What Does the User do: A Tool for Visualising the Novice User's Interaction Relative to Optimum Path

Mikael Goldstein, Jost Werdenhoff and Thomas Backström[1]
Ericsson Radio Systems AB
Applications Research, Usability Lab
Torshamnsgatan 23
164 80 Kista, Sweden
mikael.goldstein, jost.werdenhoff {@era.ericsson.se}
{+46 8} 757 3679, 404 4644
[1]Ericsson Business Network AB
131 89 Stockholm, Sweden
thomas.backstrom@ebc.ericsson.se
(+46 8) 422 2348

ABSTRACT

The generic KeystrokeMapper data visualisation tool proposes a new way of visualising and annotating novice user behaviour when interacting with an interface as expressed in the unit keystroke actions. By displaying the Optimum path keystroke actions when accomplishing a task as a straight diagonal string of characters, each character representing a keystroke action, and plotting the novice user's keystroke action path in the same plot, deviating (keystroke) actions are displayed visually. The KeystrokeMapper tool thus displays a novice user's navigation path relative to the Optimum path for a particular task. Each task plot generates a topological user keystroke action map suitable for qualitative analysis. The KeystrokeMapper tool only exists as a paper-based mock up. It has not yet been evaluated and does not exist as software.

Keywords

Usability, Optimum path, KLM, GOMS, keystroke, data visualisation, novice user, expert user

1. INTRODUCTION

Usability has recently become a buzzword in the new IT age. With the miniaturisation of hardware and the increasing complexity of software, the usability of the interface itself is becoming the new challenge. Both for the designer as well as for the novice user. We specially refer here to off the shelf products containing a GUI (Graphical User Interface). The novice user is usually on his own when trying to interact for the first time with a product (e.g., a palmsized handheld computer device). Since the user's mental model and the designer's conceptual model (Norman 1983) usually differ, the keystroke action path that the user will pursue will probably deviate from the designer's action path, usually referred to as the Optimum path (Mohageg 1992).

1.1 Optimum path

Optimum path is the path a *skilled,* experienced user (or the designer) of an interface would use to accomplish a predefined task. The Optimum path is thus the shortest route to accomplish a task in a given software or system. The total number of keystrokes by the novice user in relation to the Optimum path can be expressed as a deviation metric (D). Mohageg (1992) proposes a deviation from the Optimum path (Equation 1) metric. The total number of keystrokes by the novice user to accomplish a task is divided by number of relevant keystrokes=Optimum path:

$$D = \frac{\# of\ keystrokes}{\# of\ relevant keystrokes} \quad Eq.1$$

A low keystroke deviation ratio (the minimum value is 1.0 if the user chooses Optimum path) indicates that few irrelevant keystroke actions were used. Often it is possible to accomplish the same task by engaging a different keystroke route than the Optimum path. One could also express the difference by using the term's conceptual and mental model (Norman 1983). Whereas the conceptual model could be regarded as the designer's way of traversing the interface (Optimum path), the novice user's mental model signifies the user's path through the interface in order to accomplish the same task. Often the novice user's path does not coincide with the designer's conceptual model and this deviation may be operationally defined by using Mohageg's (1992) keystroke deviation metric.

The Novice Expert Ratio Method (NEM) proposed by Urokohara et al. (2000) is based on the ratio of completion time to perform each successive keystroke by a Novice and an Expert user. For each successive keystroke, a different Novice/Expert user completion time ratio is computed. Here each successive keystroke is substituted for the completion time necessary for the novice/expert user to accomplish it. When a certain Novice/Expert duration ratio (duration to accomplish a certain keystroke) is high (e.g., 6-12), this may be regarded as a usability problem. A similar usability metric based on total task completion time has been proposed (Book and Goldstein 1999). For seven tasks related to adding voice control to a familiar cellular phone interface, the task Efficiency criterion was set to 3 x total Optimum path time for novice users. Whereas the keystroke time usability metric proposed by Urokohara et al. (2000) detects where usability problems occur at the keystroke level, it does not provide the experimenter with a qualitative analysis of the reason. The NEM method does not log deviating keystroke actions. Baumeister, John and Byrne (2000) compared three tools, QGOMS, CATCHI and GLEAN for building GOMS (Goals, Operators, Methods and Selection rules) models. They found the "Quick and Dirty GOMS" (QGOMS) suitable to provide a nice graphical representation to visualise the goal hierarchy and they "hope that some 2-dimensional representation would be available in future GOMS tools" (Baumeister et al. 2000, p. 508).

