Concurrent multitasking

Nijboer, Menno

Publication date:
2016

CHAPTER 3

Contrasting Single and Multi-Component Working-Memory Systems in Dual Tasking

Menno Nijboer
Jelmer P. Borst
Hedderik van Rijn
Niels A. Taatgen

Where we develop a theory of how working memory is used during multitasking.

This chapter has been accepted pending minor revisions as:
Nijboer, M., Borst, J.P., Van Rijn, H., Taatgen, N.A. (in revision). Single and Multi-Component Working-Memory Systems in Dual Tasking. *Cognitive Psychology*

Contrasting Single and Multi-Component Working-Memory Systems in Dual Tasking

Abstract

Working memory can be a major source of interference in dual tasking. However, there is no consensus on whether this interference is the result of a single working-memory bottleneck, or of interactions between different working-memory components that together form a complete working-memory system. We report a behavioral and an fMRI dataset in which working-memory requirements are manipulated during multitasking. We show that a computational cognitive model that assumes a distributed version of working memory accounts for both behavioral and neuroimaging data better than a model that takes a more centralized approach. The model's working memory consists of an attentional focus, declarative memory, and a subvocalized rehearsal mechanism. Thus, the data and model favor an account where working-memory interference in dual tasking is the result of interactions between different resources that together form a working-memory system.

Introduction

Empirical work has shown that working-memory (WM) conflicts between tasks can severely impact overall performance during multitasking (Altmann & Trafton, 2002; Borst, Taatgen, & Van Rijn, 2010; Gray, Sims, Fu, & Schoelles, 2006; Jiang, 2004; Nijboer, Borst, Van Rijn, & Taatgen, 2014; Strayer, Cooper, & Turrill, 2013). However, psychological theories of multitasking do not typically address how working memory is used during concurrent task performance in any detail, and consequently, how working memory conflicts can affect multitasking performance. Existing work on multitasking has either described WM as a monolithic, single-component system (Altmann & Gray, 2000; Altmann & Trafton, 2002; Best & Lebiere, 2003; Borst et al., 2010; Fu et al., 2004; Marois & Ivanoff, 2005; Meyer & Kieras, 1997; Salvucci & Taatgen, 2008; Salvucci, 2001; Wickens, 2002; Zylberberg, Fernández Slezak, Roelfsema, Dehaene, & Sigman, 2010) or not at all (Aasman, 1995; Pashler, 1994; Salvucci, 2005; Schoppek, 2002). This is inconsistent with an increasing number of studies that propose differentiated WM mechanisms consisting of several subsystems, typically a focus-of-attention, an activation-based short-term memory, and modality-specific systems (Baddeley, 2000; Braver & Cohen, 2001; Cowan, 1988, 1995; Ericsson & Kintsch, 1995; Lewis-Peacock, Drysdale, Oberauer, & Postle, 2012; Oberauer, 2002; Unsworth & Engle, 2007; Vosskuhl, Huster, & Herrmann, 2015).

In the current paper, we investigated the role of WM in concurrent multitasking. In particular, we investigated whether a single-component WM is sufficient to explain observed interference patterns in dual-tasks or whether a multi-component WM system is required. We will discuss two experiments, of which we modeled behavioral and neuroimaging results in the shape of a cognitive computer model. We show that a multi-component view of WM that includes a focus of attention, activated short-term memory, and an active rehearsal loop is able to better capture WM use during multitasking than a monolithic WM. Furthermore, the particular WM components, and consequently the interference patterns, vary depending on the particular tasks.

Background

Task Interference

Classical evidence of multitasking costs comes from the Psychological Refractory Period (PRP; Telford, 1931). The PRP paradigm consists of two choice-reaction tasks, of which the stimuli are presented with a short stimulus onset asynchrony. The goal is to respond to the first stimulus (task A) before the second (task B). As the time between the onset of the first stimulus and the second stimulus becomes shorter, the reaction time (RT) for task B becomes longer. This phenomenon can be explained with the response-selection bottleneck model (RSB; Pashler, 1994). The RSB model distinguishes three phases in the component tasks of a dual-task scenario: perception, response selection, and response. The critical assumption is that perception and response can occur in parallel during a dual-task, but response selection can only be performed sequentially (Hazeltine, Ruthruff, & Remington, 2006; Marti, Sigman, & Dehaene, 2012; Pashler, 1994; Sigman & Dehaene, 2008). The RSB model has greatly influenced multitasking research, but it only addresses one particular type of task interference. It cannot, for example, explain interference effects caused by peripheral sources (Wu, Liu, Hallett, Zheng, & Chan, 2013) or memory (Hazeltine & Wifall, 2011; Strayer et al., 2013) or working memory. Working memory interference in particular can be detrimental for performance, as it does not only cause delays in task execution, but can also lead to the forgetting or misremembering of task critical information (Borst et al., 2010; Nijboer et al., 2014; Nijboer, Taatgen, Brands, Borst, & Van Rijn, 2013; D. Strayer et al., 2013). For example, Strayer and Johnston (2001) found that a complex phone conversation caused drivers to miss traffic signals more than twice as often.

Single or Multi-component Working Memory

Understanding how WM interference affects concurrent task performance requires a detailed model of the WM mechanisms themselves, as well as a good description of how these mechanisms are used within tasks. Recent WM research argues for a multi-component view of WM: for example, Unsworth and Engle (2007) show evidence for a focus of attention combined with an activated short-term memory to retrieve relevant information. Similarly Lewis-Peacock et al. (2012) distinguish the focus of attention from STM, while Vosskuhl et al. (2015) present evidence for a differentiation between WM and STM. These findings are consistent with modern theories of WM (Baddeley, 2000; Braver & Cohen, 2001; Oberauer, 2002; Cowan, 1995; Ericsson & Kintsch, 1995; Cowan, 1988). In these theories, WM subsystems include elements such as a focus-of-attention, an activation-based short-term memory, or modality-specific systems.

A multi-component WM means that task interference could occur in one or more of the mechanisms that together form WM. In addition, it means that tasks that require short-term retention or memory manipulation could use different strategies in terms of the components that are recruited to perform the task. For example, remembering the number of presentations of a certain item could be done by keeping this number in the focus of attention, or by keeping the count active in short term memory through rehearsal. However, in contrast to recent WM investigations, multitasking research has typically conceptualized WM as a single element (Altmann & Gray, 2000; Altmann & Trafton, 2002; Bradley & Best, 2003; Borst et al., 2010; Fu et al., 2004; Marois & Ivanoff, 2005; Meyer & Kieras, 1997; Salvucci, 2001; Salvucci & Taatgen, 2008; Zylberberg et al., 2010; Wickens, 2002). This makes it difficult to account for differences between tasks with regard to the employed WM strategy.

Task and Memory Strategies in Multitasking

Evidence for different strategic approaches to multitasking was previously shown by Howes, Lewis, and Vera (2009), who used cognitively bounded rationality (CBR) analysis to show that strategies used to perform the PRP task can differ per participant. Expanding on this, Janssen and Brumby (2015) showed that participants adapt the way they perform dual tasks to external factors, such as task characteristics and incentives. These studies show that tasks can be performed using different strategies at a very elementary level of cognition, which means that a cognitive component (i.e., working memory) can be implemented in different ways across tasks to accomplish similar goals. From a concurrent multitasking perspective this means that having two tasks that use different WM strategies could lead to different performance from tasks that use the same WM strategy, as interference between tasks would occur in different mechanisms or components.

