In D. Reitter & F. E. Ritter (Eds.), Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016). University Park, PA: Penn State.

The Sum of Two Models: How a Composite Model Explains Unexpected User Behavior in a Dual-Task Scenario

Marc Halbrügge (marc.halbruegge@tu-berlin.de)
Quality & Usability Lab, Telekom Innovation Laboratories
Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin

Nele Russwinkel (nele.russwinkel@tu-berlin.de)
Department of Cognitive Modeling in Dynamic Human-Machine Systems
Technische Universität Berlin
Marchstraße 23, 10587 Berlin

Abstract

Maintaining cognitive control while pursuing several tasks at the same time is hard, especially when the current problem states of these tasks need to be represented in memory. We are investigating the mutual influence of a self-paced and a reactive task with regard to completion time and error rates. Against initial expectations, the interruptions from the reactive task did not lead to more errors in the self-paced task, but only prolonged the completion time. Our understanding of this result is guided by a combined version of two previously published cognitive models of the individual tasks. The combined model reproduces the empirical findings concerning error rates and task completion times, but not an unexpected change in the error pattern. These results feed back into our theoretical understanding of cognitive control during sequential action.

Keywords: Human Error; Memory for Goals; Working Memory Updating; Multi-Tasking; Threaded Cognition

Introduction

Multi-tasking and handling interruptions are very common in daily life. Both have been linked to reduced performance and increased error rates, even to road accidents (Altmann, Trafton, & Hambrick, 2014; Kujala & Salvucci, 2015). According to Borst, Taatgen, and van Rijn (2015), the central problem is the necessity to maintain several problem states at once. This can lead to interference between the respective memory traces, which manifests itself as error in one or several of the concurrently processed tasks.

Starting from this premise, we augmented an existing paradigm for error research during instruction following with a working memory updating (WMU) task. The WMU task should interfere with the primary task by a) periodically interrupting the user and b) additional memory strain. We hypothesized that this would result in increased error rates in the dual-task condition compared to a single-task baseline (Byrne & Bovair, 1997). In a previous study by Ament, Cox, Blandford, and Brumby (2010) using a comparable paradigm, high memory load was connected to higher error rates especially for device-specific tasks.

Another reason for the choice of the specific WMU task was the availability of a validated cognitive model of this task (Russwinkel, Urbas, & Thüring, 2011) that could be combined with the existing model of the primary instruction following task (Halbrügge, Quade, & Engelbrecht, 2016). This allowed to test the generalizability of the models to the new paradigm and at the same time provided the possibility to *quantify* the expectations from the memory interference effect that has *qualitatively* been layed out above.

This paper has two aims. First, we want to replicate the findings of Byrne and Bovair (1997), Ament et al. (2010) and others in an applied scenario. Second, we want to explore how much effort in terms of model development is needed to combine two existing cognitive models and how well the resulting model fits to the human data. Before presenting the empirical evidence, let us first clarify the basic concepts that are used in this paper.

Sequential Action and Procedural Error

Error research is usually concerned with failures on Rasmussen's rule-based level of action control (Rasmussen, 1983), i.e., well-learned routine activities like commuting to work or preparing breakfast. Errors on this level of control are relatively rare (below 5%), but pervasive (Reason, 1990). They are defined as the violation of the optimal path to the current goal, either by adding an unnecessary action (called *intrusion*), or by skipping an action (called *omission*).

A promising model for cognitive control during rule-based behavior is the Memory for Goals theory (MFG; Altmann & Trafton, 2002). It proposes that the steps that lead to the completion of a task are represented as subgoals in declarative memory (as defined within ACT-R, Anderson et al., 2004). Whether these subgoals can be retrieved and come into action depends on general memory effects like gradually decaying *activation*, *interference*, and *priming*. These effects are sufficient to explain important features of sequential action like postcompletion errors (Byrne & Bovair, 1997) and have been successfully implemented as computational cognitive models (e.g., Trafton, Altmann, & Ratwani, 2011; Tamborello & Trafton, 2015; Halbrügge, Quade, & Engelbrecht, 2015).