1.2 Usability Evaluation Tools

Iterative design has been the hallmark in designing complex products in order to improve their usability by making the conceptual and mental model coincide (Norman 1983). The interface is exposed to different users in a usability lab and the whole interaction process is videotaped.

Beside the usability evaluation of different user interfaces that are brought into the usability lab, the usability of various tools and methods employed when making a usability evaluation has also been in focus. During the last 15 years, there has been considerable work on improving the effectiveness of usability evaluation methods (Hudson, John, Knudsen and Byrne 1999). There are several methods that can be used for usability evaluation. Empirical techniques such as keystroke logging, think-aloud usability tests, heuristic evaluation, expert review and user models such as GOMS.

1.2.1 GOMS

GOMS analysis has been used as a *predictive* modelling technique in human-computer interaction (Card Moran and Newell 1983, Hudson et al. 1999). GOMS describes the *skilled* (experienced) user's knowledge of how to perform a task in terms of Goals, Operators, Methods and Selection rules. A GOMS analysis can produce quantitative and qualitative predictions of how *skilled* people will use a system. It takes into consideration the perception, cognition and motor response times. Goals are simply the user's goals. The primary goal of a task could be decomposed into a hierarchical tree of subtasks that have to be accomplished in order to complete the task (Eberts 1994). For GUIs, operators are menu selections, button presses or direct-manipulation actions. Methods are well-learned sequences of subtasks and operators that accomplish a task. A classic example is deleting a paragraph in a text editor (Hudson et al. 1999). Selection rules are the personal rules that users follow in deciding what method to us in a particular context. GOMS has often been viewed as extremely time- and labour-intensive (Hudson et al. 1999).

1.2.2 The Key-Level Model

A simplified method is called the Key-Level Model (KLM) (Card et al. 1983). Whereas the GOMS model accounts for different types of unobservable events such as visual search and retrieval from long-term memory, the KLM *only* uses *keystroke-level* operators. The KLM analysis simply lists the keystrokes, mouse-movements and mouse-button presses that the user must perform to accomplish a task. One could thus argue that Optimum path and KLM analysis could be regarded as two sides of the same coin. The KLM is useful for quantitatively predicting task execution time. Since most KLM is done by hand, which takes a lot of time, an automated tool named CRITIQUE (Hudson et al. 1999) has been proposed. The CRITIQUE tool automates a number of manual evaluation tasks. It automatically generates predictive performance models from demonstrations of tasks. However, as the authors point out, the benefits of automation is to provide support for creating higher-level goal hierarchy.

1.3 Lack of Visualisation in Current Software Tools

Traditional usability evaluation tools available on the market have not, to our knowledge, the facility to provide the usability evaluator with visualisation of the novice user's interaction with an interface. The Noldus Observer Pro and the UsabilityWare (Noldus 2000) software tools only prompts the experimenter to classify and tag deviating events or states. It is implicitly understood that only the keystroke actions that deviate from the correct keystroke actions are of interest to classify. When the user pursues the Optimum path, this is not displayed in any way. Furthermore, the path analysis is only displayed on the text level and not visualised. Would it be possible to visualise the user's action path when performing a predefined task relative the Optimum path? By transforming the text-classified keystroke events into a plot, it may be easier to understand what the user is doing. Also the fact that the user's path is displayed in relation to the intended path (Optimum path) may ease understanding.