The CBR approach is one way to avoid the issue that was pointed out by Roberts and Pashler (2000): fitting theories to data does not necessarily provide empirical support for the theory. The CBR analysis allows for assessing the strengths of the support by systematically searching for the optimal rational strategy as the initial model to be tested against the data, thereby constraining the space of implementations of the theory. Our approach to avoid the modeling pitfalls identified by Roberts and Pashler (2000) was to create model predictions of both behavioral and neuroimaging data *before* running the experiments. This way the model's behavior cannot be the result of overfitting, but is driven by the model's theoretical design. In particular, we created a model beforehand that could perform the paradigm by integrating existing task models within our own modeling framework. Subsequently, we made the predictions public before performing the actual experiment.

Cognitive Architectures for Multitasking

To evaluate our WM models we integrated them into an existing cognitive architecture. This allowed us to model our paradigm in a consistent way using a framework for cognition that has been extensively tested. As stated before, current models of multitasking are unspecific in how WM is simulated. This extends to cognitive architectures in general, such as ACT-R (Adaptive Control of Thought-Rational; Anderson, 2007) and EPIC (Executive-Process Interactive Control; Meyer & Kieras, 1997). For example, EPIC assumes that working memory is not a bottleneck at all when combining multiple tasks (Meyer & Kieras, 1997). Like all cognitive resources in EPIC, different processes can access working memory in parallel (Kieras, Meyer, Mueller, & Seymour, 1999). Instead, task interference in EPIC is the result of a task-specific control strategies created by the modeler. In ACT-R the working-memory function is performed by a combination of a one-element focus of attention and an unlimited, but decaying, long-term declarative memory. Both of these resources can process just a single chunk of information at a time. Salvucci and Taatgen (2008) extended ACT-R with threaded cognition, an account of how a serial architecture can account for multitasking. Instead of an explicit control strategy, threaded cognition adds a simple interleaving task-scheduling method to ACT-R: an applicable rule is picked from the task with the highest urgency, where the most urgent task is defined as the one that has least recently had a rule selected for execution. Furthermore, tasks must follow a specific etiquette: Each thread can use modules in a greedy manner, but has to release them politely. This means that a thread will use a required module as soon as it is available, and release the module as soon as its action has been completed. Given these constraints, task interference in ACT-R is the result of contention between tasks for cognitive resources, governed by threaded cognition. The memory components of EPIC and ACT-R we have discussed here show that cognitive architectures lend themselves well for monolithic WM systems, where WM content is easily accessed from a single resource. This may have led to a bias in existing multitasking models toward single-component WM systems.

Overview

In this work we investigated how working memory is used in multitasking. We focus on the complexity of, and control over, the working-memory system, and how this complexity affects interference between separate tasks. In particular, we constructed an a priori computational cognitive model to examine the interactions between two working-memory tasks and one peripheral task. The model tested whether a single-component WM is able to generate accurate predictions for behavioral and neural data. We tested the behavioral predictions of this model in Experiment 1. Next, we compared neural predictions of this model to an fMRI dataset previously presented in Nijboer, Borst, Van Rijn, and Taatgen (2014), here referred to as Experiment 2. As the single-component model could account for the fMRI data, we then compared a multi-component WM model against both the behavioral and neuroimaging data of Experiment 2. We also test the generalizability of this model by fitting it to the behavioral data of Experiment 1. The degree to which the models fit the data and generalized over datasets was used to determine whether a multi-component view of WM could better explain multitasking interference data than a single-component implementation.

In the remainder of this paper we will first explain the details of the experimental paradigm and the modeling approach. We will then proceed to discuss the a priori model, followed by the comparisons of the model against data from Experiment 1 and 2. Next, we elaborate on the changes to the model required for a better fit of neuroimaging data. We finish with a general discussion of the implications of the model for the conceptualization of WM in multitasking.

Paradigm

Working memory resources involved in multitasking were investigated using three different tasks: n-back, tracking, and tone-counting. During n-back, a series of letters is presented on screen. The participant is asked to indicate for each letter if it was the same as, or different from *n* letters ago. In terms of resources, the n-back task uses motor, visual, and WM resources (Juvina & Taatgen, 2007; Owen, McMillan, Laird, &

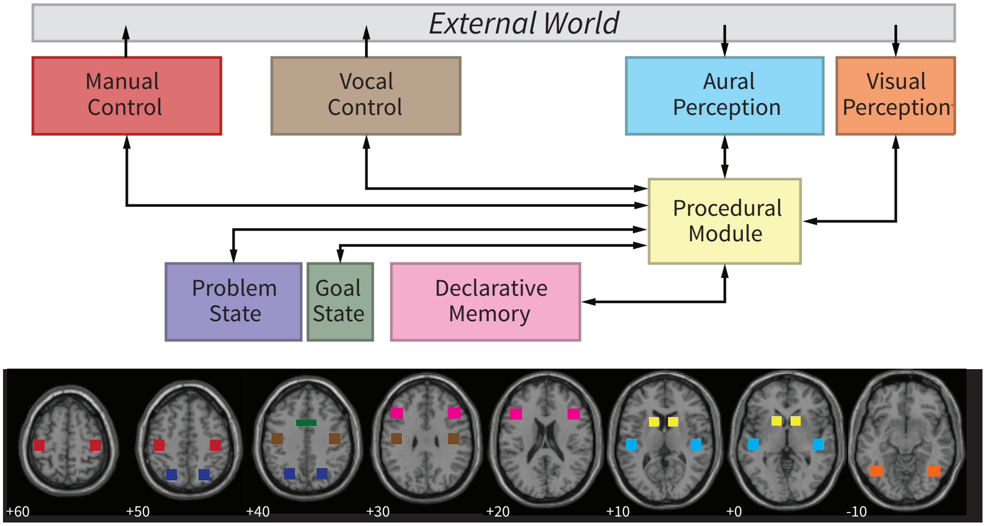

Figure 3-1. The ACT-R cognitive architecture and its mapping on brain regions (Borst & Anderson, 2015). Numbers indicate the z-coordinate of each slice (MNI coordinates). Length and position of the boxes corresponds with the location of that module in the neuroimaging slices.

Bullmore, 2005). In this paradigm we used $n = 2$, and in the remainder of the paper we will refer to this task as the 2-back task. In the tracking task, the goal was to keep a cursor close to a randomly moving target using a left and right button (Martin-Emerson & Wickens, 1992). Two lines that flanked the target signaled the maximum allowed error. The tracking task was predicted to use visual and motor resources, making it the only task not to use memory resources. Finally, during tone-counting a random series of low and high pitch tones were played at different intervals. The goal was to only count the high pitch tones. After a trial, participants were asked to enter the total count. The tone-counting task was expected to use aural and WM resources during the trial, and the motor resource during the response phase.

We designed a set of six conditions: the 3 single-task and all 3 possible dual-task conditions (i.e. A, B, C; AB, AC, and BC). This setup has two desirable properties. Foremost, cognitive resources can be examined through interactions between tasks, as different task combinations result in different resource conflicts. Furthermore, the possible modeling space is more constrained: models need to capture the component task behavior as well as the interactions with both other component tasks. Detailed methods will be reported below.

