and receiving text messages but provide a wide range of functionality. They support an extensive range of applications accessible through an elaborate hierarchical menu.

Although the menu hierarchies appear easy to use, they are complex. Mobile phones mostly come with a static, hierarchical structure of the menu, which due to the small screen sizes work against many aspects of usability. One problem is that, not all information fits the small display and hence need to split into several different screens, causing a trade-off between content and user interaction. The number of screens depends on the depth and breadth of the menu structure (Ziefle, 2002), so the balance between breadth and depth of the hierarchical menu structure becomes an important factor. Deeper depth in the structure leads to navigation problems while breadth, in its turn, to crowding of functions on the screen (Chae& Kim, 2004). As the overall structure of the menu is not transparent, i.e. does not show through to the user and the users have, as a consequence, difficulties building a mental representation of the sub-menu they are navigating through (Ziefle&Bay, 2006; Ziefle, Arning& Bay, 2006).

In a test,Bay&Ziefle(2004) showed that cognitive processes have a large effect on navigation performance in small screen devices. High menu foresight, i.e. that the user can see many options, helps to pre-structure coming information and provide better orientation in the menus. People with good spatial ability performed very well in these conditions, however the ones with lower ability got information overload and disoriented. Therefore, design solution should not discriminate these people, but also not hinder strong people (Ziefle& Bay, 2004). Still, filling the small screen completely has negative effects on

* Department of Computer Science and Engineering National Institute of Technology Silchar.
Email: pantha.k.nath@gmail.com

navigation performance on mobile phones and should be avoided [Ziefle, Arning& Bay, 2006).

One of the outcomes of this study along with studies conducted earlier (St. Amant, Horton& Ritter, 2004; 2006) shows that, different menu hierarchies perform differently when subjected to user testing. Although the relative arrangement of menu items and the paths between them seem to be an insignificant aspect, their importance becomes prominent when they are comparatively studied in terms of usability.

Efficiency is one of the quintessential usability metrics, measured as time on task. For tasks done repeatedly on a cell phone interface, saving a couple of seconds can mean saving minutes per day and hours per week for users. This study has a narrow focus on efficiency (speed) attribute of usability and assumes errorless interaction by experts.

The idea is; if we can build a model of how a user works, then, we can predict how a user will interact with the interface. Fitts' and GOMS methods can be used to predict the time of users, though it's not the only target for them. The times for ordinary end users who do a task over and over can be predicted by the models,whereas, usability testing is a good method for finding the initial ease of use problems. But it does not find long term ease of use issues – inefficiencies with tasks that are done repeatedly. The models do that. The following two sections give a brief introduction to the two models for predicting this long term user behaviour.

1.1. Fitts' Model

Paul Fitts(Fitts,1954; Fitts&Peterson, 1964) demonstrated that the action of pointing to or tapping a target object could be measured and predictedmathematically, allowing to model the ease at which a person could perform the same action with a different target object. Fitts' Law, at its simplest form, states that the bigger an object and the closer it is to us, the easier it is to move to. A target object, in the context of UI, can be any interactive element (e.g. as a submit button, an input field in a form). The quicker you can reach a target object, the most convenient and easy it is to use. Fitts' Law is a binary logarithm. This means that the predicted results of the usability of an object run along a curve, not a straight line. A very small object will become significantly easier to click when given a 20% size increase, while a very large object will not share the same boost in usability when given the same 20% boost in size. Apart from the size of the target object, another primary factor in Fitts' Law is the distance between where the pointer currently is and where it needs to be. Placing key interface components far apart will increase the amount of time required in completing sequential tasks in an interface.

The Fitts' Law (Fitts,1954) states that the movement time (MT) is a function of target amplitude or distance (A) and target width (W). This study is based on Mackenzie's (Mackenzie, 1995) version of Fitts' Law in which the movement time (MT) follows the equation:

$$MT = a + b \times \log_2 (A / B + 1) \quad (1)$$

The second term of the equation (1): $log_2\ (A/W+1)$ is known as the index of difficulty ID, where a and b are constants derived empirically. They can be interpreted by the y-intercept and the slope of a predictive linear regression equation (MacKenzie, 1995; MacKenzie*et al*., 1991).Fitts' Law is an intensively used tool in Human Computer Interaction. It can be used in assisting interface designs (theinvention of pop-up menus), in interface evaluation of touch screens (Sears &Shneiderman,1991), and standard GUI with lots of small target areas. Fitts' Law based evaluation has been carried out for modeling cell phone interaction with keypads (Mackenzie, 2003), this further provided the required stimulus in pursuing this study.

1.2. GOMS Model

DefineGOMS is modeling technique (more specifically, a family of modeling techniques) (John,2003; Kieras,1999; John &Kieras,1996a, 1996b; Gray, John&Wood,1993) that analyses the user complexity of interactive systems. GOMS is a theory of the cognitive skills involved in human-computer tasks. It is based upon an information processing framework that assumes a number of different stages or types of memory (e.g., sensory store, working memory) with separate perceptual, motor, and cognitive processing. GOMS methods for task analysis produce hierarchical descriptions of methods and operators needed to accomplish goals; some GOMS models have been strikingly successful in critical Human Computer Interaction domains (Kieras,1999). It is used by interface designers to model user behaviour. The user's behaviour is modeled in terms of Goals, Operators, Methods and Selection rules.

Briefly, a GOMS model consists of Methods that are used to achieve the Goals.

