Modeling Triple-Tasking without Customized Cognitive Control

Jelmer P. Borst (jpborst@ai.rug.nl)[1]
Niels A. Taatgen (taatgen@cmu.edu)[1,2]
Hedderik van Rijn (d.h.van.rijn@rug.nl)[1]
[1]Dept. of Artificial Intelligence, University of Groningen, The Netherlands
[2]Dept. of Psychology, Carnegie Mellon University, USA

Abstract

Cognitive models of multitasking typically use control strategies that are customized for the tasks at hand. Salvucci and Taatgen (2008) have shown that it is possible to account for dual-tasking without using customized control: they let task properties determine how tasks are interleaved. If this is how the human cognitive system tackles multitasking, it should be possible to account in the same way for more than two tasks. In the current paper we investigate whether this approach can be extended to three concurrent tasks. Two experiments are presented: a dual- and a triple-task. We show that cognitive models without fixed control strategies cannot only account for the dual-task, but for the triple-task as well.

Keywords: multitasking; threaded cognition; cognitive control, ACT-R.

Introduction

The ability to execute multiple tasks at the same time is an impressive feat of the human cognitive system. People can almost effortlessly combine previously unrelated tasks. To account for multitasking, most cognitive models use a *customized executive* (Kieras et al., 2000): a control strategy that determines how tasks are interleaved and that is specialized, and only suitable for, the tasks at hand. This seems to be at odds with the observation that people can flexibly combine tasks. A control strategy suitable for combining arbitrary tasks, a *general executive,* seems to be a more plausible psychological construct (e.g., Kieras et al., 2000; Salvucci, 2005). Kieras et al. implemented such a general executive, but concluded that their customized executive model accounted better for the human data.

Recently, Salvucci and Taatgen proposed a new theory of multitasking, called 'threaded cognition' (2008; Salvucci, Taatgen, & Borst, 2009). Threaded cognition does not assume any task-specific supervisory or executive processes and can therefore combine arbitrary tasks. Salvucci and Taatgen have shown that it accounts well for dual-tasking in a number of different domains, ranging from dual-choice tasks to driving a car and using a cell phone concurrently. According to threaded cognition, human multitasking is not limited by the number of tasks that have to be performed, but only by the capacity of general cognitive resources. We will test this assumption by extending the approach to three concurrent tasks. First, we will describe a dual-tasking experiment and a well-fitting cognitive model; secondly, we will add a task to the experiment, and show that the existing model can account for the new data by just adding the new task to it. Before describing the experiments, we will briefly introduce the threaded cognition theory.

Threaded cognition is implemented in the cognitive architecture ACT-R (Anderson, 2007). ACT-R describes human cognition as a set of independent modules that interact through a central production system. For instance, it uses a visual and an aural module for perception and a motor module to interact with the world. Besides these peripheral modules, ACT-R also has a number of central cognitive modules: the procedural module that implements the central production system, the declarative memory module, the goal module, and the problem state module (sometimes referred to as imaginal module or problem representation module). All modules operate in parallel, but each module in itself can only proceed serially. Thus, the visual module can only perceive one object at a time and the memory module can only retrieve one fact at a time.

A task is represented in ACT-R by the contents of the goal module and the problem state module. In the case of solving an algebra problem like '$8x–5 = 7$', the goal module can hold for instance 'algebra - unwinding', while the problem state module can be used to hold the intermediate solution '$8x = 12$' (Anderson, 2007). Thus, the goal module holds the current state of a task, while the problem state holds intermediate information necessary for performing the task. In line with the serial processing in the other modules, the goal module can only hold a single goal and the problem state module can only hold a single problem state.

Threaded cognition extends ACT-R by allowing for multiple goals, and thus multiple tasks (called 'threads'), to be kept active (Salvucci & Taatgen, 2008). While it is proposed that the goal module can hold multiple goals, the other modules are still singular, and have to be shared by the different threads. The modules are shared on a first-come-first-served basis: a thread will 'greedily' use a module when it needs it, but also has to let go of it 'politely', that is, as soon as it is done with it. The seriality of the modules results in multiple potential bottlenecks: when two threads need a module concurrently, one will have to wait for the other (Salvucci & Taatgen, 2008; Borst & Taatgen, 2007). Note that while the modules are serial in themselves, the different modules operate in parallel.