Modeling Approach

We based our models[1] on threaded cognition (Salvucci & Taatgen, 2008), which itself is an extension of the ACT-R cognitive architecture (Anderson, 2007). ACT-R is a

[1] The models in this work can be found at http://www.mennonijboer.nl/

general psychological theory that has been instantiated as a simulation environment in which computational cognitive models can be developed. These models allow for precise tests of the theory by forcing one to formally implement all theoretical assumptions.

ACT-R contains modules for cognitive and peripheral functions, shown in Figure 3-1. Procedural memory is used to coordinate actions between these modules, using a set of if-then rules. A rule can for instance be 'if the visual module contains a letter, then store this letter in declarative memory'. Execution of a rule takes 50 ms, and rules are executed serially. In other words, the procedural module creates a bottleneck (Anderson, Taatgen, & Byrne, 2005; Byrne & Anderson, 2001). While access to the remaining modules is serial as well, different modules can process requests in parallel. Threaded cognition extends the procedural system by adding the possibility to interleave the execution of production rules from different tasks: The procedural system will pick an applicable rule from the task with the highest urgency. The most urgent task is defined as the one that has least recently had a rule picked for execution. If no rule of the most urgent task matches, the next most urgent task is selected. To stop tasks from keeping modules occupied indefinitely, they are required to release a module as soon as its action has been completed. This makes it possible to have concurrently performed tasks: the production system works as a dispatcher, sending requests to various different modules and waiting for those requests to complete in order to continue. So as long as two tasks share few or no resources, the procedural bottleneck does not need to impede execution of either task, as the modules work on the tasks independently.

While ACT-R has no dedicated WM system, it has two modules that can be used as part of a WM strategy: declarative (long-term) memory and the problem state. Although the capacity of declarative memory is essentially unlimited, the chance of being able to retrieve an item decreases over time. This is implemented by giving each item an activation value, which is a numerical expression of its strength in memory. This activation value decays over time (Anderson, 2007), but increases when the associated fact is retrieved: Items that have been used more often or more recently will have a higher activation. The activation value is used during the retrieval phase, where it determines how long the retrieval will take (a higher activation value results in shorter retrievals), and whether retrieval is possible at all (a chunk can only be retrieved if its activation value exceeds a predefined retrieval threshold value).

The problem state is a buffer that can contain a single piece of intermediate information used by a task (Borst et al., 2010). Thus, it is similar to the 'focus of attention' concept (McElree, 2001; Oberauer, 2009). For example, when presented with a 'solve-for-x' equation with two steps, an intermediate solution can be stored in the problem state. This partial solution can then be used to calculate the final answer. While information in the problem state is accessed without a time cost, replacing it takes a relatively long time (the ACT-R default value is 200 ms, see Anderson, 2007, which has functioned well to account for multitasking interference in the past, e.g., Borst et al., 2010; Borst, Taatgen, & Van Rijn, 2015). When a problem state is replaced, the old state is stored as an item in declarative memory. The problem state is therefore also the place where new declarative knowledge can be built. In general, there are more methods to support a working memory function besides declarative memory and the problem state: a subvocalized rehearsal loop, using hands and fingers, and, by extension, pen and paper.

ACT-R can be used to make predictions of the fMRI BOLD (Blood-Oxygen-Level Dependent) signal (Anderson, 2007), as the different ACT-R modules have been mapped to specific brain regions. These regions have been developed from literature (for an overview see Anderson, Fincham, Qin, & Stocco, 2008) and refined through a data-driven model-based approach (Borst & Anderson, 2013; Borst, Nijboer, Taatgen, Van Rijn, & Anderson, 2015). The regions that were used for the current analysis are from the data-driven mapping from Borst et al. (2015). They selected the best fitting regions through a meta-analysis of five very different tasks, which were modeled independently from these regions. In particular interest to the current work are the regions corresponding to the problem state and declarative memory, which are part of the fronto-parietal network that is frequently associated with working memory, cognitive control, and attentional selection (Cole & Schneider, 2007; Dosenbach et al., 2007). Specifically, Borst and Anderson (2013) found that a region of the intraparietal sulcus correlated exclusively with the problem state, while the declarative module correlated exclusively with a region of the inferior frontal gyrus. This is consistent with earlier literature regarding these areas, which implicate the intraparietal sulcus in memory retrievals and updates (Bunge, Hazeltine, Scanlon, Rosen, & Gabrieli, 2002; Olesen, Westerberg, & Klingberg, 2004; Sohn et al., 2005; Wager & Smith, 2003), and prefrontal region in episodic-memory retrievals and working memory activity (Buckner & Wheeler, 2001; Cabeza et al., 2003; Dobbins & Wagner, 2005; Smith & Jonides, 1998).

Activation in the ACT-R modules can be compared to fMRI activity in the corresponding brain regions, making it possible to constrain models not only with behavioral data, but also with fMRI data (Anderson, 2007; Anderson et al., 2011). To predict the BOLD response, the activity of each ACT-R module is convolved with a hemodynamic response function that represents the BOLD response (in this paper we use the HRF proposed by SPM, which is a mix of two gamma functions; Friston, Ashburner, Kiebel, Nichols, & Penny, 2007). The result of this convolution is the activity of the module over time, expressed as a BOLD curve. Furthermore, by calculating the area under this curve we obtain the total activation for that module during a certain timespan. For a more detailed overview of this technique, see Borst and Anderson (2015).

The goal of this work is to explore the level of complexity of working memory that is required to explain interference patterns observed during concurrent task performance. To accomplish this, we created an a priori model to generate behavioral and neuronal predictions of the paradigm. The a priori model assumed that WM

