KogWis 2014 Tübingen manuscript No.
(will be inserted by the editor)

An activation-based model of execution delays of specific task steps

Marc Halbrügge · Klaus-Peter Engelbrecht

Halbrügge, M. and Engelbrecht, K.-P. (2014). An activation-based model of execution delays of specific task steps. *Cognitive Processing*, 15, S107-S110.

Received: date / Accepted: date

Abstract When humans use devices like ticket vending machines, their actions can be categorized into task-oriented (e.g. selecting a ticket) and device-oriented (e.g. removing the bank card after having paid). Device-oriented steps contribute only indirectly to the user's goal; they take longer than their task-oriented counterparts and are more likely to be forgotten. A promising explanation is provided by the activation-based memory for goals model (Altmann and Trafton 2002). The objectives of this paper are, first, to replicate the step prolongation effect of device-orientation in a kitchen assistance context, and secondly, to investigate whether the activation construct can explain this effect using cognitive modeling. Finally, a necessity and sensitivity analysis provides more insights into the relationship between goal activation and device-orientation effects.

Keywords Cognitive Modeling · Human-Computer Interaction · ACT-R · Memory · Human Error

Introduction and related work

While the research on task completion times in human-computer interaction (HCI) has brought many results of both theoretical and practical nature during the last decades (see John and Kieras 1996, for an overview), the relationship between interface design and user error is still unclear in many parts. Notable exceptions are post-completion errors, when users fail to perform an additional step in a procedure *after* they have already reached their main goal (Byrne and Davis 2006). This concept can be extended to any step that does not directly support the users' goals, independently of the position in the action sequence, and has been termed *device-orientation* in this context (Ament et al 2009). The opposite (i.e. steps that *do* contribute to the goal) is analogously called *task-orientation*. Device-oriented steps take longer and are more prone to omission than task-oriented ones (Ament 2011).

A promising theoretical explanation for the effects of device-orientation is provided by the memory for goals model (MFG; Altmann and Trafton 2002). The main assumption of the MFG is that goals underlie effects that are usually connected to memory traces, namely time-dependent activation and associative priming. Within the theoretical framework of the MFG, post-completion errors and increased execution times for post-completion steps are caused by lack of activation of the respective sub-goal. A computational implementation of the MFG that can be used to predict sequence errors has been created by Trafton et al (2009).

This paper aims at investigating the concept of device-orientation on the background of the MFG using cognitive modeling with ACT-R (Anderson et al 2004). The basic research question is whether human memory constructs as formalized within ACT-R can explain the completion time differences between task- and device-oriented steps found in empirical data.

Marc Halbrügge · Klaus-Peter Engelbrecht
Quality and Usability Lab, Telekom Innovation Laboratories, Technische Universität Berlin, Ernst-Reuter-Platz 7, D-10587 Berlin
E-mail: marc.halbruegge@tu-berlin.de

Experiment

As the empirical basis for our investigation, we decided not to rely on synthetic laboratory tasks like the Tower of Hanoi game, but instead use an application that could be used by everyone in an everyday environment. Our choice fell on a HTML-based kitchen assistant that had been created for research on ambient assisted living. Besides other things, the kitchen assistant allows to search for recipes depending on regional cuisine (French, Italien, German, Chinese) and type of dish (main dish, appetizer, dessert, pastry). Our experiment was built around this search feature.

12 subjects (17% female, M_{age}=28.8, SD_{age}=2.4) were invited into the lab kitchen and performed 34 individual search tasks of varying difficulty in five blocks. The user interface (UI) of the kitchen assistant was presented on a personal computer with integrated touch screen. Task instructions were given verbally and all user clicks were recorded by the computer system.[1] Individual trials consisted of five phases:

1. Listening to and memorizing the instructions for the given trial.
2. Entering the search criteria (e.g. "German" and "Main dish") by clicking on respective buttons on the screen. This could also contain deselecting criteria from previous trials.
3. Initiating the search using a dedicated "Search" button. This also initiated switching to a new screen containing the search results list if this list was not present, yet.
4. Selecting the target recipe (e.g. "Lamb chops") in the search results list.
5. Answering a simple question about the recipe (e.g. "What is the preparation time?") as displayed by the kitchen assistant after having selected the recipe.