Postcompletion errors occur when an action sequence contains a final step *after* the goal already has been achieved, e.g., taking the original from a photocopier (the final step) after making a copy (the goal). What is so special about this final step? It does not contribute to the users' goal, but stems from the design of the device operated by them. This property of a task has been coined *device-orientation*, its opposite being *task-orientation* (Ament, Cox, Blandford, & Brumby, 2013;

Gray, 2000). Within the MFG theory, higher omission rates for device-oriented tasks can be explained by lower activation of the corresponding subgoals. While task-oriented subgoals receive priming from the overall goal, device-oriented subgoals do not. Previous modeling studies have shown that this differentiation is sufficient to explain disadvantages of device-orientation both in the completion time and the error domain (Halbrügge & Engelbrecht, 2014; Halbrügge et al., 2015).

Because of the downsides of device-oriented tasks, they are usually avoided during the design of user interfaces (UI). In case this is not possible, device-oriented tasks are often made *obligatory*, i.e., the users are forced to perform them by the application logic. Examples for this practice are the login button that users have to press after entering their credentials, or teller machines that return the card before delivering money. Making a step obligatory is quite effective. In a previous study, the expected error increase for device-oriented steps did only occur if the respective step was also non-obligatory (Halbrügge et al., 2015). Task necessity is therefore an important factor for the genesis of errors.

According to the MFG and Byrne and Bovair's (1997) preceding work, procedural error is caused by goal forgetting which in turn can be stimulated by high working memory load. In the context of this paper, we introduce memory load based on the WMU concept.

Working Memory Updating (WMU)

WMU is a task characteristic rather than a task itself. This concept describes the ability to maintain accurate representations of information changing over time (see Ecker, Lewandowsky, Oberauer, & Chee, 2010). Ecker et al. identified three putative phases of WMU – Retrieval (R), Transformation (T) and Substitution (S) of information.

The complexity of a WMU task can vary on two dimensions: *coordinative* complexity increases with the number of representations that have to be maintained at the same time while *sequential* complexity increases with update frequency (Mayr, Kliegl, & Krampe, 1996).

In the case of Ament et al. (2010), a low and a high memory demand condition was created by manipulating the coordinative complexity of the secondary (WMU) task. During the course of their experiment, the participants produced virtual doughnuts (main task) and had to count the amount of produced items of either one (low coordinative complexity) or two (high coordinative complexity) specific kinds of doughnuts.

For the present purpose, we considered the increase in complexity from one to several WMU targets as being too large. Instead, only one target (a pictogram) had to be counted and the sequential complexity was varied depending on how quickly the count had to be updated.

Experiment

We examined our assumptions using a kitchen assistant that has been created by computer scientists of TU Berlin as part of a smart home project (Feuerstack, 2009). The assistant aids in the preparation of meals by suggesting recipes, calculating ingredients and maintaining shopping lists. Its UI features all four possible combinations of device-orientation and task necessity, examples are given in Figure 1.

Method

Participants Twelve members of the Technische Universität Berlin paid participant pool took part in the experiment. There were four men and eight women, with their age ranging from 18 to 51 (M=33.7, SD=9.5). As the instructions were given in German, only fluent German speakers were allowed to take part. Written consent was obtained from all participants.

Materials The experiment was conducted in a neutral laboratory. A personal computer with 23" (58.4 cm) monitor with optical sensor 'touch' technology was used to display the interface of the kitchen assistant. Seven pictograms of common household interruptions (e.g., phone ringing, doorbell, baby crying; see Figure 2) served as stimuli of the WMU task. The stimuli were superimposed on the UI of the kitchen assistant using dedicated Javascript code running within the browser that displayed the assistant. All user actions were recorded by the computer system. The subjects' performance was additionally recorded on videotape for subsequent error identification.