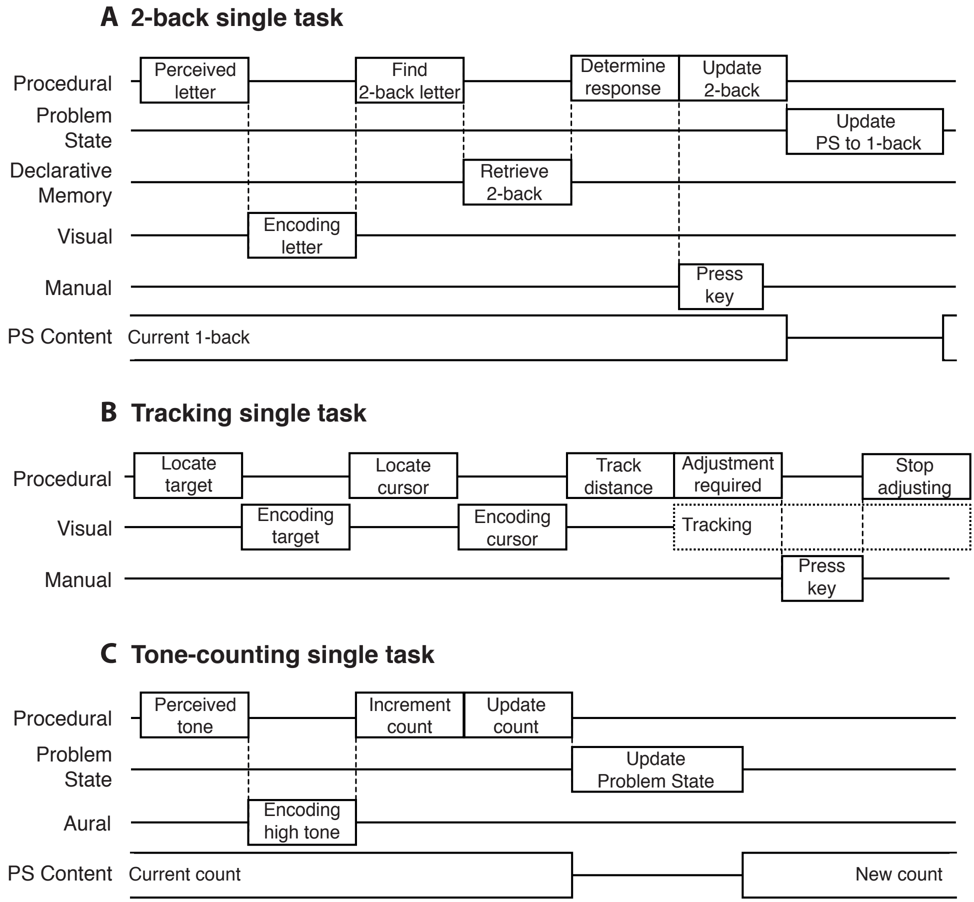

Figure 3-2. Process timelines for each single-task in Model 1-pre. *Panel A:* Seeing a new letter during the 2-back task. *Panel B:* The start of the tracking task. *Panel C:* Hearing a new high tone during tone counting. PS Content = Problem State content.

interference is caused by a single bottleneck: the problem state resource (cf. Borst et al 2010; Borst et al., 2013). These predictions were compared against experimentally obtained data. The model was used to construct a second model that uses a working memory system with multiple potential bottlenecks. The two distinct model implementations were examined to investigate how working-memory interference occurs dual task situations, and what mechanisms control interleaving of working-memory processes in concurrent dual tasking.

Model 1

The general modeling approach that we have taken is to design models for the individual tasks, and then combine these models without the need to add anything specific for multitasking coordination (see Salvucci & Taatgen, 2008). In designing the components for Model 1, we tried to be as parsimonious as possible: WM mechanisms were limited to the problem-state and declarative-memory modules. Furthermore, declarative memory is only used when the memory load of a task exceeded a single item. Given that threaded cognition does not specify how the problem state can be shared by two tasks, we used the same strategy as Borst et al. (2010): if a task needs the problem state resource, it can take it (greedy), even if it contains information for another task. This means that from the perspective of an individual task, its problem state may suddenly be missing. If that is the case, it will need to recover its state from declarative memory (or from other sources, see Borst, Buwalda, Van Rijn, & Taatgen, 2013).

The basis of the 2-back task model was the low-control model by Juvina and Taatgen (2007). It perceives the stream of letters belonging to the 2-back task, and maintains a problem state that contains the previously attended letter and a reference to the 2-back letter in declarative memory. This is a simplified version of the "time tags" used by Juvina and Taatgen (2007): Their low-control model used the time of stimulus encoding to determine the recency of a letter in declarative memory. This age indicator can be used to judge if the retrieved letter is the target n-back. The original implementation is quite sophisticated as it was used to model learning effects. This is not of interest in the current work, so in our model, the reference in the problem state is simply used to search for the letter in declarative memory that has the appropriate expected age. As such, the problem state represents the head of a linked list of every letter stimulus observed by the model, with all other letters stored in declarative memory. When the model attends a new letter, the 2-back letter is retrieved from declarative memory and compared against the new letter. The model then gives a response with its left hand for 'same' or 'different', depending on the retrieved letter. If the 2-back letter cannot be retrieved because its activation fell below the retrieval threshold, the model guesses either 'same' or 'different' with equal likelihood. Following the response, the problem state is replaced with a new state containing the current letter and a reference to the old problem state. This causes the old state, which contains the 1-back letter, to move into declarative memory. Figure 3-2A shows a timeline of the model when it perceives a new letter. It should be noted that the Juvina and Taatgen (2007) model does not use the problem state, but relies purely on declarative memory. We changed the model to rely primarily on the problem state, in line with the more recent findings pointing to posterior-parietal areas involved in representing a single-component WM system. Thus, declarative memory in this case is used as a general information store that requires specific cues to retrieve information from.

The 2D tracking model of Chavez and Salvucci (2003) was the inspiration for our tracking model. Some adaptations were required to make the original model suitable for our paradigm, where the tracking was 1D and the input was a keyboard instead of a joystick. The model first perceives the positions of the cursor and the moving target dot (Figure 3-2B). While tracking these positions, the distance between the

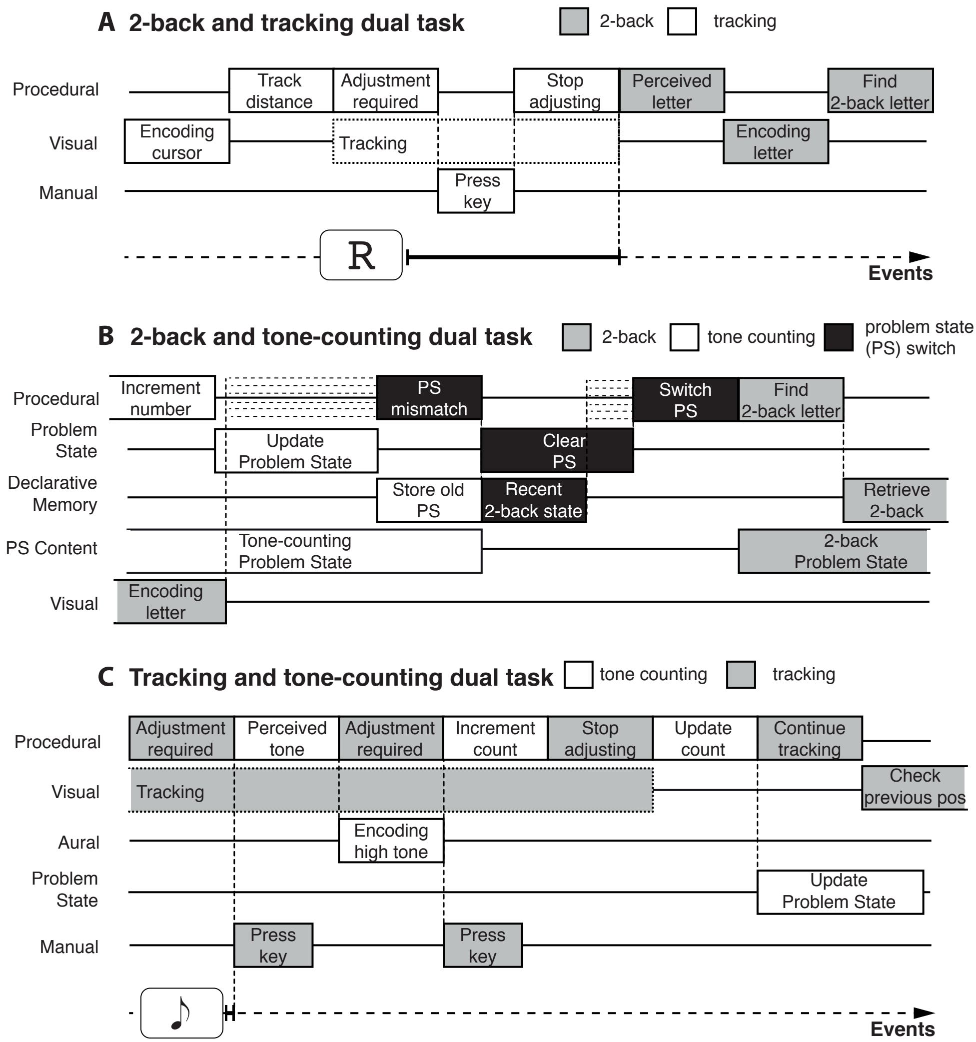

Figure 3-3. Process timelines for dual tasks in Model 1-pre. *Panel A:* Tracking causes a delayed response to a newly presented letter of the 2-back task. *Panel B:* Performing 2-back and tone counting concurrently results in problem state switching. *Panel C:* Tracking and tone counting can be performed with very little interference as the production rules can be interleaved almost optimally.