2. THE KEYSTROKEMAPPER TOOL

The KeystrokeMapper data visualisation tool works at the keystroke level. It is a generic tool, based on the simultaneous visualisation of the Optimum path to accomplish a certain task as well as the novice user's navigation path to accomplish the same task. In this way, the novice user's keystroke actions can be compared to the intended Optimum path keystroke actions performed by an experienced user for a particular task, for each successive keystroke. A task could be regarded to be made up of several sub-tasks. In order to accomplish each sub-task in an efficient and effective way, the Optimum path route is taken. An analogy to fitting a regression line to a scatterplot is also relevant. However, in this case, the regression line is drawn first (Optimum path). Then the user data is plotted and the fit relative to Optimum path is analysed.

2.1 Proposed Annotations

2.2 Optimum path user

All the expert and novice user's (keystroke) actions to accomplish a predefined task are logged. For each task, which can be looked upon as a series of sub-tasks, the Optimum path keystroke action route is depicted as a straight diagonal string of unfilled circles (ooo) in the plot. A keystroke action could be any kind of behaviour. A stylus tap, a key depression, as well as entering information in a field or asking for help. For the sake of simplicity, a double-tap is classified as a single keystroke event. Entering textual information into a field is also classified as a single keystroke action. Successive taps on the scroll bar are also treated as a single keystroke.

2.3 The novice user's path

The novice user's keystroke path is also depicted as a straight diagonal string of characters in the plot. Various keystroke annotations are used to differentiate between successful and unsuccessful strategies. If the novice user's keystroke actions mimic the Optimum path user actions (relevant keystrokes), they will appear in the Optimum path string as filled circles (•••). If the novice user's keystroke actions do not mimic the Optimum path actions, they are displayed as a straight diagonal string of plus signs (+++) parallel to the Optimum path (see Figures 2 and 3). As mentioned earlier, there are usually several different alternative paths that a user can pursue in order to accomplish a task successfully. They might not be as effective as the Optimum path, which may include "short-cuts" but in spite of this, the task is accomplished. If a user accomplishes various subtasks that leads to the successful accomplishment of the task, but does not use the Optimum path, this is annotated as an *alternative* (aaa) path.

System-generated (error) messages are classified as a single keystroke action as well. If the user enters information in a field that the system does not approve of, it will generate an error message in the form of a displayed window with an OK icon. When the system generates a (error) message, this is annotated with the letter "s".

When there is a switch or deviation between the Optimum path and the novice user path this is depicted with an arrow.

When a skilled expert user accomplishes a task (the goal is to accomplish the task using the shortest route (Optimum path)), he will choose a path that requires the minimum amount of keystrokes. Thus, to accomplish a task correctly using the Optimum path requires the least amount of keystrokes. That is to say only relevant keystrokes. When the last novice user keystroke sequence is accomplished, the sequence ends with a "p" (for Pass) or an "f" (for Fail), if the novice user has requested help from the experimenter.

In order to compute a deviation metric (D), the number of novice user keystrokes is divided by the number of relevant keystrokes=Optimum path (Equation 2). The number of keystrokes performed by a novice user (Equation 3) is the sum of all the alternative keystrokes (a), those keystroke actions that coincide with the Optimum path user actions (•) and the number of additional irrelevant keystrokes, including help (+). The number of system generated error messages (s) are added as well (if any).

$$\#\ of\ relevant\ keystrokes = (o) + (\bullet) \qquad Eq.\ 2$$

$$\#\ of\ keystrokes = (a) + (\bullet) + (+) + (s) \quad Eq.\ 3$$

3. THE EXPERIMENT

The KeystrokeMapper tool was used to visualise one single task that was given to two (out of five) novice users when interacting with a palmsize computer named Ericsson MC16 (Figure 1) in a previous study. Two of the novice user accomplishments will be analysed using the above-described generic annotation procedure.