We did not analyze the first and last phase as they do not create observable clicks on the touch screen. Of the remaining three phases, entering search criteria and recipe selection are task-oriented, while the intermediate "Search"-click is device-oriented.

Results

We recorded a total of 18 user errors. Four were intrusions, nine were omissions, five were selections of wrong recipes. The application logic of the kitchen assistant inhibits overt errors during the device-oriented step. We therefore focused on completion time as dependent variable and discarded all erroneous trials.

As our focus is on memory effects, we concentrated on steps that task only the memory and motor system. We removed all subtasks that need visual search and encoding (phase 4: searching for the target recipe in the results list and clicking on it), and steps that incorporated substantial computer system response times (i.e. moving to another UI screen).

817 clicks remained for further analysis; 361 (44%) of these were device-oriented. The average time to perform a click was 764 ms (SD=381) for task-oriented and 977 ms (SD=377) for device-oriented steps.

As the kitchen assistant has been created for research in an area different from HCI, it introduces interfering variables that need to be controlled. The motor time needed to perform a click on a target element (i.e. button) depends strongly on the size and distance of the target as formalized in Fitts' law (Fitts 1954). Fitts' index of difficulty (ID) can not be held constant for the different types of clicks, we therefore introduced it into the analysis. As the click speed (i.e. Fitts' law parameters) differs between subjects, we used linear mixed models (NLME; Pinheiro et al 2013) with subject as grouping factor and Fitts' law intercept and slope within subject. We also observed a small, but consistent speed-up during the course of the experiment that led us to the introduction of the trial block as additional interfering variable. The analysis of variance was conducted using R (R Core Team 2014). All three factors yielded significant results, we obtained a prolongation effect for device-oriented steps of 104 ms. The results are summarized in Table 1.

Table 1 Regression coefficients (coef.) with confidence intervals (CI) and analysis of variance results for the experiment. Individual slopes for Fitts' difficulty (ID) ranged from 121 to 210 ms/bit.

Factor name	coef.	95% CI of coef.	$F_{1,802}$	p
Fitts' ID	165 ms	126 to 204 ms	111.1	< .001
trial block	-55 ms	-71 to -39 ms	45.9	< .001
device-orient.	104 ms	53 to 154 ms	16.4	< .001

Discussion

The first objective of this paper is met, we could identify a significant execution time delay for device-oriented steps. How does this effect relate to the existing literature? Ament et al (2009) report an insignificant difference of 181.5 ms between task-oriented and device-oriented steps. This fits well with the empirical averages reported at the beginning of the results section, although the experimental procedure used there (flight simulation game) led to longer steps with completion times well above two seconds.

What remains open is whether the proposed cognitive mechanism behind the time difference, namely lack of activation, can account for this time difference. The next section addresses this question.

The MFG model

We implemented the memory for goals theory based on the mechanism provided by the cognitive architecture ACT-R

[1] The experiment as described here was embedded in a larger usability study. See Quade et al (2014) for more details. The instructions are available for download at http://www.tu-berlin.de/?id=135088

(Anderson et al 2004), as the MFG is originally based on the ACT-R theory (Altmann and Trafton 2002). Within ACT-R, memory decay is implemented based on a numerical activation property belonging to every chunk (i.e. piece of knowledge) in declarative memory. Associative priming is added by a mechanism called *spreading activation*.

This led to the translation of the tasks used in our experiment into chains of goal chunks. Every goal chunk represents one step towards the target state of the current task. One element of the goal chunk ("slot" in ACT-R speak) acts as a pointer to the next action to be taken. After completion of the current step, this pointer is used to retrieve the following goal chunk from declarative memory. The time required for this retrieval depends on the activation of the chunk to be retrieved. If the activation is too low, the retrieval may fail completely, resulting in an overt error.

The cognitive model receives the task instructions through the auditive system, just like the human participants did. For reasons of simplicity, we reduced the information as much as possible. The user instruction "Search for German dishes and select lamb chops" for example translates to the model instruction "German on; search push; lamb-chops on". The model uses this information to create the necessary goal chunks in declarative memory. No structural information about the kitchen assistant is hard coded into the model, only the distinction that some buttons need to be toggled on, while others need to be pushed.