target and cursor is observed. If this distance exceeds 5 pixels, the model presses the key required to reposition the cursor on top of the target, using its right hand. Movements of the cursor will be rapid when it is far removed from the target (more than 10 pixels), but slow when it is close to the target, as the model will re-evaluate the distance between cursor and target when they are close to each other. This allows the model to make precise tracking adjustments. The tracking model uses visual and

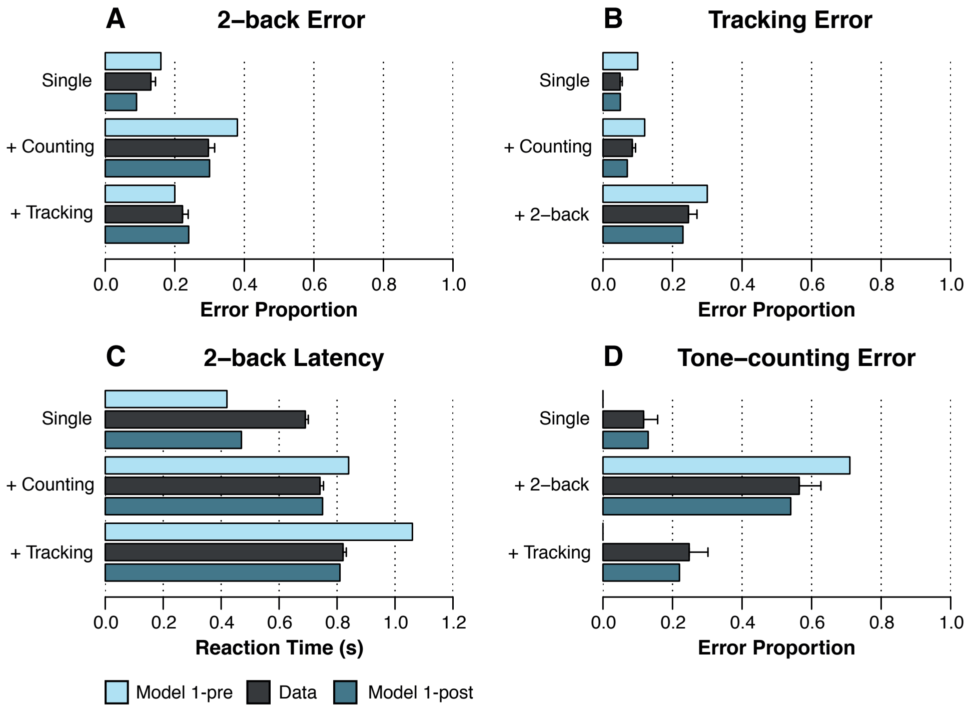

Figure 3-4. Comparison of model 1 with behavioral data of Experiment 1. Light-blue bars belong to the Model 1-pre prediction made before running the experiment. Dark-grey bars represent the averages of the collected data, and the dark-blue bars are the final fit of Model 1-post to the data. All model data was averaged over 1500 runs. See the text for the descriptions of Model 1-pre and 1-post.

manual resources, but neither the problem state nor declarative memory.

The tone-counting task model was based on the counting task model from an ACT-R tutorial (see http://act-r.psy.cmu.edu/software/). The model stores the current count in a problem state. After a high tone is perceived, the model increments the current count from *n* to *n*+1. This can be seen in Figure 3-2C. When this process is complete, the problem state is updated with the new count value. Low tones are ignored after they have been identified. When the trial has ended and the count input screen appears, the task model uses the appropriate hand to input the count that is currently in the problem state.

In Model 1 interference during multitasking arises from two sources. The first source is visual attention: When 2-back is combined with tracking, the visual module has to divide its time between the letters on the left half of the screen and the tracking target on the right side (Figure 3-3A). While both tasks use the manual module, interference is limited as the two tasks use different hands, and the 2-back task only requires periodic input. As presented in Figure 3-3B, the second source of interference is memory related: Both the 2-back and tone-counting threads use

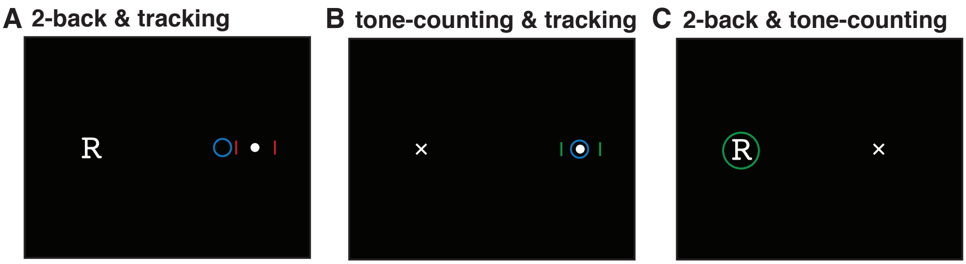

Figure 3-5. The possible dual-task screens as presented to the participants. *Panel A:* The left side of the screen shows one of the letters presented during the 2-back task, with the right side shows inaccurate tracking during the tracking task: the vertical lines indicate the maximum allowed distance between the cursor (blue circle) and the target (white dot). They are colored red when the cursor is outside this boundary. *Panel B:* Tone-counting is represented on the left side as a fixation cross and tracking is again on the right, with a correct cursor position. *Panel C:* The left side contains 2-back, with the green circle indicating a correct response, and to the right is the tone-counting task.

declarative memory and the problem state. As switching out problem states has been found to take a significant amount of time (estimated at 200 ms; see Anderson, 2007; Borst et al., 2010; Borst et al., 2013), this should lead to a larger decrease in performance than the visual interference. This is reflected in the a priori prediction[2] of the model, which can be seen in Figure 3-4 (light-blue bars), and is referred to as Model 1-pre. Model 1-pre predicts that the single tasks show the best performance. When counting is combined with tracking, the tracking task is predicted to shows a slight increase in errors, while in the tone-counting task accuracy is predicted to remain maximal: as Figure 3-3C shows, the tasks can be interleaved almost perfectly. Performing 2-back together with tracking results in a noticeable predicted increase in errors for both tasks. However, 2-back with tone counting clearly leads to the largest predicted increase in errors for both tasks. To test these predictions, we conducted Experiment 1.