3.1 The Device

The MC16 runs on Windows CE and features a black-and-white touch-sensitive screen (Figure 1). A stylus is used instead of a mouse to interact with the screen. In order to enter characters, it features a miniaturised hard QWERTY keyboard. To move between input fields *within* a window, it is possible to use either the stylus *or* the black the Tab key located to the left on the QWERTY keyboard on the third row.

Figure 1. The Ericsson MC16 palmtop computer features a miniaturised hard QWERTY keyboard and a touch-sensitive screen. The stylus is pointing at the Calendar application.

3.2 Users

Two out of the five novice users interacting with the device when solving the task "Make an appointment" were analysed on the keystroke level (see Figures 2 and 3). The novice users were originally exposed to seven different tasks that were presented in consecutive order. The users had no prior familiarity with using a palmsize handheld computer but were regular PC users and were familiar with using Windows. None of them had experience from using Windows CE.

3.3 Procedure

The users were allowed to familiarise with the device for 10 minutes before being exposed to the seven tasks. No manual was available during the whole test. They were given the time necessary to solve each task at their own pace. If the task was accomplished successfully, this was classified as Pass (p). If they encountered a problem, they could ask for Help. If they succeeded in

Keystroke Number / Keystroke Action	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Optimum path (o)= 25 keystrokes																										
Dbletap Calendar	•																									
Sel Day view icon		o																								
Scroll Month			o																							
Sel W 2 in Month				o																						
Sel Week 2 view					o																					
Scroll up/down						o																				
Double-tap Thursd							o																			
Enter App descript																										
Sel Location field									•																	
Enter location										•																
S Starttime dpdlist											•															
Enter hh Start time												•														
Enter : delimiter													•													
Enter mm Start tim														•												
S Endtime dpdlist															•											
Scroll down																•										
Enter hh End time																	•									
Enter : delimiter																		•								
E mm End time																			•							
Sel Remark tic box																				•						
Sel Bell icon																					•					
Sel Remind opt.																						•				
Scroll down																							•			
Select OK icon																								•		
Sel OK icon x 2																									•	
User HS																										
Select Week icon		+																								
Sel Go menu			+																							
Sel Week menu				+																						
Sel Forward icon					+																					
Sel Edit menu						+																				
Sel View menu age							+																			
Sel Week icon								+																		
Sel Filter menu									+																	
Close Filter menu										+																
Sel Week icon											+															
Sel Go menu												+														
S GoTo Date opt													+													
Sel dropdown list														+												
S Scroll icon in mo															+											
S 11th in week 2																+										
S OK in GoTo date																	+									
Sel File menu																		a								
S New Appointm																			a							
Read Error msg																				s						
Sel X icon																					+					
Sel OK icon																						+				
Effectiveness																							p			

Figure 2. Visualised Optimum path and novice user path for user HS at the keystroke level for accomplishing Task 6: Make an appointment in the Calendar. (o). Optimum path (relevant keystrokes) to accomplish the task (25 relevant keystrokes). (+). The user's deviating keystroke actions. (•). The user's keystroke action coincides with the Optimum path keystrokes. (a). An alternative path to Optimum path is used. (s). System error message. (p). Pass.

Keystroke Number Keystroke Action	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Optimum path (o)= 25 keystrokes																										
Dbletap Calendar	•																									
Sel Day view icon		o																								
Scroll Month			o																							
Sel W 2 in Month				o																						
Sel Week 2 view					o																					
Scroll up/down						o																				
Double-tap Thursd							o																			
Enter App descr								o																		
Sel Location field																										
Enter Location										•																
S Starttime drpdlist											•															
Enter hh Start time												•														
Enter : delimiter													o													
E mm Start time														•												
S Endtime drpdlist															•											
Scroll down																•										
Enter hh End time																	•									
Enter : delimiter																		o								
E mm End time																			•							
Sel Remark tic box																				•						
Sel Bell icon																					o					
Sel Remind opt.																						o				
Scroll down																							o			
Select OK icon																								o		
Select OK icon x 2																									o	
User KV																										
Sel Fw icon M wie		a																								
Sel 11th week 2			a																							
Sel 12th				a																						
Sel 13th					a																					
Sel 14th						a																				
Sel 15th							a																			
Sel 14th								a																		
Sel File menu									a																	
Sel New Appointm										a																
Enter . delimiter											+															
Enter . delimiter												+														
Error message													s													
Sel OK icon														+												
Help															+											
Effectiveness																f										