Figure 2: Examples of the pictograms used in the experiment. Image credits: Baby © UN OCHA, CC-BY 3.0; Door, Cup with tea bag © Freepik, CC-BY 3.0

Design We applied a three-factor within-subjects design, the factors being device- vs. task-orientation, task necessity (non-obligatory vs. obligatory), and secondary task difficulty (none vs. onset to onset stimulus intervals 5s, 4s, 3s). User tasks were grouped into four blocks of eleven to twelve individual tasks. Each participant was randomly assigned to one of eight pre-selected block sequences so that block position and block succession were counterbalanced across participants as well. The secondary WMU task was always introduced after the completion of the first block and its sequential demand was gradually increased from 5s stimulus interval in the second to 3s in the fourth and last block. Each interval was split into equally long stimulus and and blank phases.

Procedure Every block started with comparatively easy recipe search tasks, e.g., "search for German main dishes and select lamb chops". Users would then have to change the search attributes, e.g., "change the dish from appetizer to dessert and select baked apples". The second half of each block was made of more complex tasks that were spread over

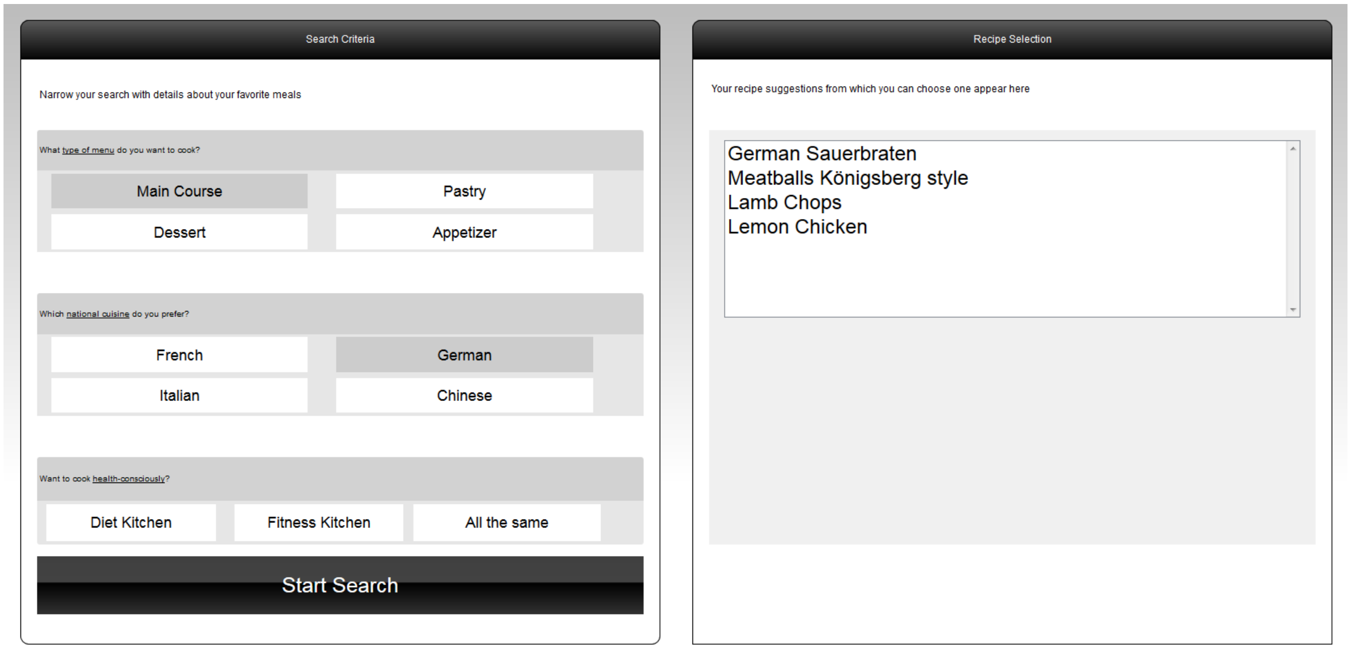

Figure 1: Screenshot of the English version kitchen assistant. Search attributes on the left are task-oriented and non-obligatory. The "Start Search" button is device-oriented and obligatory. The entries of the search results list on the right unhide subsequent options, they are therefore task-oriented and obligatory.

more screens of the interface and/or needed memorizing more items. The subjects either had to create ingredients lists for a given number of servings, or had to make shopping lists using a subset of the ingredients list, e.g., without salt and flour. All instructions were read to the subjects by the experimenter. Every individual trial was closed by a simple question the subjects had to answer to keep them focused on the kitchen setting, e.g., "how long does the preparation take?" During each instruction phase the complete screen was blanked (see Figure 3).