Experiment 1

Participants

A total of 29 participants performed Experiment 1 (19 female, Mage = 23.0, age range: 19-30). The Ethical Committee Psychology of the University of Groningen granted approval for the experiment, and written informed consent was obtained from all participants. Participants received €10 upon completion. All participants had normal or corrected-to-normal vision. We excluded one participant who did not follow the instructions correctly, and three others for performing the 2-back task at

[2] This prediction was generated before the experiment was performed; it was posted on the ACT-R mailing list on October 29, 2012.

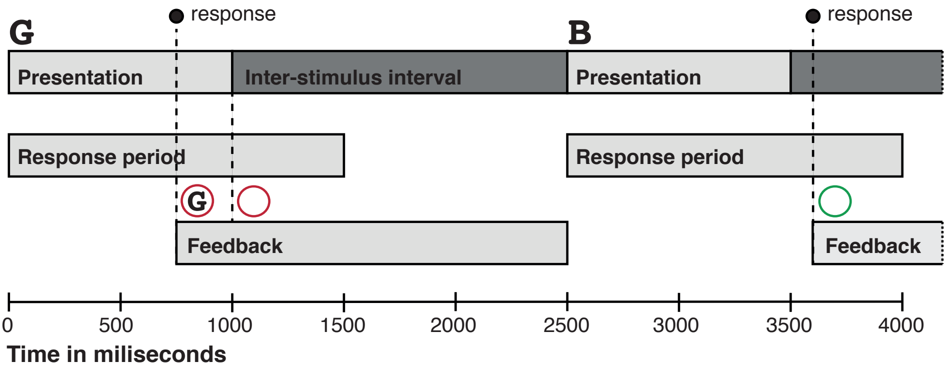

Figure 3-6. Timeline for the n-back task. Two different situations are presented: a response before the ISI, and one after.

chance level, leaving 25 participants for analysis.

Design

Participants started with a practice block in which they performed each single-task condition twice. The practice block was followed by 62 trials distributed over all six conditions. These 62 trials were split in two blocks of 31 trials, with a short break in between. Of each 31 trials, 1 trial consisted of the presentation of just a fixation point. During this fixation-only trial, participants did not need to perform any task. To minimize learning effects the conditions were presented in a pseudo-random order: each condition appeared every six trials, resulting in 10 trials per condition. The fixation trial was inserted into the trial order at a random place.

Task performance was measured for each task. For the 2-back we measured the number of correct responses to the letter stimuli, as well as the reaction time, which was defined as the time between the appearance of a letter and the key press response. Tracking error was defined as the percentage of a trial in which the cursor was not located between the two lines that flanked the target. Finally, for tone counting we took the proportion of correct counts given at the end of all trials combined.

Materials and Procedure

During the instructions participants were told not to prioritize any task over the others, and that they should perform both tasks to the best of their ability. Next they were given short descriptions of each task and the possible conditions present in the paradigm.

The 2-back task (Figure 3-5A and 3-5C) consisted of 12 letters that were presented for 1000 ms with a blank screen presented between each letter that lasted for 1500 ms. The response window started at the presentation of the letter, and lasted up to 1500 ms after the onset of the presentation. Participants were instructed to press E to indicate that the letter was the same as the letter 2-back, or R if it was not. After the response (or expiration of the response period) a green circle was shown for a correct choice, and a red circle for an incorrect choice or a too late response. Feedback remained visible until the next letter appeared. A detailed timeline of the 2-back task can be seen in Figure 3-6.

In the tracking task (Figure 3-5A and 3-5B), participants had to track a white circle (the target) from left to right with a small blue circle (the cursor). The position of the cursor could be moved to the left or right using the U and I keys, respectively. On either side of the target circle was a line that indicated the maximum allowed distance between the cursor and the target. When the cursor position fell outside these two lines, their color changed from green to red. Perturbation of the target was implemented in the same way as in Martin-Emerson and Wickens (1992): a summation of different sinusoid terms that resulted in an unpredictable curve, which can be used as perturbation for the target. Using this function the chance of a tracking error increased by 20% every second, on average.

Finally, in the tone-counting task (Figure 3-5B and 3-5C) participants were presented with 20 tones. The interval between tones was randomized between 0.5 and 1.5 seconds. The low tone had a pitch of 260 Hz, while the high tone had a pitch of 490 Hz. The tones lasted for 200 milliseconds. The number and order of high tones was randomized; out of the 20 tones, between 10 and 17 tones were high pitched. After the trial a response screen was presented, giving participants 9.5 s to provide a response. This response was generated using two keys, required to increment the displayed count number. The keys used for this were U and I if tone counting was combined with 2-back, and E and R in all other conditions: E/U increased the number of tens, while R/I increased the number of ones. The numbers loop around, so 0 would come after 9. After 10 seconds passed the count on the screen would be submitted as the final answer. After the response, feedback indicating whether the response was correct or not was presented for 500 ms.

Participants were instructed to position their left index and middle-finger over the R and E key, while keeping their right index and middle-finger over the U and I keys. This was done in order to minimize input differences with Experiment 2, which was performed in an MRI machine.

Results

Data

Unless mentioned otherwise, all *p*-values of the main effects are from analyses of variance performed on linear mixed-effects models (LME). Within the models the condition was used as fixed effect, with random slopes and intercepts for participants. Accuracy data were modeled using binomial LMEs. The *p*-values of individual comparisons between conditions were computed by performing a Tukey honest significant difference test on each LME. All models were constructed and analyzed in R (3.0.2) with the lme4 package (1.0-5). All error bars in figures depict the upper half

Comparisons	2-back error rate			2-back latency		
	z	β	p	z	β	p
2b vs. (2b + Co)	9.36	1.32	< .001	-2.90	-0.05	.010
2b vs. (2b + Tr)	6.54	0.89	< .001	-7.02	-0.13	< .001
(2b + Tr) vs. (2b + Co)	5.49	0.43	<. 001	6.00	0.08	< .001
	Tone-counting error rate			**Tracking error rate**		
	z	β	p	z	β	p
Co vs. (Co + Tr)	3.09	0.90	.006			
Co vs. (2b + Co)	8.94	2.57	< .001			
(2b + Co) vs. (Co + Tr)	-7.50	-1.69	< .001			
Tr vs. (Co + Tr)				7.72	0.55	< .001
Tr vs. (2b + Tr)				11.73	1.78	< .001
(2b + Tr) vs. (Co + Tr)				-9.27	-1.23	< .001

Table 3-1. Between-conditions comparisons of the reaction times and accuracy data of Experiment 1. Abbreviations: 2b = 2-back, Co = tone counting, and Tr = tracking. Comparisons were computed by applying a Tukey honest significant difference on the linear mixed-effects models. The resulting z values, p values, and estimates (β) are reported.

of 95% confidence intervals for the mean.

The dark-grey bars in Figure 3-4 show the averaged participant data, and Table 3-1 shows that all differences between conditions were significant. In Figure 3-4A the 2-back error is defined as the proportion of incorrect responses per trial. This subfigure shows that having to perform a second task affected the error rate on the 2-back task. Furthermore, the increase in error rate depended on the properties of the second task. On average, the proportion of errors increased by 9% when 2-back was combined with tracking, but by 17% when combined with tone counting. The 2-back latency data presented in Figure 3-4C shows a pattern that differs from the error data. While single-task 2-back has the lowest RT on average, it is the 2-back plus tracking condition that has the highest RTs, not 2-back plus tone counting: a difference that was significant.