- Goals are what the user is trying to accomplish. These can be defined at various levels of abstraction, from very high-level goals (e.g. COMPOSE an SMS) to low-level goals (e.g. TYPE a character 'C'). Higher-level goals are decomposable into sub goals, and are arranged hierarchically.
- Operators are the elementary perceptual, motor or cognitive actions that are used to accomplish the goals (e.g. DOUBLE-CLICK-MOUSE, PRESS-A-KEY). Operations are atomic elements in GOMS model. Further, it is generally assumed that each operator requires a fixed amount of time for the user to execute, and that this time interval is independent of context (e.g. The CLICK - MOUSE button takes a constant time to execute).
- A method is a sequential list of operations that the user performs and (sub) goals that must be achieved. If there is more than one method, which may be employed to achieve a goal, a Selection rule is invoked to determine what method to choose, depending on the context. For example, one method to accomplish the goal DELETE-WORD in the Emach text editor would be to MOVE-MOUSE to the beginning of the word, and PRESS-ALT-D-KEY-COMBINATION (the use-mouse-delete-word method). Another method to accomplish the same goal could involve using the arrow keys to reach the beginning of the word and delete word (the use-arrows-delete-word method).
- Selection rules generally take the form of a conditional statement, such as "if the word to be deleted is less than three lines away from the current cursor location, then use the use-arrows-delete-word-method, else use the use-mouse-delete-word method".

As a theory of Human Computer Interaction, GOMS (Card, Moran& Newell, 1983) models can be classified as predictive, descriptive, and prescriptive. A GOMS model is predictive because it can be used to predict the time it will take a user to perform the tasks under analysis, as long as the developer can come up with time estimates for the operators involved in each model and the user does not make an error. A GOMS model is descriptive in the sense that it is a representation of the way a user performs tasks on a system. The methods, sub-goals and selection rules provide the designer with a description of the process, rather than simply a time estimate. A GOMS model can also be considered prescriptive because it can serve as a guide for developing training programs and help systems.

Figure 1: The MotoRokr E6

Once a GOMS model has been developed for achieving a certain goal, this model can be used to teach new users how to achieve the goal. Because GOMS can provide quantitative estimates of task execution time, it can properly be considered a verifiable theory.

A GOMS model provides the designer with a model of a user's behaviour while performing error-free well-known tasks. It can predict the time it will take for the user to carry out a goal (assuming an expert user with no mistakes). This allows a designer to profile an application to locate bottlenecks, as well as compare different UI designs to determine which one allows users to execute a task quicker. It seems likely that models like Fitts' and GOMS for accurately predicting performance can make a substantial contribution to the design and evaluation of menu hierarchies. This study differs from previous ones (St. Amant, Horton& Ritter, 2004; 2006) where model based studies were conducted to evaluate menu hierarchies on cell phones with keypads; whereas, this study makes a comparison of two models on touch screen interfaces.

A touch screen cell phone differs from an ordinary cell phone in many aspects: The number of menu items displayed on a single screen is generally larger. Menu items can be selected directly using finger or stylus thereby minimising scrolling actions using buttons in ordinary cell phones. Also, there is no standardisation in the distances traversed in moving from one item to another, unlike standard cell phones where the distances between the buttons is fixed. This fact plays an important role in the calculation of time taken in a traversal.

1.3. Menu Hierarchies

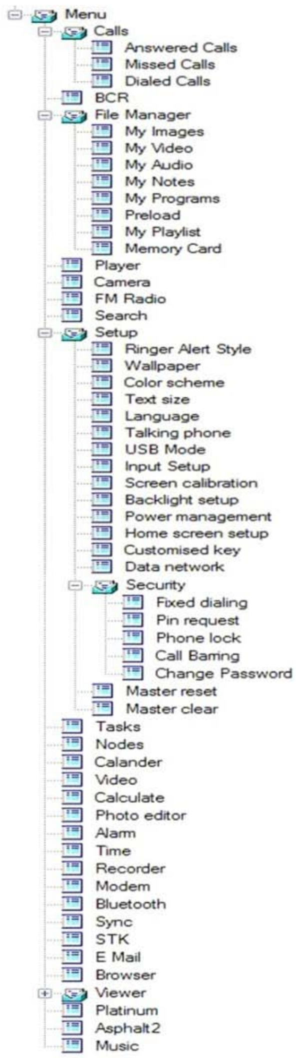
Users have our own personal interests, preferred settings and favourite functions. This variety should be taken into account when developing mobile UIs with static menus. The customisation possibilities in recent mobile phone models are however not flexible enough to handle the individual preferences (Yukikazu&Mitsuko, 2006) and limited to some simple graphical attributes such as colour themes, welcome texts and background pictures. On some mobile phone models a few shortcuts can also be changed and set by the user. Keeping in mind that users would be using phones functionality of their interests with varying frequencies, the static, hierarchical structure of the menu hierarchy should be targeted for a fixed set of users having similar profile called 'user profile'. This is evident more about the outcome of this paper (menu hierarchy evaluation in the results section) that, no single menu hierarchy design will serve best for all classes of users. The final part of the paper provides a designer solution to address this issue.

1.4. Java Simulation versus Real Cell-phone User Interface

User interface evaluation based on simulation do not simulate the context where cell phones are used (Johnson, 1998) and lack the desired ecological validity. Interruptions, movement, noise, multitasking etc. (Tamminenet al.,2004) that could affect the user performance are not present in Java simulations. Mobility and test environment are assumed to set special requirements for mobile applications. Orientation (vertical or horizontal) and how the device is held is an import parameter for mobile interface design for optimum UX.

Even if there seems to be a common concern about the adequateness of evaluation based on simulation, evaluation based on real mobile device have been rather less (Kjeldskov&Graham, 2003). This may be due to data collection techniques such as activity logging in mobile device, think aloud, video recording or observations being difficult to perform.

Figure 2: Manu Hierarchy for MotoroKr E6



This study aims at evaluating error-free use of cell phone menu systems, that is, for tasks that are repeated frequently. For control user testing, it was not possible to collect data from users on their own cell phones, due to unavailability of functionality to change the menu hierarchy in chosen model.As a compromise, a Java application was used to simulate the cell phone display on touch sensitive LED monitor. The Java applications were equipped with a logging mechanism to record user activities and time taken. Figure 3 shows the message generated after the call barring task is completed.

The study conducted may be divided into following parts:

- The first part describes a small empirical study of the traversal of cell phone menus, along with two models for predicting user performance: a Fitts' Law model, and GOMS model. Model performance comparison was made on the menu hierarchy for reference cell phone Motorokr E6 (Figure2). GOMS was found to give good predictions of user behaviour.