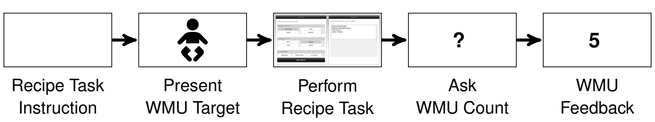

Figure 3: Sequence of screens within a single trial in the dual task condition.

In the dual-task condition, one of the seven WMU stimuli was selected as target for the current trial and presented to the participants after the instructions for the next trial had been given. Subsequently, the UI of the kitchen assistant was uncovered. WMU stimuli appeared in random order on the lower right of the screen and the participants had to count the number of appearances of the target stimulus. After the completion of the trial, the screen was blanked again and the participants were asked how often they had seen the WMU target. With an initial training phase and exit questions the whole procedure took approximately one hour.

Results

We recorded a total of 3464 user actions and 407 minutes of video. The system logs were synchronized with the videos and semi-automatically annotated using ELAN (Wittenburg, Brugman, Russel, Klassmann, & Sloetjes, 2006).

Recipe Task We recorded a 88 (2.5%) omissions and 133 (3.8%) intrusions. Contrarily to our assumptions, the error rate did not increase in the dual-task condition, nor when interruptions by the secondary task became more frequent. Adding the block (A–D in Figure 4) to a mixed logit model with task block and subject as random factors (Bates, Maechler, Bolker, & Walker, 2013) did not explain more variance ($\chi_3 = 1.30, p = .730$). Descriptively, the error rate even decreased while the memory updating task became harder.

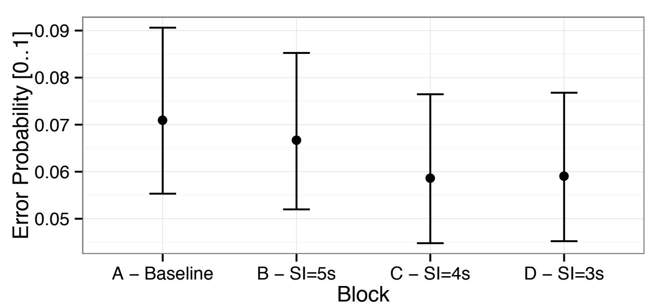

Figure 4: Error probabilities for the recipe task per experiment block. Error bars denote 95% confidence intervals based on the Agresti-Coull method.

There was a significant influence of the WMU task on the time needed to perform the recipe task. In the dual-task condition, participants needed approximately 100 ms longer per individual click (mixed model with click type[1] and subject as random factors, $t_{2695} = 2.31, p = .021$)

The influence of device-orientation and task necessity on errors was analyzed separately for omissions and intrusions (see Figure 5). Obligatory tasks led to fewer omissions than non-obligatory ones (logit mixed model with subject and task block as random factors, $z = -2.56, p = .011$). We observed fewer intrusions for obligatory tasks ($z = -4.16, p < .001$) and for device-oriented tasks as well ($z = -3.01, p = .003$). The significant interaction between both factors ($z = 2.41, p = .016$) is due to non-obligatory task-oriented actions (i.e., search attributes, left part of Figure 1) showing the highest intrusion rates.

Working Memory Updating Task Contrarily to our assumptions, the error rate in the memory updating task did not increase with the demand of task. Adding the block to a mixed logit model with task block and subject as random factors did not explain more variance ($\chi_2 = 0.12, p = .942$). Descriptively, we see a small increase, but the overall error rate is very high (see Figure 6).