Examining Figure 3-4B, tracking error appears to be higher in both dual-task conditions. The 2-back plus tracking conditions has the largest increase in average error, as well as the largest variance in error across participants. While the difference between the single-task and tracking plus tone-counting appears small, all comparisons were significant.

Tone counting yielded significantly different error proportions depending on the condition. As seen in Figure 3-4D, the error increased by 13% in the tone-counting plus tracking condition, but by 45% in the 2-back plus tone-counting condition; almost four times the single-task error.

The behavioral data indicates that dual-task costs depend strongly on the particular combination of tasks. When performed simultaneously, tasks that are expected to

Measurement	Model 1-pre		Model 1-post	
	R^2	RMSE	R^2	RMSE
2-back error	.84	.05	.97	.03
2-back latency	.91	.22	.78	.13
Tracking error	.99	.05	1.0	.01
Tone-counting error	.92	.18	.99	.02

Table 3-2. Measurement fit for the prediction and fit model.

have a significant overlap in resource use lead to more frequent errors than tasks that have a small degree of overlap. Behavioral patterns of a task are not necessarily the same in both accuracy and latency, as seen in the 2-back results. This indicates different sources of task interference, most likely involving peripheral and central cognitive resources.

Model 1-post

When we compare the a priori prediction to the observed data (Figure 3-4, light-blue vs. dark-grey bars), we find that the prediction has a good qualitative fit to the data (Table 3-2), taking into account that these predictions were generated before collecting the data. The only pattern that was not entirely predicted was that of the tone-counting errors. Model 1-pre made no mistakes in the single task and tone counting plus tracking conditions, whereas participants did make some mistakes. Overall, the model performed slightly worse than participants did.

In order to assess what changes were required to the model to better fit the data, we made several small changes that together resulted in Model 1-post: To accurately capture tone-counting performance, we introduced the possibility of making a mistake during counting. Anecdotal evidence suggested that participants were sometimes mistaking low tones for high tones. Therefore, a new production rule was added to the tone-counting thread that sometimes counted a low tone as a high tone, instead of ignoring it. This rule competes with the correct rule for handling low tones, but has a much lower utility. The effect of this rule is a lower performance ceiling for tone counting across all conditions. Finally, model parameters were adjusted (Table 3-3) to optimize the quantitative fit: The retrieval threshold determines the minimal amount of activation required to retrieve a chunk. The activation noise is the amount of random activation added or subtracted to individual chunk activations. The latency-scaling factor scales the retrieval times of chunks based on their activation. Finally, the sound-decay time determines how long sounds observed by the aural module are available for further processing before they are discarded.

Discussion

The general conclusion that can be drawn from Model 1 and the participant data is

Parameter	Model 1-pre	Model 1-post
Retrieval threshold τ	0.1	0.03
Activation noise ε_i	0.1	0.05
Latency-scaling factor F	0.1	0.14
Sound-decay time *Sd*	3.0	0.4

Table 3-3. ACT-R parameters for Model 1-pre and Model 1-post.

that interference depends on the overlap in resources used by the individual tasks, which is in agreement with earlier multitasking research (Nijboer et al., 2013, 2014; Salvucci & Taatgen, 2008). Looking at working memory specifically, the interference effects between 2-back and tone counting indicates that the problem state can be a source of interference between tasks. Consequently, the model indicates that WM interference in dual tasking can be described by a single bottleneck. Furthermore, the observed WM interference is more substantial than interference in other areas: Visual interference as produced when 2-back is combined with tracking led to lower accuracy loss than WM interference between 2-back and tone counting. When there were no overlapping resources overall performance was highest, as evident from the tone-counting plus tracking results.

A different explanation for the results is that the 2-back task was simply more difficult than the other two tasks, causing the observed task interactions. But this possibility runs counter to both the a priori model prediction and the data: while 2-back plus tone-counting showed the highest error rate increase, it showed only an intermediate reaction time increase. Instead, 2-back plus tracking showed the highest reaction time increase: If the 2-back difficulty itself were the source of the interaction pattern, one would expect the same pattern across conditions for both error rate and reaction times.

Model 1-post shows that a very close match to the participant data is possible. As the conflict between WM tasks in this model is based on the assumption that interference occurs through problem-state contention, it strengthens the argument for the problem state as a WM interference source. However, the current model is based on behavioral results, which give us limited insight into the processes occurring in the brain. Functional MRI provides an additional stream of data that can be used to further constrain the modeling space of the paradigm. In particular, because ACT-R modules have neural correlates (Anderson, 2007; Borst et al., 2015), the data can be used to evaluate the cognitive resources that are used during a particular task. Therefore, we compared Model 1 to fMRI data previously acquired by Nijboer et al. (2014). While the investigation of Nijboer et al. focused on a different area of multitasking, the data was found suitable for comparison against Model 1. It should also be noted that Model 1 was developed before the fMRI study was performed, and is therefore Model 1-post provides true predictions of the results in the fMRI study.

Experiment 2

Paradigm

Experiment 2 was previously reported in Nijboer et al. (2014), where we investigated whether neural activity found in dual tasks is simply the summation of the single-task activity, or includes an additional, dual-task specific, component. Nijboer et al. (2014) found that dual-task activation was a summation of the single-task activity, and that dual-task interference could be predicted from the overlap in active brain regions in single-task conditions, however, no formal model was provided in Nijboer et al. (2014). The paradigm was identical to that of Experiment 1, except that participants were placed in an fMRI scanner to collect neuroimaging data in addition to the behavioral data.

Participants & Procedure

A total of 20 right-handed adults participated in Experiment 2. While the paradigm remained unchanged from Experiment 1, several changes had to be made for fMRI compatibility. Instead of a regular computer screen, the experiment was projected onto a mirror mounted to the fMRI head coil. Furthermore, participant input was not gathered with a keyboard, but with a four-button response box that was placed on the abdomen. Before the scanning session, participants performed a 20-minute practice session with the paradigm outside of the scanner. In a second session, participants performed six 10-minute blocks while in the scanner. Each trial lasted 30 seconds, with an additional 10 seconds to enter the tone count if needed. Functional MRI data were acquired with a TR of 2 seconds, which resulted in 15 scans per trial. For a more detailed overview of the experimental setup we refer to Nijboer et al. (2014).

Model 2-pre

For Experiment 2 we also generated a priori predictions. To this end, we adapted Model 1-post to the setup of Experiment 2: in terms of cognitive processes Model 1-post and Model 2-pre are identical. The behavioral predictions of Model 2-pre are presented as light-green bars in Figure 3-7. The model predicts large interference between 2-back and tone counting (4A and 4D), moderate interference between 2-back and tracking (4A and 4B), and little interference between tone counting and tracking (4B and 4D).

In Figure 3-8, the fMRI predictions of the model are presented[3]. The length of each bar represents the total activity during a trial of the problem state module (Figure 3-8A) and declarative memory (Figure 3-8B). It should be noted that the scale of the model results is not meaningful: the BOLD signal is a relative measure, and thus we can only compare the activation *patterns*. The main prediction of Model 2-pre

[3] This prediction was posted on the ACT-R mailing list on January 16, 2013.