While the model should in principle be able to complete the recipe search tasks of our experiment with the procedural knowledge described above, it actually breaks down due to lack of activation. Using unaltered ACT-R memory parameters, the activation of the goal chunks is too low to be able to reach the target state (i.e. recipe) of a given task. We therefore need to strengthen our goals and spreading activation is the ACT-R mechanism that helps us doing so. How we apply spreading activation in our context is inspired by close observation of one of our subjects who used self-vocalization for memorizing the current task information. The self-vocalization contained only the most relevant parts of the task, which happen to be identical to the task-oriented steps of the procedure. We analogously theorize that the goal states representing task-oriented steps receive more spreading activation than their device-oriented counterparts. This assumption is also in line with the discussion of post-completion errors on the basis of the memory for goals model in Altmann and Trafton (2002).

For the evaluation of the model, we used ACT-CV (Halbrügge 2013) to connect it directly to the HTML-based user interface of the kitchen assistant. In order to be able to study the effect of spreading activation in isolation, we disabled activation noise and manipulated the value of the ACT-R parameter that controls the maximum amount of spreading activation (mas). The higher this parameter, the more additional activation is possible.[2]

Results

We evaluated the overall fit of the model by dividing the clicks into eight groups by the screen areas of the origin and target click position (e.g. from type of dish to search; from search to recipe selection) and compared the average click times per group between our human sample and the model. Besides the traditional goodness of fit measures R^2 and root mean squared error (RMSE), we applied the maximum likely scaled difference (MLSD; Stewart and West 2010) which also takes the uncertainty in the human data into account. The relative difference between the empirical means and the model predictions is given in percent (%diff). The results for five different amounts of activation spreading are given in Table 2.

The model is overall slower than the human participants, resulting in moderately high values for RMSE, MLSD, and relative difference. The explained variance (R^2) on the other hand is very promising and hints at the model capturing the differences between different clicks quite well.

Table 2 Average click time (M_{time}), average memory retrieval time (M_{mem}), determination coefficient (R^2), root mean squared error (RMSE), maximum likely scaled difference (MLSD), and maximum relative difference (%diff) for different amounts of activation spreading (mas).

mas	M_{time}	M_{mem}	R^2	RMSE	MLSD	%diff
2	1785 ms	591 ms	.759	982 ms	16.5	66%
4	1509 ms	315 ms	.738	687 ms	12.1	58%
6	1291 ms	99 ms	.881	477 ms	8.5	50%
8	1231 ms	37 ms	.912	422 ms	7.9	48%
10	1210 ms	15 ms	.893	406 ms	7.8	48%

Sensivity and necessity analysis

In order to test whether our model also displays the device-orientation effect, we conducted a statistical analysis analysis identical to the one used on the human data and compared the resulting regression coefficients. While an acceptable fit of the model is necessary to support the activation spreading hypothesis, it is not *sufficient* to prove it. By manipulating the amount of activation spreading, we can perform a sensitivity and necessity analysis that provides additional insight about the consequences of our theoretical assumptions (Gluck et al 2010). Average coefficients from a total

[2] The ACT-R code of the model is available for download at http://www.tu-berlin.de/?id=135088.

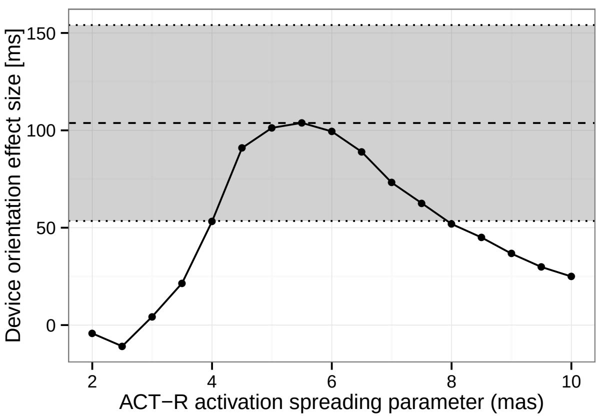

Fig. 1 Device orientation effect size depending on spreading activation amount. The shaded area between the dotted lines demarks the 95% confidence interval of the effect in the human sample.

of 400 model runs are displayed in Figure 1. It shows an inverted U-shaped relationship between spreading activation and the device-orientation effect. For intermediate spreading activation values, the time delay predicted by the model falls within the confidence interval of the empirical coefficient, meaning perfect fit given the uncertainty in the data.