In Figure 1 an example processing stream of a dual-task is shown: white boxes depict a task in which a key-press is required in reaction to a visual stimulus and grey boxes depict a task in which a vocal response is required in reaction to an auditory stimulus. The A shows interference, caused by the fact that both tasks want to use the procedural module concurrently: the aural-vocal task has to wait for the visual-manual task.

Experiment 1: Subtraction & Text Entry

Experiment 1 consists of a complex dual-task, to show how threaded cognition can account for such a task without using customized cognitive control. Participants had to perform two tasks concurrently: a subtraction task and a text entry task. Both tasks were presented in two versions: an easy version in which there was no need to maintain a problem state, and a hard version where participants had to maintain a problem state from one response to the next. As threaded cognition claims that the problem state module can only be used by one task concurrently, we hypothesized that when a problem state is required in both tasks, participants will be significantly slower or make more errors than in the other conditions (cf., Borst & Taatgen, 2007).

Method

Participants 15 students of the University of Groningen participated in the experiment for course credit (10 female, age range 18-31, mean age 20.1). All participants had normal or corrected-to-normal visual acuity. Informed consent was obtained before testing.

Design During the experiment participants had to perform a subtraction task and a text entry task concurrently. The subtraction task was shown on the left side of the screen, the text entry task on the right. Participants had to alternate between the tasks: after entering a number, the subtraction task was disabled, forcing a participant to subsequently enter a letter. After entering a letter, the text entry task was disabled and the subtraction task became available again.

The interface of the subtraction task is shown on the left side of Figure 2. Participants had to solve 10-digit subtraction problems in right to left order; they had to respond with their left hand using the keyboard. In the easy version, the upper term was always larger or equal to the lower term; these problems could be solved without ‘borrowing’. In contrast, the hard version (Figure 2) required participants to borrow six times. The assumption is that participants need to use their problem state resource to keep track of whether a ‘borrowing’ is in progress.

The second task in the experiment was text entry. The interface is shown on the right in Figure 2: by clicking on the keypad 10-letter words had to be entered. In the easy version of the text entry task, the words were presented one letter at a time. Participants had to click the corresponding button on the keypad, after which the next letter appeared. In the hard version, a complete word appeared at the start of a trial. When the participant clicked on the first letter, the word disappeared and had to be entered without feedback

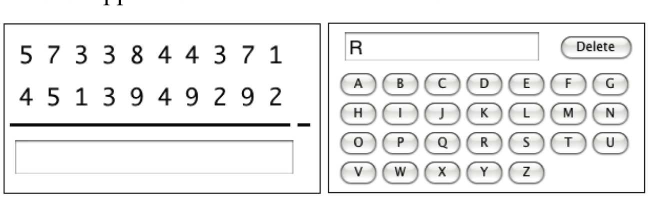

Figure 2. Example screens of the experiments. Note that there is more space between the tasks in the experiment.

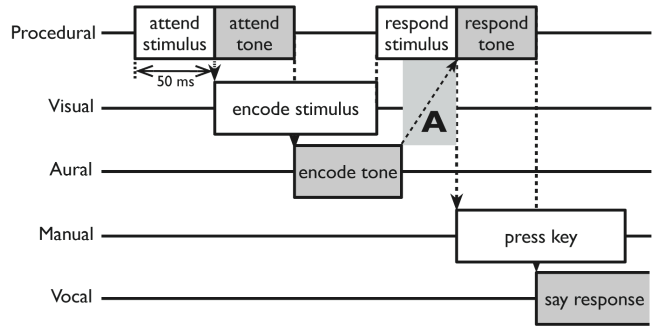

Figure 1. Module processing in threaded cognition in a dual-choice task. The ‘A’ depicts interference.

(thus, participants could neither see what word they were entering, nor how many letters they had entered). Here we assume that participants need their problem state to keep track of what word they were entering. When the text entry interface was disabled to force alternation between the tasks, the mouse pointer was hidden to prevent participants from putting the pointer on the next letter as a memory aid.

Stimuli The stimuli for the subtraction task were generated anew for each participant. The subtraction problems in the hard version always featured six ‘borrowings’, and resulted in 10-digit answers. The 10 letter words for the hard version of the text entry task were handpicked from a list of high frequent Dutch words (CELEX database), to ensure that similarities between words were kept to a minimum. These stimuli were also used in the easy text entry task, except that the letters within the words were scrambled to create nonsense letter strings, under the condition that a letter never appeared twice in a row.