- In the second part of the study, GOMS was used for evaluating three different menu hierarchies.
- Considering the results from evaluation of different menu hierarchies, in the final part of the paper, we turn to the issue of designer-level support for evaluating cell phone menus. Here an evaluation metric is used for cell phone menu hierarchy selection with respect to a specific set of user group.

2. Methods

The MotoRokr E6 (Figure 1) is used as a reference cell phone model for the study. It runs on the Linux OS. There is a line of the most used shortcuts at the top of the screen and another line close to the bottom of the screen. The top line has Main Menu, Messages, Number keypad and Contacts, which cannot be customised. The lower line has ring styles, Real player, Video Camera and two other shortcuts which can be customised. The menu hierarchy is shown in Figure2. A set of common utilities accessed by an average user frequently were chosen. The tasks are chosen in such a manner so as to demonstrate variations in the number of intermediate steps involved in the selection. All the tasks are available as terminal menu items in the MotoRokrE6.

The following sections discuss the participants, apparatus and experimental setup, design and procedures used in this study.

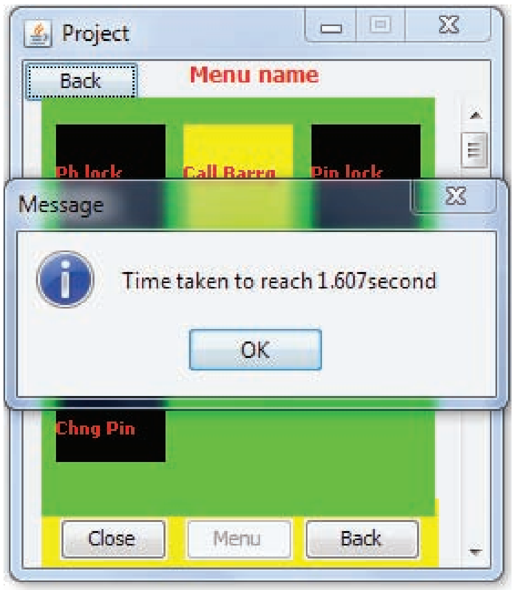
2.1. Participants

Forty-five undergraduate, graduate students, teaching assistants and staff (13 females and 32 males) voluntarily participated in the experiment. Their ages ranged from 18.3 to 36.7 years old, with a mean of 23.7 years and a standard deviation of 3.5 years. Thirty out of forty-five participants had experience in using touch cell phone; others had prior experience of general cell phone without a touch screen. They all had normal vision and no color-blindness.

2.2. Apparatus and Experimental Setup

The user-interface interaction is based on the use of a touch screen, assuming that users remain seated at a distance of 30cm from the touch screen, and point directly on the display objects by touching them using the finger. The hand is held down before starting the interaction, which constitutes the start position. After each pointing action, users returned to the start position, and the procedure continued like that. Adobe Photoshop cs3 was used to measure the distance and target width of interface elements for Fitts' model.

Figure 3: Record of the Total Time Taken to Reach "CallBarring".



2.3. Design and procedures

Two models were used for predicting user task time, each supported by a considerable background literature. A Fitts' law model and a GOMS model were applied independently of each other. The two models used the same simulated environment that involved over the course of this research. The simulation provides a common specification of the MotoRokr E6 cell phone, including the screen size and positions of menu items and the distances between them, as measured on the physical device.

The simulation also supports a common representation of the menu hierarchy shown in (Figure2, 4, 5 & 6).

Participants started with a practice stage, in which they familiarised themselves with the cell phone and its menu system. Each participant was shown how five terminal menu items were to be reached by printing form, as shown in the third column of Table 1. Each ">" represents a scrolling action, with commas separating consecutive selection actions. Reaching each of the terminal items (those at the end of each sequence) constituted a task in the study. Participants practiced each task until they could carry it out three times in a row without error to simulate a task performed frequently. Participants were required to abstain from PC use for one hour before the formal experiment to reduce chances of errors due to eye strain. This was kept in view that the study aims on expert use (practiced and error free).

Figure 4: Hierarchy 1 implemented as JavaTree

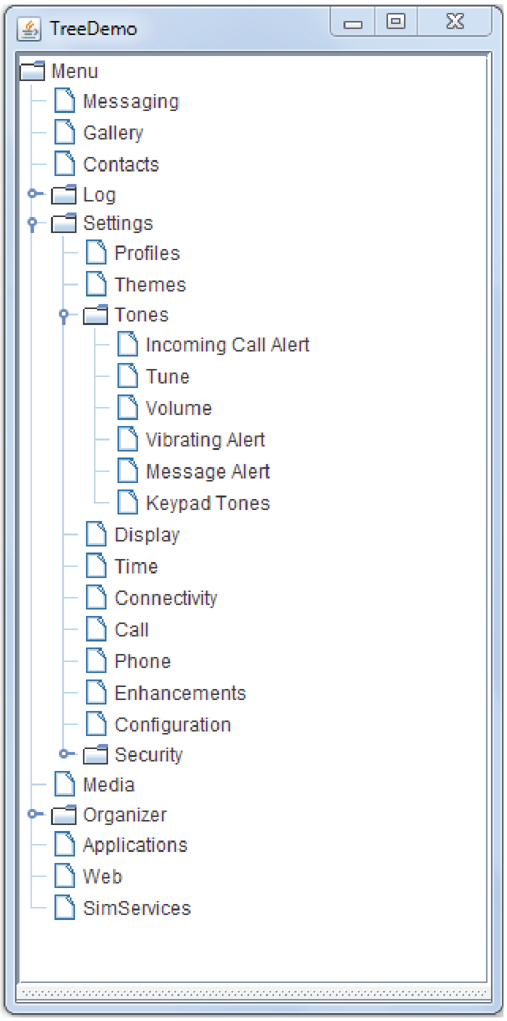
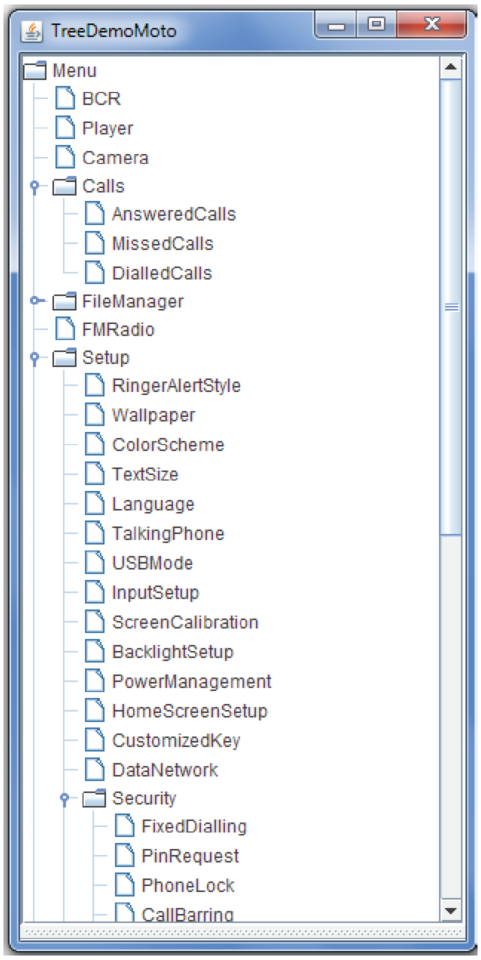