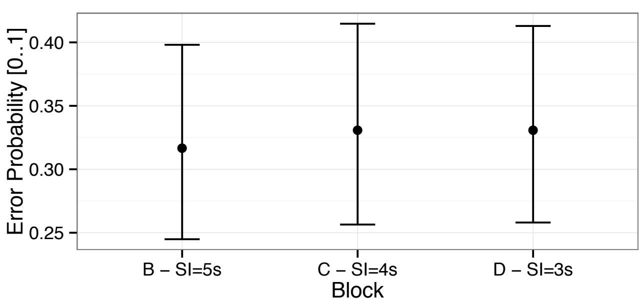

Figure 6: Error Probabilities per Block for the WMU Task. Error bars denote 95% confidence intervals based on the Agresti-Coull method.

Discussion

Interruptions are often used in error research as a means to increase the error base rates (e.g., Trafton et al., 2011; Li, Blandford, Cairns, & Young, 2008). In line with this thinking and based on the results of previous research (e.g., Byrne & Bovair, 1997; Ament et al., 2010), we expected that the increased memory load due to the WMU task would result in degraded performance in the main recipe task. But the empirical data tells a different story. While the participants needed more time to complete the recipe tasks, they did not make significantly more errors. Why is this the case?

First, the baseline error rate of 7% is already quite high, in particular compared to the 1.3% observed during a previous study using a similar paradigm (Halbrügge et al., 2015). This could be due to the blanking of the screen during the instruction phase that was added to the procedure in the present experiment. The blanking should have impaired the learning of the UI of the kitchen assistant. In previous studies, the participants could visually plan their actions during the instruction phase, while the current experiment demanded memorizing all subgoals without any visual reference.

Second, the high error rate of the WMU task suggests that the participants spent most attention on the recipe task. But as we only observed a very slight decrease of WMU performance with increased difficulty, this point remains unsatisfactory.[2] The analysis of the tasks based on the cognitive model presented below will provide additional insights.

The prolongation of the time needed to perform the recipe task in the dual-task condition is in line with the expected interference between both tasks. The real effect is probably underestimated by our analysis, because we found learning effects of approximately -50 ms per block in previous studies (Halbrügge & Engelbrecht, 2014). Assuming that learning still took place in the current study, it should have counteracted the prolongation effects of increasing WMU complexity.

Cognitive Model

Based on the well-established MFG theory, we proposed that having to perform two memory-intensive tasks would lead to more errors, but the data did not confirm our hypothesis. Does this disprove the theory, or was our understanding of it insufficient? In order to elaborate on the second option, we combined an existing MFG-based model of sequential behavior (Halbrügge et al., 2015) with an existing model of WMU (Russwinkel et al., 2011) using the threaded cognition extension of ACT-R (Anderson et al., 2004). The threaded cognition theory (Salvucci & Taatgen, 2008) assumes that task switching is not necessarily conscious behavior, but may emerge as concurrent tasks have to wait for cognitive resources (e.g., memory, vision) that are currently held by other tasks.

Recipe Task Model

The recipe task model extends on the MFG theory (Altmann & Trafton, 2002) by highlighting the importance of environmental cues during sequential behavior. Whenever the purely memory-based process as proposed by the MFG fails, the model reverts to a vision-based strategy that searches the environment for appropriate cues for the next action to take (see flowchart in Figure 7). This addition has been shown to be explain the effects of obligatory vs. non-obligatory steps with regards to omissions (Halbrügge et al., 2015), it has expanded MFG-based models to intrusion errors, and it has recently been confirmed by gaze data (Halbrügge et al., 2016).

For the current paper, the model was adapted to threaded cognition by adding extra checks for the current availability

[1] Four click type groups: same button, same group of buttons, different group of buttons, buttons on different UI screens; see Quade, Halbrügge, Engelbrecht, Albayrak, and Möller (2014) for reference.

[2] Unfortunately, Ament et al. (2010) give no results of the secondary task that could be used for comparison.