Procedure A trial started with the appearance of the two tasks. Participants could choose which task to start with; after the first response they were forced to alternate between the tasks. After the last response of a task, a feedback display appeared, showing how many letters/numbers were entered correctly. After giving the last response of a trial, there was a 5 second break until the next trial.

Before the experiment, participants completed 6 practice trials for the separate tasks, and 4 dual-task trials. The experiment consisted of three blocks. Each block consisted of four sets of three trials per condition. These condition-sets were randomized within a block, with the constraint that the first condition of a block was different from the last condition in the previous block. Thus, 36 trials had to be performed overall. The experiment lasted about 45 minutes.

Model

We will first describe the model[1], after which the behavioral and modeling results will be presented side by side.

Of particular importance for the tasks at hand is ACT-R’s problem state module. This module can hold a problem

[1] Available at http://www.ai.rug.nl/cogmod/models/.

state, accessible at no time cost. However, changing a problem state takes 200 ms (Anderson, 2007). Because the problem state module can only hold information on a single task at a time, the module has to be updated multiple times when multiple tasks require a problem state. For instance, when thread A needs to inspect its problem state and the resource is already occupied by a problem state of thread B, thread A has to retrieve its own state from memory and restore it. By restoring the problem state, the other thread's problem state is automatically moved to memory. Thus, when multiple threads need the problem state, the execution time of tasks is increased by 200 ms plus retrieval time per change of task. Note that because problem states need to be retrieved from memory, it is possible that a thread retrieves an older, and thus incorrect, problem state from memory, often resulting in behavioral errors.

The experiment consisted of two independent tasks, implemented as two threads. Both threads use the visual module to perceive the stimuli and the manual module to operate the mouse and the keyboard. In the easy version of the subtraction task, the model perceives the numbers, retrieves a fact from memory (e.g., 5-2=3) and enters the difference. In the hard version, the model also retrieves a fact from memory, if its outcome is negative (e.g., 3-6 =-3) the model will add 10 to the upper term, store in its problem state that a ‘borrowing’ is in progress, and retrieve a new fact (13-6=7). If the problem state indicates that a ‘borrowing’ is in progress, the model first subtracts 1 from the upper term before the initial retrieval is made.

In the easy version of the text entry task, the model perceives the letter and clicks on the corresponding button. In the hard version, the model has to recall for each response what the target word was, and what the current position is. Thus, it requires the problem state resource to store what word it is entering and at which position of the word it is (‘informatie, 4^{th} position’).

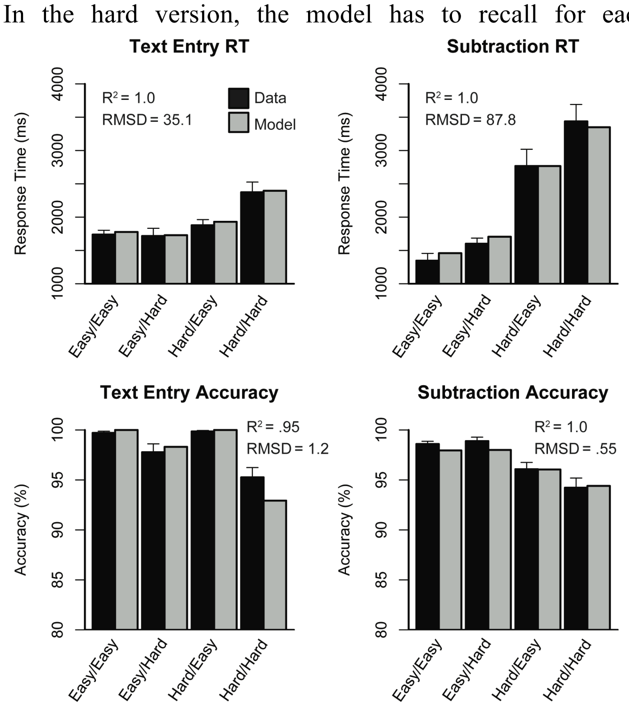

Figure 3. Results of Experiment 1. Labels represent Subtraction / Text Entry.

Thus, when both tasks are hard the model requires two problem states, one for each task. As the problem state resource can only maintain one state concurrently, the problem state has to be constantly replaced in the hard/hard condition. In comparison, all other conditions require at most a single problem state. Therefore, the model predicts considerable interference in the hard/hard condition.