Figure 5: Hierarchy 2 Implemented as JavaTree



Each trial in the study required reaching one of the five target terminal items without access to the printed form. Tasks were presented to participants in a randomised order. Eight correct trials per participant (i.e., without errors or extraneous actions) were performed. Table 1 shows the mean duration for all five tasks over all participants in the study with standard deviations.

For user evaluations, participants were required to use these interfaces for the same set of target items and their activities and times were recorded. Then, the times taken were calculated by applying Fitts' and GOMS model to menu hierarchy of reference cell phone (Figure2). All the measurements involved in the Fitts' and GOMS analysis were based on the same interface used by the volunteers. Based on these times, the most efficient hierarchy was chosen. Incidentally, it was found that the times obtained from the logging mechanisms were very close to those predicted by GOMS model.

Figure 6: Hierarchy 3 Implemented as JavaTree

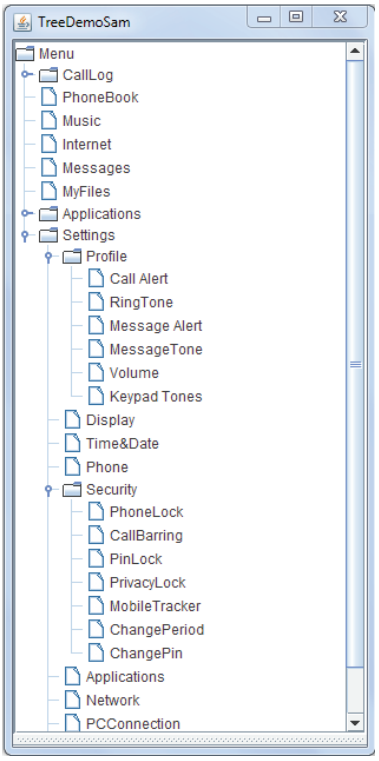


Table 1: Task chosen with Duration, Mean value & Standard deviation (in Sec)

Task No.	*Task Details, User Time*		
	Task	*Steps to reach menu item*	*Mean Time, Standard Deviation*
1	Missed calls	menu, calls, missed calls	1.61 (0.17)
2	Calculator	menu>>>>>>>calculator	3.77 (0.47)
3	Ringer volume	menu, setup, ringer alert style, edit, ringer volume	2.03 (0.32)
4	Time	menu>>>>>>>>>>time	3.54 (0.55)
5	Call barring	menu, setup>>>>>>>> security, call barring	3.75 (0.35)

Table 2: Movement time with Distance & Target width for Interface elements used

Task No.	*Movement Time For Interface Elements*		
	Element	*Distance, Target Width in cm*	*Movement Time in Sec*
1	Menu	3.2, 0.4	0.56
2	Calls	1.33, 1.1	0.25
3	Missed calls	1.31, 3.75	0.20
4	Calcula-tor	3.06, 1.1	0.3
5	Setup	3.75, 1.1	0.31
6	Alert	2.59, 3.45	0.23
7	Edit	3.63, 2.28	0.26
8	Time	3.06, 1.1	0.3
9	Security	1.86, 3.35	0.21
10	Call bar-ring	1.83, 3.35	0.22
11	Scroll	5.02, 0.3	0.44
12	Scrolling	--	0.18

Table 3: Task Duration using the Fitts' model (in Sec)

Task No.	*Task Details*		
	Task	*MT Calculation*	*Total Duration*
1	Missed calls	menu + calls + missed calls	1.01
2	Calculator	menu + scroll + 7 * scrolling + calculator	2.53
3	Ringer volume	menu + setup + ringer alert style + edit	1.36
4	Time	menu + scroll + 10 * scrolling + time	3.06
5	Call bar-ring	menu + setup + scroll + 8 * scrolling + secu-rity + call barring	3.08

Motivated with good results from GOMS, the second phase of this study evaluated menu hierarchy solely using the GOMS model. For this, three different menu hierarchies were considered. These menu hierarchies were based on three randomly chosen commercial available touch based cellphone interfaces. They were represented graphically using JavaTrees (Figure4, 5 and6) and simulated using Java applications.

The following sections give details of the model specific procedures involved in the study.

3. Model Specific Procedure

3.1. Predicting user time with Fitts' Model

MacKenzie's(2003) version of Fitts' Law for one finger typing for text entry on mobile phone suggested 0.176 and 0.064 for the values of constants a and b respectively (in equation 1), which was used in this study. The values for distance (A) and target width (W) are calculated by measuring physical distances from screen captures using Adobe Photoshop cs3 for individual interface items. Width (W) of a target is measured along the angle of approach from the starting point (MacKenzie& Buxton, 1992).