Figure 3. Visualised Optimum path and novice user path for user KV at the keystroke level for accomplishing Task 6: Make an appointment in the Calendar. (o). Optimum path (relevant keystrokes) to accomplish the task (25 relevant keystrokes). (+). The user's deviating keystroke actions. (•). The user's keystroke action coincides with the Optimum path keystrokes. (a). An alternative path to Optimum path is used. (s). System error message. (f). Fail.

accomplishing the task after the experimenter offered help, this was classified as Fail (f). Seven tasks were given to each user in consecutive order. The task “Make an appointment” was given as task number six. Thus, the users had already by then had around 30 minutes of practice with the novel interface.

3.4 Task

Task six; “Make an appointment” reads as follows:

The wireline phone rings. Answer the call. Make an appointment in the Calendar. Go to week 2 in year 1999 (the evaluation took place in November-December 1998). *See where there is time available to book a meeting about mobile devices that starts at quarter past seven and ends at quarter past twelve the same day. Set the alarm as a reminder 10 minutes before the meeting starts.*

3.4.1 Optimum path for task six

This task is broken down into a series of subtasks. The Optimum path to accomplish the task is displayed in Figures 2 and 3 as the least amount of keystrokes (stylus taps/Tabs) necessary to accomplish it. To accomplish task six successfully using Optimum path, 25 different keystroke actions are necessary. When text information is entered in a field this is annotated as one single keystroke action. When time information is entered, this is regarded as three separate keystroke actions: Enter hours (hh Starts/Ends), enter delimiter (:) and enter minutes (mm Starts/Ends). Double-taps which appear in the application picker menu (to select the Calendar application) and when opening up the New Appointment window when double-tapping on the day area, is classified as one keystroke action as well.

To pursue the Optimum path, the user first double-taps on the Calendar icon in the application picker (see Figure 1), then proceeds to the “Week view” of week 2. Week 2 is prepared in advance by the experimenter. Some days are already booked. In order to get an *overview,* it is important to get the Week view. Since the only time available is on Thursday morning, the user double-taps on the 07:00-08:00 area. There is at least one alternative to Optimum path here. It is also possible to choose the menu item “New Appointment” in the File menu (Figure 4). In both cases, the (New) Appointment window opens (Figure 5). Here the user enters the description of the meeting (meeting with Jost, the experimenter) in the Appointment Description field and the location of the meeting (room Ros) is entered in the Location field below. Start and end times are entered in the Starts and Ends time fields. It is also important that the Reminder box is ticked and the Bell icon is tapped on in order to set the reminder to 10 minutes in the Reminder Defaults window. This altogether makes 25 different “keystroke” actions. In both Figures 2 and 3, the Optimum path to accomplish task six is displayed as a diagonal string of unfilled circles (oooooo).

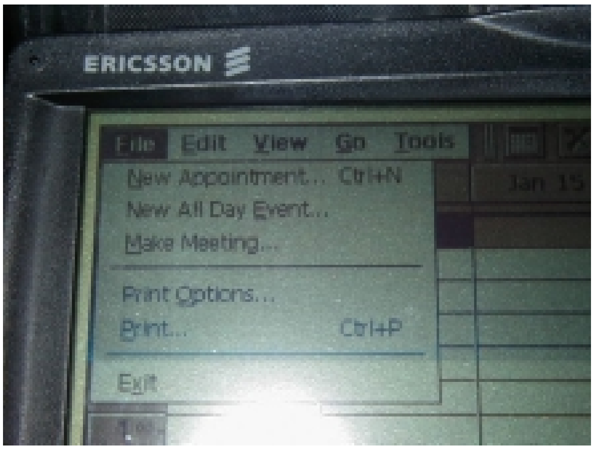

Figure 4. The item New Appointment in the File menu is selected to create a new appointment.