Results

Only the data of the experimental phase were analyzed. Two participants did not adhere to task instructions and were removed from the dataset. Outliers in response times exceeding 2.5 standard deviations from the mean per condition per participant were removed (2.9% of the data). All reported *F*- and *p*-values are from repeated-measure ANOVAs, all error bars depict standard errors. Accuracy data was transformed using an arcsine transformation before doing the ANOVA. Figure 3 shows all main results, black bars depict experimental data, grey bars model data.

Response time on the text entry task was defined as the time between entering a number in the subtraction task and clicking on a button of the text entry task. First responses of each trial were removed. The upper left panel of Figure 3 shows the results. First, an interaction effect between Subtraction Difficulty and Text Entry Difficulty ($F(1,12)=27.78$, $p<.001$) was found. Next, we performed a simple effects analysis, showing an effect of Text Entry Difficulty when subtraction was hard ($F(1,12)=11.47$, $p<.01$), and an effect of Subtraction Difficulty when text entry was hard ($F(1,12)=55.87$, $p<.001$), both effects driving the interaction. The other simple effects did not reach significance. Thus, there was an over-additive effect of task difficulty on response times of the text entry task; participants were slowest to respond in the hard/hard condition, no other effects were found.

Figure 3, upper right panel, shows the average response times on the subtraction task. This is the time between clicking a button in the text entry task and entering a number in the subtraction task. Again, first responses of a trial were removed, as were responses that occurred in the hard conditions before a ‘borrowing’ had taken place, as those are in effect easy responses. An interaction effect between Subtraction Difficulty and Text Entry Difficulty was observed ($F(1,12)=5.50$, $p=.04$). A simple effects analysis revealed that all effects were significant: Subtraction Difficulty when text entry was easy ($F(1,12)=65.02$, $p<.001$), Subtraction Difficulty when text entry was hard ($F(1,12)=105.11$, $p<.001$), Text Entry Difficulty when subtraction was easy ($F(1,12)=13.27$, $p<.01$), and Text Entry Difficulty when subtraction was hard ($F(1,12)=12.29$, $p<.01$). Thus, the more difficult the tasks, the higher the response times, with an over-additive effect in the hard/hard condition, reflected in the interaction.

Figure 3, lower left panel, shows the accuracy on the text entry task, in percentage correctly entered letters. Both main effects were significant: Subtraction Difficulty ($F(1,12)$ = 7.31, p=.02), and Text Entry Difficulty ($F(1,12)$=21.57, p<.001). The interaction effect shows a trend towards significance ($F(1,12)$=4.65, p=.052). Thus, accuracy on the text entry task was lower when one of the tasks was hard.

In the lower right panel of Figure 3, the accuracy on the subtraction task is shown. Here, a significant interaction effect between Subtraction Difficulty and Text Entry Difficulty was observed: $F(1,12)$=10.50, p<.01. A simple effects analysis subsequently revealed that three simple effects reached significance: Text Entry Difficulty when subtraction was hard ($F(1,12)$=6.68, p=.02), Subtraction Difficulty when text entry was easy ($F(1,12)$=7.17, p=.02), and Subtraction Difficulty when Text Entry was hard ($F(1,12)$=87.7, p<.001). Text Entry Difficulty when subtraction was easy did not reach significance. Thus, when subtraction is difficult accuracy decreases, but even more so when text entry is difficult as well.

The grey bars in Figure 3 show the results of the model. It seems to reflect the data faithfully; R^2- and Root Mean Squared Deviation-values are displayed in the graphs.

Discussion

The interaction effects in the data are in agreement with our model predictions: a time penalty for both tasks in the hard/hard condition. As described above, the model explains the interaction effects by proposing a problem state bottleneck: this results in higher response times on the one hand (caused by constantly swapping out the problem state) and higher error rates on the other (caused by retrieving older, wrong problem states). The errors in the other conditions are caused by sometimes retrieving wrong facts from memory (i.e., 9 - 6 results in 2 instead of 3).

The model does not use customized executive control, but instead lets the use of general cognitive resources determine how the tasks are interleaved (see also Figure 1). Threaded cognition claims that this is how multitasking functions in general, and it should therefore be possible to extend this approach to more than two tasks. To test this, we extended both the experiment and the model with a third task.

Experiment 2: Triple-tasking

For Experiment 2, a listening task was added to the first experiment. While performing the subtraction and text entry tasks, participants had to listen to short stories, about which questions had to be answered. The experiment consisted of two parts, one part in which the task was comparable to Experiment 1, and one part in which the participants had to carry out the listening task as well. To measure baseline performance, the listening task was also tested separately.