Table 2 shows various interface elements with its distance and width computed using Adobe Photoshop cs3 in third column. Scrolling action is taken to be a constant action requiring 0.176 Sec. Its fourth column gives the computed movement time for selecting respective interface elements. Table 3 shows the predicted time required for the completion of five tasks using Fitts' model.

Figure 7: GOMS method for selecting the Ringer volume

S1		Look at location for X.
S2		
	S21	If (found-X and not on-X),
		move-to X;
		select-X;
		Go to S4.
	S22	If (found-X and on X),
		select-X;
		Go to S4.
	S23	If (not found-X and on-scroll),
		scroll.
	S24	If (not found-X and not on-scroll),
		move-to scroll;
		scroll.
S3		Go to S2.
S4		Return with goal accomplished.

<B level>Predicting user time with GOMS Model

The paths to success is defined for each task in the study and are automatically generated from the menu hierarchy, specification, based on the same path traversals used for the Fitts' law model. Within a path, each step corresponds to the selection of a menu item. The steps are labeled for convenience in calculating time required. For example, the call barring task involves the steps of selecting the menu, settings, security, and finally call barring items, in sequence. Each of these steps in turn decomposes into a Select-X subtask, which involves scrolling until a given item X in the sequence is reached. The GOMS method for selecting the terminal item Ringer volume is shown in Figure7.

Time taken for user perception time is assumed to be 50 milliseconds as demonstrated by the method time measurement of Card, Moran and Newell GOMS model. Guidelines established by Kieras in his work on GOMSL and GLEAN3 (Kieras,1999), further established that all steps in Figure 7 have a duration computed from movements time, following the Fitts' model (Table 2) and a 50 millisecond duration for S1 and S4. S1 and S4 are the empirical estimation of the time a person takes to look at interface elements (user perception time) whereas the latteris an estimation of system response time. Table 4 shows the predicted time required for completing the five tasks using GOMS Model.

Table 4: Task Duration using the GOMS model (in Sec)

Task No.	*Task Details*		
	Task	*Steps Involved*	*Duration*
1	Missed calls	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.31
2	Calculator	s1,s21,s4,s1,s24,7*s23,s4,s1, s21,s4	3.99
3	Ringer volume	s1,s21,s4,s1,s21,s4,s1,s21,s4, s1,s21,s4	1.76
4	Time	s1,s21,s4,s1,s24,10*s23,s4,s1 ,s21,s4	3.31
5	Call bar-ring	s1,s21,s4,s1,s21,s4,s1,s24,8*s 23,s4,s1,s21,s4,s1,s21,s4	3.53

4. Results

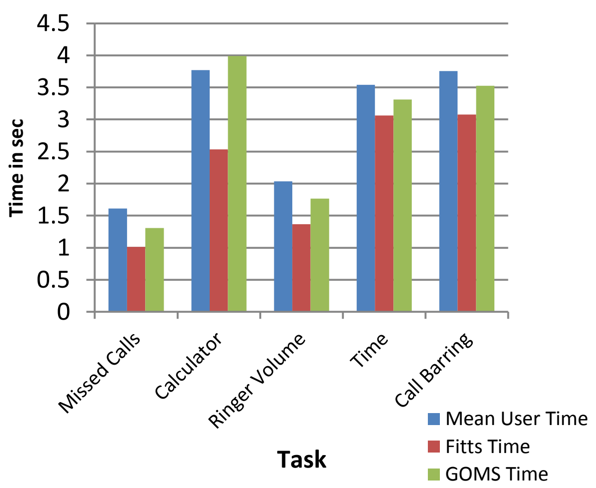
4.1. Performance Comparison: Fitts' and GOMS

The comparative chart in Figure 8 plots predicted time for Fitts', GOMS and mean user time against each of five tasks. Predicted time for GOMS was found to be nearest to mean user time with a maximum absolute deviation of 0.3 Sec. Fitts' predictions were not as close with a maximum absolute deviation of 1.2 Sec. Fitts' Law is often stated as valid, but limited. First of all, it does not address numerous factors other than target amplitude and width that affect user input performance. Secondly, the focus of Fitts' Law is to model low-level tasks of pointing and moving. It does not consider numerous other parameters, such as system response time, mental preparation time for the user, home timing, and selection rules for alternative methods, etc. (MacKenzie, 1991, PhD Thesis).

These intermediate and high-level activities have been addressed in the GOMS. In the menu selection with the assumption that users will not have memorised the positions of items in each menu, significant visual and cognitive processing is needed at each step of the process. This processing is not captured by Fitts' Law (St. Amant, Horton& Ritter, 2004).

GOMS performance is explained by the specific steps that are represented, the hierarchical structure in which they are combined, and depends on a cognitive processing framework that provides specific timing predictions (e.g. for Fitts' Law movements). The accuracy of a GOMS model is highly dependent on the estimates of the operator time values, a result that does not match the prediction may be explained away by variations in the operator time values for a specific user.

Figure 8. Comparative chart of User Model Predictions



GOMS accuracy with respect to real users reduces to the level of granularity of the analysis performed, GOMS is unable to capture random activities by the user, such as people scratching their nose, adjusting glasses, and other small physical movements. Further, some properties of tasks are not captured because either more than just procedural knowledge is involved or because the knowledge is not in the form of routine cognitive skill, but rather in a state that requires complex reasoning or problem solving. There is further scope for testing to get still better accurate results, but given current analysis, it seems sufficient for more detailed experiments involving the use of GOMS for evaluating menu hierarchies. Following section discusses the results of three menu hierarchies and their comparison.