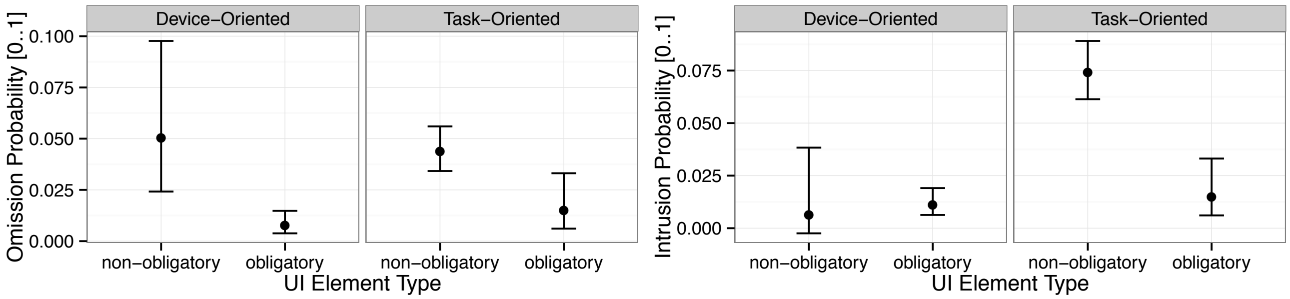

Figure 5: Omission and intrusion probabilities per UI element type. Error bars denote 95% confidence intervals based on the Agresti-Coull method.

of the declarative module to several productions. All numerical ACT-R parameters remained unchanged.

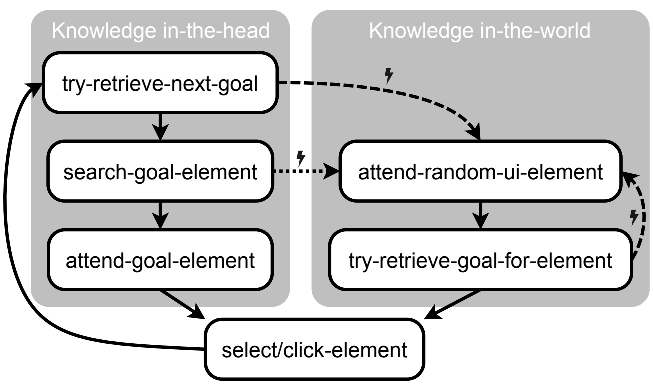

Figure 7: Simplified Flow Chart of the recipe task model. Dashed arrows denote retrieval errors, the dotted arrow denotes visual search failure.

Working Memory Updating Model

The code was adapted from an existing model that has been tested in different kind of tasks and settings (Russwinkel et al., 2011; Pape & Urbas, 2009; Russwinkel & Schinkmann, 2011). The WMU model uses a single representation (i.e., memory chunk) for each target that pairs it with its current count. This representation is manipulated using the three Retrieval, Transformation, and Substitution phases as proposed by Ecker et al. (2010). After its retrieval (R), the count slot of the chunk is updated (T). The resulting new representation is encoded in declarative memory (S), where it may interfere with older versions featuring outdated count values.

In the combined model, buffer stuffing is used to detect the visual targets of the WMU task. In case a new object is found at the right bottom of the screen and both the visual and the declarative modules are available, the model attends the new object and at the same time retrieves the most highly activated WMU count chunk. Because of activation noise, the declarative module may return an older copy of that chunk which subsequently leads to an error.

Goodness-of-Fit

The combined model[3] was run 500 times and all resulting errors and completion times were recorded. Contrarily to our expectations, but consistent with the empirical findings, the combined model does not show an increased error rate when the WMU task is present (odds ratio = 1.04, well within the empirical 95% CI from 0.62 to 1.20).

Regarding the effects of device-orientation and task necessity, the overall model fit is not good with R^2=.174 and RMSE=.029. This is mainly due to the unexpectly high intrusion rate for non-obligatory task-oriented steps (see Figure 8). When regarding only omissions, the fit is much better with R^2=.789 and RMSE=.014.