Method

Participants 23 students of the University of Groningen participated in the triple-task experiment for course credit; one participant had to be excluded because of technical difficulties, resulting in 22 complete datasets (17 female, age range 18-47, mean 22.0). 6 students participated in the listening baseline experiment (5 female, age range 18-21, mean 19.3). All participants had normal or corrected-to-normal visual acuity and normal hearing. Informed consent was obtained before testing.

Design The subtraction and text entry tasks remained unchanged, apart from one thing: columns in the subtraction task that were solved were masked with #-marks, preventing display-based strategies. The listening task consisted of listening to a short story during each trial, about which a multiple-choice question was asked at the end of the trial. After answering the question, participants received accuracy feedback, to ensure they kept focusing on the stories. The design of the baseline experiment was similar, instead of the subtraction and text entry tasks a fixation cross was shown.

Stimuli Stimuli for the subtraction and text entry task were the same as in Experiment 1, except that six additional words were selected. The listening task was compiled out of two official Dutch listening comprehension exams (NIVOR-3.1/3.2, Cito Arnhem 1998). The story length ranged between 17 and 48 seconds (mean 30.4, sd 10.9). The multiple-choice questions consisted of three options.

Procedure The procedure was identical to Experiment 1 if not noted otherwise. At the start of a trial, participants had to start with the subtraction task. In the listening condition, playback of the story was initiated simultaneously with the presentation of the subtraction task. A question for the listening task was presented either after the feedback screens of the other tasks, or after the story was completely presented, whichever came last. The feedback screen for the listening task was presented for 4 seconds after answering the question. Participants were instructed that the listening task was the most important task, and had to be given priority over the other tasks, while still performing the other tasks as quickly and accurately as possible.

Participants practiced 4 example stories. The experiment consisted of 4 blocks of 12 trials each, 48 trials in total, in a similar setup as Experiment 1. Either the first two blocks were combined with the listening task, or the last two blocks, counterbalanced over participants. The order of the stories was randomized. The complete experiment lasted approximately 60 minutes.

Model

The same model as for Experiment 1 was used for the subtraction and text entry tasks, adjusted for the changes in arithmetic skills (i.e., retrieval speed) between both groups of participants.

To model the listening task, we added a third thread to the model. This thread aurally perceives words, retrieves spelling and syntactic information from memory, and builds simulated syntactic trees. The same approach was used in Salvucci and Taatgen (2008) to model the classical reading and dictation study by Spelke, Hirst and Neisser (1976) and by Van Rij et al. (2009) to account for children's pronoun processing in speech. This model is based on Lewis and Vasishth's model of sentence processing (2005), that constructs syntactic trees for sentence processing. For the current model we do not need that kind of linguistic detail, as we are mostly interested in how the tasks influence one another in a multitasking setting. Thus, it suffices to account for the use of procedural and declarative memory. For each word, two procedural rules fire and two facts have to be retrieved from memory, which results in about 160 ms processing time per word. We did not add any control or executive mechanisms, the threads function independently.

Results

The same exclusion criteria were used as in Experiment 1 (8.2% of the data was rejected). One question from the listening task was removed, as it was consequently answered incorrectly. If not noted otherwise, analyses were the same as in Experiment 1.

Figure 4, upper panel, shows response times on the text entry task, on the left without and on the right with the listening task. As there is no effect of Listening, nor any interaction effects involving Listening (all *F*'s<1), we collapsed over Listening. The interaction between Text Entry and Subtraction Difficulty was significant ($F(1,21)=52.55$, $p<.001$), and a simple effects analysis showed effects of Text Entry Difficulty when subtraction was hard ($F(1,21)=35.95$, $p<.001$), Subtraction when text entry was easy ($F(1,21)=32.74$, $p<.001$), and Subtraction Difficulty when text entry was hard ($F(1,21)=82.75$, $p<.001$). Text Entry Difficulty when subtraction was easy did not reach significance. Thus, there was no effect from the listening task on the response times on the text entry task. Response times did increase when subtraction became hard and even more when both tasks were hard: resulting in the interaction effect.