Table 5: Task Duration using the GOMS model for Hierarchy 1 (in Sec)

Task No.	Task Details		
	Task	Steps Involved	Duration
1	Missed calls	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.22
2	Calcula-tor	s1,s21,s4,s1,s21,s4,s1,s21,s4,s1,s21,s4	1.76
3	Ringer volume	s1,s21,s4,s1,s21,s4,s1,s21,s4,s1,s21,s4	1.63
4	Time	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.22
5	Call barring	s1,s21,s4,s1,s21,s4,s1,s21,s4,s1,s21,s4	1.53

Table 6: Task Duration using the GOMS model for Hierarchy 2 (in Sec)

Task No.	Task Details		
	Task	Steps Involved	Duration
1	Missed calls	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.22
2	Calcula-tor	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.37
3	Ringer volume	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.14
4	Time	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.40
5	Call barring	s1,s21,s4,s1,s21,s4,s1,s21,s4,s1,s21,s4,s1,s22,s4	1.99

4.2. Menu Hierarchy Evaluation

The results of GOMS application to menu hierarchy 1, 2 and 3 are shown in Tables 5, 6 and 7 respectively. User mean and standard deviation for the three hierarchies are shown in Table 8. Predicted values from GOMS for the three menu hierarchies were compared with respect to a mean user time (Figure9, 10, & 11). It can be seen that predicted values are near to actual user mean values for almost all cases except time task in hierarchy 2 and hierarchy 3 in Figure 10 and Figure 11, respectively. GOMS predictions for missed calls and time task was found to be consistently lower than mean user time for all studied hierarchies. Hierarchy 1, hierarchy 2 and hierarchy 3 had a mean difference of 0.10 Sec, 0.14 Sec and 0.12 Sec respectively, with respect to predicted values across all five tasks. GOMS predictions for hierarchy 1 were found to be closest to mean user time with only difference of 2.64%, whereas; hierarchy 2 and 3 differed by 6.94% and 7.31% respectively. Considering these results we can say that predictions for hierarchy 1 are nearest to actual user values and so, hierarchy 1 seems to be the best choice considering forty-five participants and five chosen tasks of this study.

But this does not give us clear indication regarding the effectiveness of hierarchy 1 for all user groups and all tasks available on cell phones. This is evident by observing the predictions for hierarchy 1 which shows, prediction for the call barring task is almost exactly the same as observed, whereas predictions for the calculator and ringer volume tasks are over or under the estimated time respectively. Therefore, we require an evaluation process for the menu hierarchy dependent on usage patterns, because:

- Different users choose different items, and
- Items are chosen with varying frequency.

In other words, different usage patterns favor different designs. Next section considers these issues and discusses a scheme to make correct choice of a hierarchy.

Table 7: Task Duration using the GOMS model for Hierarchy 3 (in Sec)

Task No.	*Task Details*		
	Task	*Steps Involved*	*Duration*
1	Missed calls	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.22
2	Calcula-tor	s1,s21,s4,s1,s21,s4,s1,s21,s4	1.17
3	Ringer volume	s1,s21,s4,s1,s21,s4,s1,s21,s4,s1, s22,s4	1.26
4	Time	s1,s21,s4,s1,s21,s4,s1,s22,s4	1.01
5	Call bar-ring	s1,s21,s4,s1,s21,s4,s1,s21,s4,s1, s21,s4, s1,s21,s4	2.22

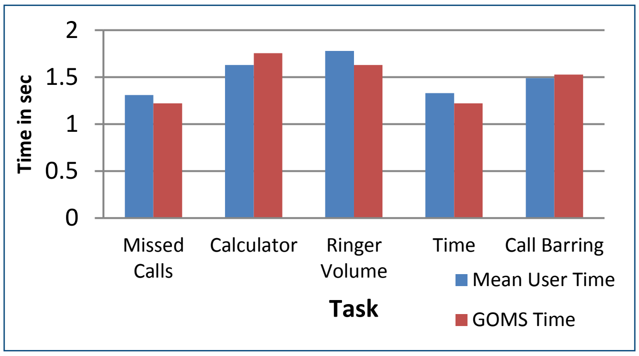
4.3. User Profile & Menu Hierarchy

Information about different user categories can be incorporated in the evaluation process. Comparing different menu hierarchy involves analyzing designs for a group of users called user profile. The user profileis defined as a probability distribution over the set of terminals in the hierarchy. The *coverage* of a profile reflects the ratio of users it represents with respect to all users of the cell phone (some profiles will be more common than others), and the *usage* of a profile gives the average frequency with which items are chosen (e.g., two profiles may contain similar probabilities, but one may apply to users who access their cell phonesseveral times as often as the other set of users). As an example, let's assume that 30% of the users of a given cell phone access only two items, View message and View call history, each on average twice a day. In the probability distribution of the profile for this set of users, these two items have a probability of 0.5 and all others have 0.0; the coverage of the profile is 0.30; and its frequency is 4.

Table 8: User time for three Hierarchy, Mean value & Standard Deviation (in Sec)

Task No.	*Mean Time, Standard Deviation*			
	Task	*Hierarchy 1*	*Hierarchy 2*	*Hierarchy 3*
1	Missed calls	1.31 (0.31)	1.34 (0.27)	1.27 (0.16)
2	Calculator	1.63 (0.43)	1.54 (0.39)	1.32 (0.2)
3	Ringer volume	1.78 (0.12)	1.09 (0.27)	1.18 (0.19)
4	Time	1.33 (0.19)	1.62 (0.3)	1.28 (0.35)
5	Call bar-ring	1.49 (0.32)	2.11 (0.21)	0.27)

Figure 9: Comparative chart of Model Predictions for Hierarchy 1



Formally, the design problem involves the construction of a mapping (in the form of a hierarchical ordering h) between T, the set of terminal menu items, and U, the set of all user profiles defined on T. A reasonable evaluation measure for a given menu hierarchy h is its efficiency: the expected cost of reaching a terminal item.Expected cost is an estimation of the expected duration of choosing an arbitrary terminal menu item *t in* hierarchy *h*. This concept can be represented in terms of Expected cost (equation 2) and has been used in paper by St. Amant, Horton& Ritter, (2004; 2006) satisfactorily to produce a very straightforward performance measure that represents the cost of traversing a menu hierarchy, over different possible user profiles, for all menu items that are accessed.