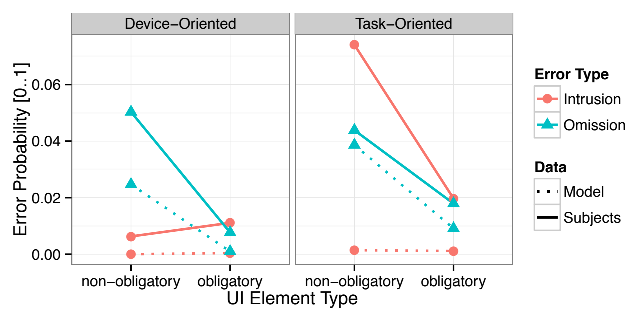

Figure 8: Model Predictions and Empirical Error Rates.

The combined model does show longer click times in the dual-task condition. Here, each click takes 120 ms longer on average, which is close to the empirical effect of 94 ms (95% CI from 14 ms to 174 ms).

Discussion

When the combined model performs both the recipe and the WMU task, the number of errors in the recipe task does not increase, it just takes longer to perform the individual actions. This means that our empirical results, although being unexpected, fit with the theoretical underpinnings presented above. Close inspection of the model traces shows that the

[3]The source code of the cognitive model is available for download at http://dx.doi.org/10.5281/zenodo.55224

model predictions are caused by both tasks demanding visual and memory ressources. Only during the motor phase of the recipe task model (i.e., when a click is performed, “select/click-element” in Figure 7), the WMU task can take over. This observation is also consistent with the high error rate of the WMU task during the experiment.

Combining both models using threaded cognition was less easy than expected. A central assumption of threaded cognition is that ACT-R buffers are shared between all running tasks. In our case, this made some states ambigous. Especially retrieval errors could not easily be attributed to the recipe or the WMU task (R phase of the WMU model vs. dashed arrows in Figure 7). This problem was solved by using the visual system. Retrieval errors are only attributed to the WMU task if the model visually attends a WMU target at the same time.

General Discussion and Conclusions

We have presented empirical data and a cognitive model of human performance and error in a dual-task scenario. Contrarily to our expectations, the dual-task condition did not lead to more errors, but longer task completion times, only. These results are nonetheless consistent with the Memory for Goals theory that had led to our expectations, as shown by the cognitive model simulations. The good fit is remarkable as no numerical parameter fitting was applied to achieve it. To our knowledge, this is also the first time that a MFG model has been combined with the threaded cognition theory.

The model has several limitations. First, the increased overall error rate, especially concerning intrusions, is not covered by the model. Second, the model does not show any specific visual behavior during the instruction phase, but only listens to the experimenter. Compared to previous studies that did not use a blank screen during the instruction phase (Halbrügge et al., 2015, 2016), the current data show relatively high error rates even in the single-task condition. This suggests that the human participants visually prepared their action sequence while listening to the experimenter in previous studies. More research is needed to elaborate on this point. We are therefore planning to examine the visual processing of the screen during sequence planning using eye-tracking.

As final remark we would like to highlight how the approach taken here exemplifies the benefits of computational cognitive modeling as a method. Because of the use of ACT-R as common denominator, is was possible to take two cognitive models created by different researchers and to combine them to something new that created new evidence and sparked new questions.

Acknowledgments

This work was partially supported by DFG grant MO 1038/18-1 (“Automatische Usability-Evaluierung modellbasierter Interaktionssysteme für Ambient Assisted Living”).

References

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive science*, *26*(1), 39–83. doi: 10.1207/s15516709cog2601_2

Altmann, E. M., Trafton, J. G., & Hambrick, D. Z. (2014). Momentary interruptions can derail the train of thought. *Journal of Experimental Psychology: General*, *143*(1), 215–226. doi: 10.1037/a0030986

Ament, M. G., Cox, A. L., Blandford, A., & Brumby, D. (2010). Working memory load affects device-specific but not task-specific error rates. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd annual conference of the cognitive science society* (pp. 91–96). Portland, OR.