Discussion

The MFG model is able to replicate the effects that we found in our initial experiment. The model being overall slower than the human participants could be caused by the rather low Fitts' law parameter used within ACT-R (100 ms/bit) compared to the 165 ms/bit that we observed.

Spreading activation is not only necessary for the model to be able to complete the tasks, but also to display the device-orientation effect (Figure 1). We can infer that the activation assumption is a sound explanation of the disadvantage of device-oriented steps. Too much spreading activation reduces the effect again, though. This can be explained by a ceiling effect: The average retrieval time gets close to zero for high values of mas (M_{mem} in Table 2), thereby diminishing the possibility for timing differences.

How relevant is a 100 ms difference in real life? Probably not too much by itself. What makes it important is its connection to user errors. Errors itself are hard to provoke in the lab without adding secondary tasks that interrupt the user or create strong working memory strain, thereby substantially lowering external validity.

Conclusions

The concept of device-orientation versus task-orientation is an important aspect of human-computer interaction. We could replicate that the device-oriented parts of simple goal-directed action sequences take approximately 100 ms longer than their task-oriented counterparts. With the help of cognitive modeling, associative priming could be identified as a possible explanation for this effect.

Acknowledgements The authors gratefully acknowledges financial support from the German Research Foundation (DFG) for the project "Automatische Usability-Evaluierung modellbasierter Interaktionssysteme für Ambient Assisted Living" (AL-561/13-1).

References

Altmann EM, Trafton JG (2002) Memory for goals: An activation-based model. Cognitive Sci 26(1):39–83

Ament MG, Blandford A, Cox AL (2009) Different cognitive mechanisms account for different types of procedural steps. In: Taatgen NA, van Rijn H (eds) Proceedings of the 31nd Annual Conference of the Cognitive Science Society, Amsterdam, NL, pp 2170–2175

Ament MGA (2011) The role of goal relevance in the occurrence of systematic slip errors in routine procedural tasks. Dissertation, University College London

Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004) An integrated theory of the mind. Psychol Rev 111(4):1036–1060

Byrne MD, Davis EM (2006) Task structure and postcompletion error in the execution of a routine procedure. Hum Factors 48(4):627–638

Fitts PM (1954) The information capacity of the human motor system in controlling the amplitude of movement. J Exp Psychol 47(6):381–391

Gluck KA, Stanley CT, Moore LR, Reitter D, Halbrügge M (2010) Exploration for understanding in cognitive modeling. J Artif Gen Intell 2(2):88–107

Halbrügge M (2013) ACT-CV: Bridging the gap between cognitive models and the outer world. In: Brandenburg E, Doria L, Gross A, Günzlera T, Smieszek H (eds) Grundlagen und Anwendungen der Mensch-Maschine-Interaktion – 10. Berliner Werkstatt Mensch-Maschine-Systeme, Universitätsverlag der TU Berlin, Berlin, pp 205–210

John BE, Kieras DE (1996) Using GOMS for user interface design and evaluation: Which technique? ACM Trans Comput-Hum Interact (TOCHI) 3(4):287–319

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2013) nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-113

Quade M, Halbrügge M, Engelbrecht KP, Albayrak S, Möller S (2014) Predicting task execution times by deriving enhanced cognitive models from user interface development models. In: Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, ACM, New York, pp 139–148

R Core Team (2014) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org. Accessed 7 May 2014

Stewart TC, West RL (2010) Testing for equivalence: a methodology for computational cognitive modelling. J Artif Gen Intell 2(2):69–87

Trafton JG, Altmann EM, Ratwani RM (2009) A memory for goals model of sequence errors. In: Howes A, Peebles D, Cooper RP (eds) Proceedings of the 9th International Conference of Cognitive Modeling, Manchester, UK