Figure 10: Comparative chart of Model Predictions for Hierarchy 2

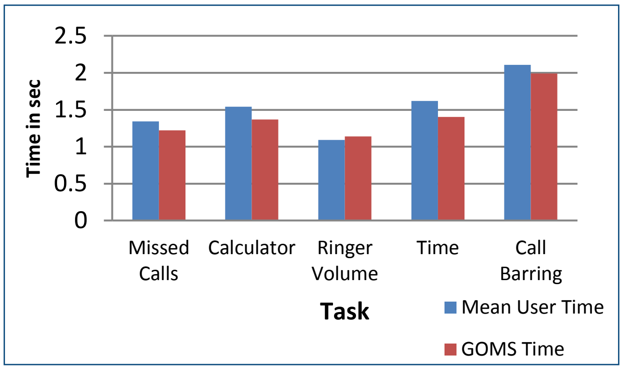


Figure 11: Comparative chart of Model Predictions for Hierarchy 3

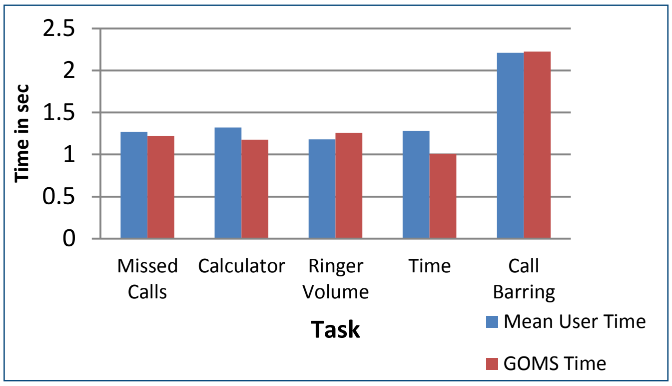


Table 9: Assumed user profile, with Probability of tasks

Task	*Missed Calls*	*Calculator*	*Ringer Volume*	*Time*	*Call Barring*
Probabil-ity	0.35	0.15	0.14	0.34	0.02

Expected cost (*EC*) of making an item selection on a menu hierarchy *h is* defined as:

$$EC(h) = \sum_{t \in Th} p(t)c_h(t) \qquad (2)$$

where, gives the probability of the occurrence of a particular terminal *t*, is the set of terminals in hierarchy *h* and gives an estimate of the cost of reaching terminal *t* in the hierarchy *h*. , can be directly estimated through usage statistics across user profiles. Maintaining a log of activities while users interact with cell phone may provide enough data to construct a probability distribution of the occurrence of terminals in . may be found out using GOMS predictions.

Considering a single user profile with an estimated cost of reaching menu items in terms of predicted GOMS time for three hierarchies (Tables 5, 6 & 7) and estimated probability of each task (Table 9) with respect to the chosen five tasks, the expected cost for each of three hierarchies is computed. Estimations costs calculated (using equation 2) for hierarchy 1, hierarchy 2 and hierarchy 3 were found to be 1.36, 1.31 and 1.15, respectively. Based on the assumed user profile of this study, hierarchy 3 was found to be the most efficient. It performed better than hierarchy 1 by a factor of 0.1548 or 15.48% and by a factor of 0.1183 or 11.83% over hierarchy 2. Such a procedure, making use of user models may be used by developers in the design and analysis of hierarchical menus, thus saving the need to recruit volunteers each time to measure the cost of a particular test hierarchy.

5. Discussion

5.1. Conclusion and Future scope

It has been shown that GOMS is a deserving candidate for measuring the efficiency of tasks that are performed repeatedly and in particular for comparing alternative designs before they are ready to show to users. Tests conducted by expert users, i.e. familiarity with the system, who tend to carry out actions faster and chain those automatically, without too much thinking support that GOMS is able to give good predictions. GOMS can be used on prototypes, wireframes, etc. as long as you know the tasks and steps that users will follow.

It is clear from the results obtained from comparing GOMS time with respect to actual mean user time for all the menu hierarchies (Figure2, 4, 5 and6) and across all five tasks that average task prediction with GOMS was lagging for all the studied hierarchies with an average difference of 6.53%. This can be attributed to the fact that testing and modeling are not alternative methods for evaluating usability. They tap different aspects of usability – initial vs. long term. Modeling is a method that gets at a unique issue and it is not an alternative to testing. GOMS predicts long term usage, whereas testing predicts initial, thus taking longer time. With practice user's time is expected to get closer to GOMS predictions. Further, the data suggest scope for improvements in the models used.

A fundamental problem with a menu interface lies in the restriction of a tree structure. The approaches provided for design of an effective menu hierarchy seem to be simple and logical; the most commonly used tasks should be put earlier and higher in the menu hierarchy. Situations when frequencies of use of tasks are not known or it varies widely among users; allowing the system to rearrange its menu system based on user's usage-patterns on demand may be a sensible method.

In this study Fitts' Law was applied in selecting menu items on touch screen, but there are tasks where it is required to drag item across the screen to fulfill a task (e.g. dragging an app or icon to a trash bin), it is yet to be seen whether Fitts' can be used for such tasks. Another important issue with Fitts' is its applicability to touch screen, if we were just going to look at directly selecting an item on a screen from discontinuous perspective taking into consideration the 3D movements, where multiple factors come into the picture such as relative distance to the target, the angle of the target with regards to the finger, the size of the target and relative positioning of the target with regards to the focus of the user.

From a practical standpoint, users do make mistakes in interacting and so, assuming an ideal error-free scenario is unrealistic. Exploring the possibility of incorporating error models in GOMS, finding out effects of learning, and performance under stress are certainly of research interests.

Since simulation was done on a desktop monitor, it would be interesting to study relative performances with respect to real hand held interfaces. Further, it's expected that users are themselves mobile while they interact, but the simulations are in sitting position. Orientation (vertical or horizontal) and how the device is held is an import parameter for mobile interface design for optimum UX.