Ament, M. G., Cox, A. L., Blandford, A., & Brumby, D. P. (2013). Making a task difficult: Evidence that device-oriented steps are effortful and error-prone. *Journal of experimental psychology: applied*, *19*(3), 195. doi: 10.1037/a0034397

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036–1060. doi: 10.1037/0033-295X.111.4.1036

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2013). lme4: Linear mixed-effects models using eigen and s4 [Computer software manual]. (R package version 1.0-5)

Borst, J. P., Taatgen, N. A., & van Rijn, H. (2015). What makes interruptions disruptive? a process-model account of the effects of the problem state bottleneck on task interruption and resumption. In *Proc. chi 2015* (pp. 2971–2980). doi: 10.1145/2702123.2702156

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive science*, *21*(1), 31–61. doi: 10.1207/s15516709cog2101_2

Ecker, U. K., Lewandowsky, S., Oberauer, K., & Chee, A. E. (2010). The components of working memory updating: an experimental decomposition and individual differences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*(1), 170.

Feuerstack, S. (2009). *A method for the user-centered and model-based development of interactive applications* (Doctoral dissertation, Technische Universität Berlin). doi: 10.14279/depositonce-2078

Gray, W. D. (2000). The nature and processing of errors in interactive behavior. *Cognitive Science*, *24*(2), 205-248. doi: 10.1016/S0364-0213(00)00022-7

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

Halbrügge, M., Quade, M., & Engelbrecht, K.-P. (2015). A predictive model of human error based on user interface development models and a cognitive architecture. In N. A. Taatgen, M. K. van Vugt, J. P. Borst, & K. Mehlhorn (Eds.), *Proceedings of the 13th international conference on cognitive modeling* (p. 238-243). Groningen, the Netherlands: University of Groningen.

Halbrügge, M., Quade, M., & Engelbrecht, K.-P. (2016). Cognitive strategies in hci and their implications on user error. In *Proceedings of the 38th annual meeting of the cognitive science society.* (accepted)

Kujala, T., & Salvucci, D. D. (2015). Modeling visual sampling on in-car displays: The challenge of predicting safety-critical lapses of control. *International Journal of Human-Computer Studies*. doi: 10.1016/j.ijhcs.2015.02.009

Li, S. Y., Blandford, A., Cairns, P., & Young, R. M. (2008). The effect of interruptions on postcompletion and other procedural errors: an account based on the activation-based goal memory model. *Journal of Experimental Psychology: Applied*, *14*(4), 314. doi: 10.1037/a0014397

Mayr, U., Kliegl, R., & Krampe, R. T. (1996). Sequential and coordinative processing dynamics in figural transformations across the life span. *Cognition*, *59*(1), 61–90.

Pape, N., & Urbas, L. (2009). Testing a quantitative model of time estimation in a load-switch scenario. In A. Howes, D. Peebles, & R. P. Cooper (Eds.), *Proceedings of the 9th international conference of cognitive modeling.* Manchester, UK.

Quade, M., Halbrügge, M., Engelbrecht, K.-P., 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* (pp. 139–148). New York, NY, USA: ACM. doi: 10.1145/2607023.2607033

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *Systems, Man and Cybernetics, IEEE Transactions on*, *13*, 257–266. doi: 10.1109/TSMC.1983.6313160

Reason, J. (1990). *Human error*. New York, NY: Cambridge University Press.

Russwinkel, N., & Schinkmann, M. (2011). Influence of working memory updating on time perception. In *9. berliner werkstatt mensch-maschine-systeme.*

Russwinkel, N., Urbas, L., & Thüring, M. (2011). Predicting temporal errors in complex task environments: A computational and experimental approach. *Cognitive Systems Research*, *12*(3), 336–354. doi: 10.1016/j.cogsys.2010.09.003

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, *115*(1), 101-130. doi: 10.1037/0033-295X.115.1.101

Tamborello, F. P., & Trafton, J. G. (2015). Action selection and human error in routine procedures. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 59, pp. 667–671). doi: 10.1177/1541931215591145

Trafton, J. G., Altmann, E. M., & Ratwani, R. M. (2011). A memory for goals model of sequence errors. *Cognitive Systems Research*, *12*, 134–143. doi: 10.1016/j.cogsys.2010.07.010

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., & Sloetjes, H. (2006). ELAN: a professional framework for multimodality research. In *Proceedings of lrec* (Vol. 2006).