Figure 5. The (New) Appointment window as it is presented to the user when opened.

4. RESULTS: VISUALISING THE USER PATH

User HS is visualised in Figure 2 and user KN is visualised in Figure 3 using the proposed annotations. The Optimum path for Task 6 is depicted on top in both figures (o). When the user's path coincides with the Optimum path, the circles are filled (•).

4.1 Analysing Visualised Keystroke Actions

4.2 User HS

User HS (Figure 2) initially did employed neither the Optimum path, nor an alternative path (+). He had problems understanding how to create a New Appointment. After 17 keystrokes, he was finally on the right track when he found the New Appointment option under the File menu (a). This is an alternative path to double clicking at the 07:00 field of Thursday in week 2. He did not enter text information in the Appointment Description field (see Figure 5) on top and entered text information in the Location field first. When trying to close the New Appointment window, he received a system error message (Read error message (annotated as "s") in Figure 2). The text read:

"You have not entered a description

for this item. Is this OK?"

The system-generated (error) message prompted him to enter information in the empty Appointment Description field, which he did. The usability problem he encountered was how to create a new appointment. He accomplished the task successfully and used 18 additional keystroke actions (+), 19 relevant (•), two alternative (a) and read one error message (s), which amounts to 40 keystroke actions altogether. Mohageg's deviation metric thus yields a value of 1.60 (D=40/25).

4.3 User KV

User KV (Figure 3) did not employ the Optimum path in order to get the overview of week 2. Instead she used the Day view and selected each of the different days in that week, one at a time, to see if they were free. Thus, her keystroke actions were annotated as an alternative (a) to Optimum path and were thus not counted as deviations from Optimum path. She succeeded in finding the New Appointment window in the File menu (see Figures 4 and 5). This was an alternative (a) to double tapping at the 07:00 field of Thursday in the week 2 view. She did not start input at he top field of the New Appointment window. She went straight to the Starts time dropdown list. She then entered the start time of the meeting in the Starts time field as "07.15" (instead of 07:15). In the Ends time field, she entered the time as "12.15" (instead of 12:15). A little later, the system displayed a Calendar system error message (s) that read (Figure 6):

"Enter a valid time"

She tapped on the OK icon in the Calendar window and checked the Starts time. However, she could not detect the error (. instead of :). The Calendar system error message (s) was displayed 8 times in succession. This loop is not visualised adequately in the plot. Finally, unable to spot the "error" she turned to the experimenter for help. If she would have done all the other remaining keystroke actions correctly, the predicted (*) number of keystrokes would have amounted to a total of 43: 16 relevant (•), 11 irrelevant, including Help (+), 9 alternative (a) and 8 system error messages (s). This yields a predicted deviation metric of 1.76* (D=43/25). However, since she asked for help, this task was classified as a Fail (f).

Figure 6. Calendar window appearing when a "user" error has occurred. In this case the user KV had entered a punctuation mark as a delimiter (.) instead of a colon (:) in both the Starts time and Ends time field.

5. DISCUSSION

The KeystrokeMapper data visualisation tool is presently only a paper-and-pencil mock-up. It has not yet been evaluated properly. However, we feel that a software where the keystroke level input is visualised in this way would make the interpretation of behavioural data easier to analyse. The KeystrokeMapper data visualisation tool also features certain drawbacks. One is scalability. For the visualised task, only up to around 25 relevant keystroke actions are possible to visualise in one plot. If the task is more complex, perhaps up to 100 relevant keystrokes have to be displayed. Sooner or later the width of the paper is filled up. The annotation of repetitive (error) behaviour has not yet been properly addressed.