The work described in this paper is a small initial stepof a broader project in automated interface evaluation for touch screen phones. More advanced Cognitive modeling architectures (e.g., ACT-R, Soar, and EPIC) are good candidates to produce reliable predictions about user performance, prior to evaluation with real users. The application of these models to specific problems, however, requires significant expertise in modeling, task analysis, and interface design. Hence, long term goal is to build what can be called a Cognitive Model Interface Evaluation tool or system for Touch Screen cell phone that support the display of the cell phone interfaces, experimental control over the cognitive model and its simulation runs, feedback on model execution, model execution diagnostics, and simple display facilities for model traces.

6. Tips for Usability Practitioners

Tips for Fitts' analysis:

- Use Fitts' as a starting point for calculating movement times; this provides valuable information that can directly help in evaluating UI with advanced modeling techniques.
- We tend to use thumbs more often while holding and interacting on mobile phones with a single hand; in this case we have to consider that the original Fitts' Law applies only within the range of motion of our thumb. So, for phones with larger screens it requires an additional effort that Fitts' Law does not account to increase movement time.

Use a GOMS analysis when you want to:

- Evaluate the efficiency of a task that is performed repeatedly over time.
- Compare the efficiency of two paths to success for the same task.
- Compare two different designs to accomplish the same goal or task.

Development of GOMS modeling techniques involves validating the analysis against empirical data. However, once the technique has been validated and the relevant parameters estimated, no empirical data collection or validation should be needed to apply a GOMS analysis during practical system design, enabling usability evaluators to obtain much faster than user testing techniques. However, the calculations required to derive the predictions may seem tedious and mechanical. The models do not cover all of the design issues; the final user test can reveal other problems, and also modeling mistakes. Comparing final test data to the model might be useful.

References

Bay, S.,&Ziefle, M., (2004).*Effects of Menu Foresight on Information Access in Small Screen Devices*.48th Annual meeting of the Human Factors and Ergonomic Society (pp. 1841-1845).

Card, S.K.,Moran, T.P.,& Newell, A. (1983). The Psychology of Human-Computer Interaction. London: Lawrence Erlbaum Associates.

Chae, M.,& Kim, J. (2004). Do size and structure matter to mobile users? An empirical study of the effects of screen size, information structure, and task complexity on user activities with standard web phones. *Behavior and Information Technology*, 23(3), 165-181.

Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6), 381-391.(Reprinted in *Journal of ExperimentalPsychology: General*, 121(3), 262-269.

Gray, W.D.,John, B.E., &WOOD, M.E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real world task performance. Human-Computer Interaction, 8(3), 237-309.

John, B. (2003). Information processing and skilled behavior. *HCI models, theories, and frameworks,* Carroll, J. (ed), SF, CA: Morgan Kaufman.

John, B.E. &Kieras, D.E. (1996a). Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Computer-Human Interaction*, 3(4), 287-319.

John, B. E., &Kieras, D. E. (1996b). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4), 320-351.

Johnson, P. (1998). *Usability and Mobility: Interactions on the Move*. In proceedings of the first workshop on Human-Computer Interaction with mobile devices. Glasgow, Scotland, GIST Technical Report G98-1.

Kieras, D. (1999)a. *A guide to GOMS model usability evaluation using GOMS L and GLEAN 3*. University of Michigan, Ann Arbor, MI.

Kjeldskov, J., & Graham, C. (2003).*A Review of Mobile HCI Research Methods*. In Proceedings of the 5thInternational Mobile HCI 2003 Conference, Udine, Italy, Sringer-Verlag.

MacKenzie, I.S., Sellen, A., & Buxton, W. (1991).*A Comparison of Input Devices in ElementalPointing and Dragging Tasks*. In Proceedings of the CHI'91 Conference on Human Factors in Computing Systems (pp. 161-166).

MacKenzie, I. S., & Buxton, W. (1992).*Extending Fitts' Law to Two-dimensional Tasks*. In Proceedings ofthe CHI'92 Conference on Human Factors in Computing Systems (pp. 219-226).

MacKenzie, I.S. (1995).*Movement Time Prediction in Human Computer Interfaces* (2nded).In Readings in human computer Interaction (pp. 483-493).

MacKenzie, I. S. (2003). *Motor behaviour models for human-computerinteraction*. HCI models, theories, and frameworks, Carroll, J. (ed), SF, CA: Morgan Kaufman.

MacKenzie, I. S. (1991).Fitts Law as a Performance Model in Human-Computer Interaction. Retrieved from http://www.yorku.ca/mack/phd.html(accessed on December 24, 2012).

Sears, A., &Shneiderman, B. (1991). High precision-touchscreens: Design strategies and comparisons with a Mouse. *International Journal of Man-Machine Studies*, 34(4), 593-613.

St. Amant, R.,Horton, T., & Ritter, F. E. (2004).*Model-based Evaluation of Cell Phone Menu Interaction*. In Proceedings of the ACM Conference on Human Factors in Computing Systems (pp. 343-350).

St. Amant, R. Horton, T. & Ritter, F. E (2006), Model-basedevaluation of expert cell phone menu interaction. *ACM Transactions on Computer-Human Interaction*, 14(1), 1-24.

Tamminen, S., Oulasvirta, A., Toiskallio, K., &Kankainen, A. (2004). Understanding mobile contexts.*Journal of Personal and Ubiquitous Computing*, 8(2), 135-143.

Yukikazu, N. &Mitsuko, S. (2006). *Design of aHyper-link-based Software Architecture for Smart Devices*. In ISORC, IEEE Computer Society (pp. 261-268).

Ziefle, M. (2002). The influence of user expertise and phone complexity on performance, ease of use and learnability of different mobile phones. *Behavior and Information Technology*, 21(5), 303-311.

Ziefle, M.,& Bay, S. (2006). How to overcome disorientation in mobile phone menus: A comparison of two differenttypes of navigation aids. *Human-Computer Interaction*, 21(4), 393-433.

Ziefle, M., Arning, K., & Bay, S. (2006).*Cross-platform Consistency and Cognitive Compatibility: The Importance of Users' Mental Model for the Interaction with Mobile Devices*.In Proceedings of CHI 2006 Workshop The Many Faces of Consistency in Cross-Platform Design.