The lower panel of Figure 4 shows response times on the subtraction task, on the left without and on the right with the listening task. The analysis shows no main effect of Listening ($F(1,21)=2.46$, $p=.13$), but an interaction effect between Listening and Text Entry Difficulty ($F(1,21)=7.10$, $p=.01$), and an interaction effect between Listening, Subtraction Difficulty and Text Entry Difficulty ($F(1,21)=4.91$, $p=.04$). Therefore, we analyzed the response times with and without listening separately. Without listening, the interaction effect between Subtraction and Text Entry Difficulty was significant ($F(1,21)=18.64$, $p<.001$), as were all four simple effects (Text Entry Difficulty when subtraction was easy: $F(1,21)=26.69$, $p<.001$; Text Entry Difficulty when subtraction was hard: $F(1,21)=27.37$, $p<.001$; Subtraction Difficulty when text entry was easy: $F(1,21)=403.94$, $p<.001$; Subtraction Difficulty when text entry was hard $F(1,21)=172.83$, $p<.001$). With listening, the interaction effect between Subtraction and Text Entry Difficulty is significant as well ($F(1,21)=8.91$, $p<.01$), as were three simple effects (Text Entry Difficulty when subtraction was hard: $F(1,21)=12.7.69$, $p<.01$; Subtraction Difficulty when text entry was easy: $F(1,21)=302.9$, $p<.001$; Subtraction Difficulty when text entry was hard $F(1,21)=224.6$, $p<.001$). Only the effect of Text Entry Difficulty when subtraction was easy did not reach significance. Thus, both with and without Listening there was an over-additive interaction effect of Subtraction and Text Entry Difficulty, and response times increased with task difficulty in general. Furthermore, in general, participants were slower to respond when they had to perform the listening task as well.

Due to space constraints, accuracy data of the subtraction and text entry tasks are not shown; the data are comparable to Experiment 1. Figure 5, left panel, shows the accuracy data of the listening task. The leftmost bar shows the results when participants only performed the listening task: 89% correct. Adding the other tasks had little effect, except when both subtraction and text entry were hard. The interaction between Subtraction and Text Entry Difficulty was significant ($F(1,21)=7.42$, $p=.01$); as were the simple effects of Text Entry Difficulty when subtraction was hard ($F(1,21)=9.18$, $p<.01$) and Subtraction Difficulty when text entry was easy ($F(1,21)=14.75$, $p<.001$), driving the interaction effect. The simple effects of Text Entry Difficulty when subtraction was easy and Subtraction Difficulty when text entry was easy were not significant.

As can be seen in Figure 4, the response times of the model fit well to the human data, especially when taken into account that this model was not especially constructed to fit this dataset. R^2- and RMSD-values are shown in Figure 4.

The right panel of Figure 5 shows the percentage of words processed by the model. The model can only process words

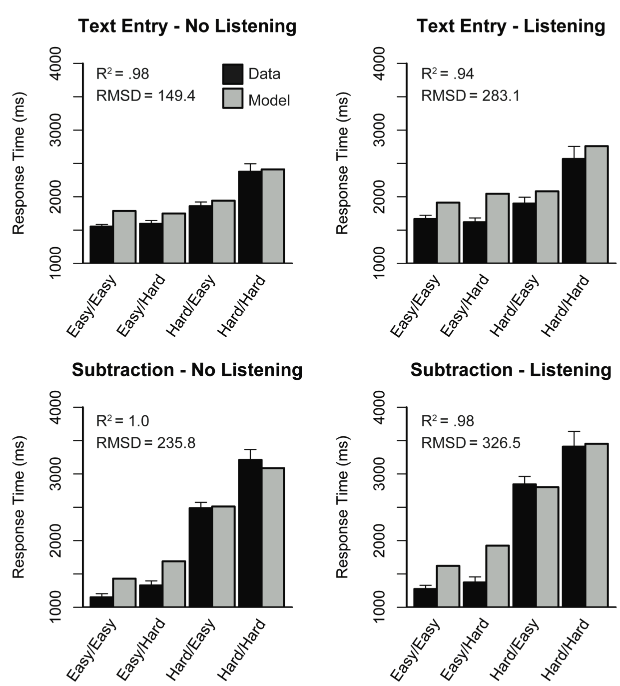

Figure 4. Response times of Experiment 2, labels represent Subtraction / Text Entry.