We deliberately have not accounted for the additional metrics that can be integrated in the future software like task completion time, time elapsed between successive keystrokes and an automatically computed deviation from Optimum path metric (D). It is assumed that either during the video recording or afterwards, during playback, it would be possible to tag the users keystroke actions when accomplishing the task. Whenever a keystroke action occurs, the experimenter tags and classifies the event. A time-stamp is then generated automatically. For each time-stamped event, the experimenter enters the textual description of the action. Then, when both the Optimum keystroke path and the different user's keystroke path is classified, the KeystrokeMapper generates the figures (see Figures 2 and 3). Another disadvantage not yet properly addressed is the difficulty to aggregate data across users in one plot. Each user´s path is presently plotted separately for each task. Thus, the tool could be regarded as enhancing the qualitative, rather than the quantitative analysis.

One possible way to compare several users at the same time is to plot limited common consecutive keystroke actions, e.g. starting from one specific position and visualise only the next few keystrokes. For example, both users had problems finding out how to create a New Appointment and used the same alternative keystroke path. This would give an indication of common problems related to the user's mental model of the interface.

Finally, we also anticipate difficulties using the KeystrokeMapper in real time during a usability test. It will probably be hard for the experimenter to keep up with the pace of the user and at the same time write down the user's path, indicate if it is an alternative path (a), a deviating keystroke (+) or following the Optimum path. The different ways of annotating the novice user's path through the interface might therefore be more suitable to perform after the usability test, and only use one symbol (e.g. x) during the actual test.

6. CONCLUSIONS

We propose a relatively simple and rapid generic data visualisation tool, the KeystrokeMapper, which can be used in the usability lab. The tool aims at giving a visual map of the novice user's interaction behaviour relative to the designer's intended interaction path (Optimum path). Any deviations from the straight and narrow diagonal Optimum keystroke path are visualised and can be analysed qualitatively. In order to validate the proposed annotation vocabulary, it is important to apply the KeystrokeMapper generic tool on a multitude of different tasks accomplished by different users. We are open for new ideas on how to aggregate the results from different user performance on the same task into one single plot. For the moment, each task is visualised on the individual level only.

7. ACKNOWLEDGEMENTS

We thank Mr. Marcus Nyberg at Ericsson Research, Kista, for valuable comments.

8. REFERENCES

Baumeister, L.K., John, B.E. and Byrne, M.D. (2000). A comparison of tools for building GOMS models. In *Proceedings of CHI'2000,* ACM Press, pp. 502-509.

Book, R. and Goldstein, M. (1999). Mental effort increases when adding a voice control feature to a familiar interface. In *Proceedings of the HCI'99 International Conference* by H-J. Bullinger and J. Ziegler (Eds.), Munich, Germany, 22-27 August 1999, Lawrence Erlbaum Associates, pp. 23-27.

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

Eberts, R.E. (1994). *User interface Design*. Prentice-Hall, Englewood Cliffs, NJ.

Hudson, S.E., John, B.E., Knudsen, K. and Byrne, M.D. (1999). A Tool for Creating Predictive Performance Models from User Interface Demonstrations. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology,* UIST'99, Asheville, North Carolina November 7-10, 1999*, CHI Letters*, **1,** 93-102.

Mohageg, F.M. (1992). The influence of hypertext linking structures on the efficiency of information retrieval. *Human Factors,* **34** (3), 351-367.

Noldus Video-Pro product sheet. Available at: http:// www.noldus.com/products/index.html (April 2000).

Norman, D.A. (1983). Some observations on mental models. In D. Gentner and A.L. Stevens (Eds.). *Mental models* (pp. 7-14). Hillsdale, NJ: Erlbaum.

Urokohara, H., Tanaka, K., Furuta, K., Honda, M. and Kurosu, M. (2000). NEM: "Novice Expert ratio Method" A usability evaluation method to generate a new performance measure. *In Proceedings of CHI'2000, Extended Abstracts,* ACM Press, pp. 185-186.