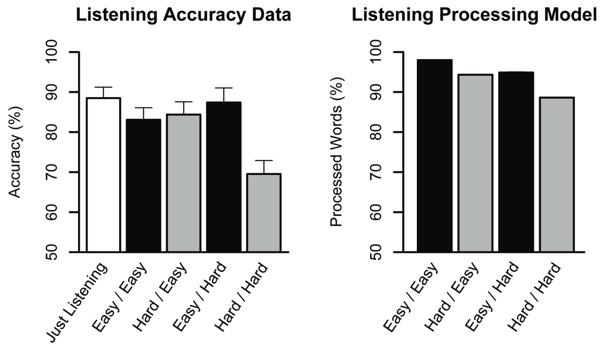

Figure 5. Accuracy data on the listening task.

when declarative memory is available. Thus, when words are presented while declarative memory is in use by the other tasks, words cannot be processed. This happens most often in the hard/hard condition, as problem states have to be retrieved for the other tasks on each step of a trial, blocking declarative memory. Percentage of processed words cannot be translated directly into number of correctly answered questions, but the model does display a rough qualitative fit: R^2=.68.

Discussion

Surprisingly, the listening task had little influence on the other two tasks. This is accounted for by the model: the listening task is continuously interleaved with the other tasks, and usually uses the slack time of the other tasks to make its declarative memory retrievals, resulting in little interference. Only the response times of the subtraction task increased slightly, which is explained by the fact that the model uses declarative memory to process the words. This sometimes blocks declarative memory for the subtraction task, resulting in a small increase of the response times, as was also observed in the human data. We did not add this explicitly to the model: it emerged out of the interaction between the listening thread and the subtraction thread.

General Discussion

Experiment 1 showed that the threaded cognition theory accounts for complex dual-task data, explaining the major phenomena by proposing a problem state bottleneck. In Experiment 2, the challenge was increased by adding a third task: listening to stories. Surprisingly, this had almost no effect on performance. This was captured by the interaction between the different bottlenecks in our model.

Both experiments could also have been modeled using customized control strategies. For Experiment 1, this would not have posed any problems: Because the tasks do not have to be performed truly at the same time, but have to be alternated, the goal could have been changed on each step. Assuming this does not take time, it would result in the same model, yielding the same results. Thus, the only necessary control strategy would be changing the goal on each step. For Experiment 2, however, things become more complicated. Because the stories have to be processed concurrently with the other tasks, a control strategy would have to be devised that regulates this. Goals would have to be changed constantly when words are perceived, and switched back when word processing is finished. This would make for a very elaborate control strategy, in which the separate tasks depend on each other. In effect, the three tasks would have become one task. While this is possible, it would mean that our cognitive system performs tasks differently when they are combined with other tasks than when they are performed separately. This would necessitate multiple sets of rules and control strategies for each task. Not only does this seem illogical from a parsimony point of view, it would also mean that we have to learn new control strategies for each new combination of tasks.

Another solution would be to have general task switching rules, explored by Kieras et al. (2000) and Salvucci (2005). However, it turned out that these approaches still needed task specific control strategies to account for expert performance. The current approach, using no control strategies, but letting task properties determine how tasks are interleaved, seems therefore promising. The models accounted well for the human data, and the ease with which tasks can be added to a model seems to reflect the flexible way in which people can combine unrelated tasks.

Acknowledgments

Thanks to Max Jensch for running Experiment 2 and to Willie Lek, ID College, for providing the listening exams.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.

Borst, J. P., & Taatgen, N. A. (2007). The Costs of Multitasking in threaded cognition. *In Proc. of the 8th International Conf. on Cognitive Modelling*, 133-138.

Kieras, D. E., Meyer, D. E., Ballas, J., & Lauber, E. J. (2000). Modern computational perspectives on executive mental processes and cognitive control: Where to from here? In S. Monsell & J. Driver (Eds.), *Attention and Performance XVIII: Control of cognitive processes*.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cogn Sci, 29*, 375-419.

Salvucci, D. D. (2005). A multitasking general executive for compound continuous tasks. *Cogn Sci, 29*, 457-492.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychol Rev, 115*(1), 101-130.

Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009). Toward a Unified Theory of the Multitasking Continuum: From Concurrent Performance to Task Switching, Interruption, and Resumption. In *Proc. of CHI 2009*.

Spelke, E., Hirst, W., & Neisser, U. (1976). Skills of divided attention. *Cognition, 4*, 215-230.

Van Rij, J., Hendriks, P., Spenader, J., & Van Rijn, H. (2009). Modeling the Selective Effects of Slowed-Down Speech in Pronoun Comprehension. In *Proc. of the 3rd GALANA conference*. Cascadilla Press.