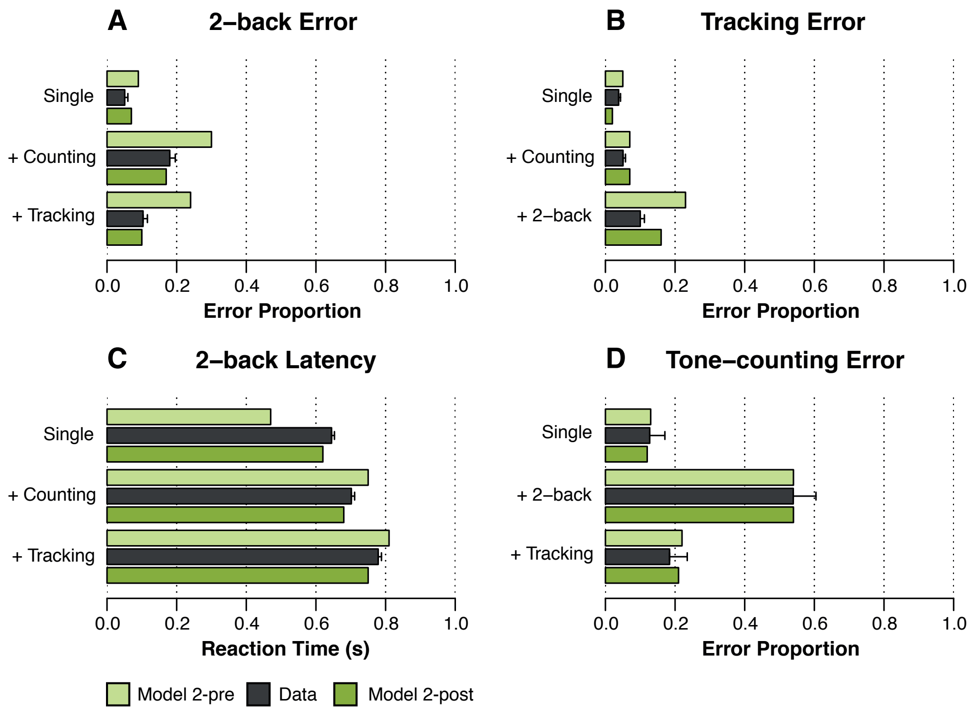

Figure 3-7. Comparison of Model 2-pre and Model 2-post against behavioral data of Experiment 2. Light-green bars belong to Model 2-pre. Dark-green bars belong to Model 2-post of Experiment 1. Dark-grey bars represent the averages of the collected data. All model data was averaged over 1500 runs. See the text for a description of Model 2-pre and Model 2-post.

is a strong over-additive effect in both the problem state and declarative memory when tone-counting and 2-back are performed concurrently. This is the result of contention for the problem-state resource: as task rules are interleaved, the problem state has to be switched to te state of the second task whenever a rule of that second task is executed. Hence, the problem-state resource will be very active when two tasks need to use it. Because the problem state that needs to be switched in is retrieved from declarative memory, we also see a large over-additive effect in that resource.

Results

Behavioral Results

As seen in Figure 3-7, the data of Experiment 2 mirrors the results of Experiment 1[4]. Performance was slightly better than in Experiment 1, which might have been caused

[4] Statistical details can be found in Chapter 2 or Nijboer et al. (2014). All behavioral measures showed significant main effects of condition. All individual comparisons were significant, with the exception of tone counting versus tone counting plus tracking.

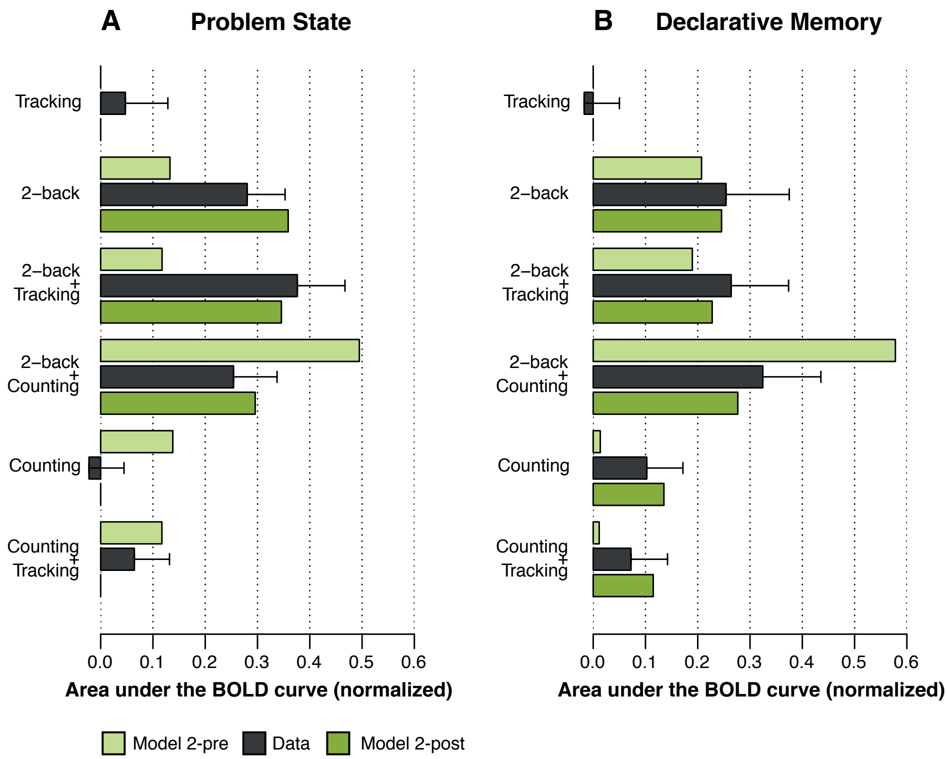

Figure 3-8. Experiment 2 neuroimaging data compared against Model 2-pre and Model 2-post. The bars represent the area under the curve for each ROI and condition combination. Light-green bars: Model 2-pre. Dark-green bars: Model 2-post. Dark-grey bars: participant fMRI data. The AUC was normalized within each ROI or module to make values comparable between model and fMRI data. All model data was averaged over 30 runs. See the text for a description of Model 2-pre and Model 2-post.

by participants having an additional practice session in Experiment 2. Despite the differences in preparation and performing the experiment in the scanner, the only qualitative difference with the data of the previous experiment is that the difference between the tone-counting and tone-counting plus tracking conditions is no longer significant (see Nijboer et al., 2014).

Neuroimaging Results

The dark-grey bars in Figure 3-8 shows the summed BOLD response during a trial, averaged over all participants. These BOLD data show that the problem state is active primarily in conditions that involve the 2-back task, as the tone counting task showed hardly any problem state activity at all. Much like the problem state region, the declarative memory area shows strong activation for the 2-back task. However, there was also considerable activity in the remaining conditions that included tone counting.

Model 2-pre Evaluation

Figure 3-7 indicates that the behavioral predictions of Model 2-pre qualitatively fit the data – as expected, given that the task was the same. In contrast, the Model 2-pre BOLD predictions were not supported by the data: no over-additive effects were found. The problem state shows that the 2-back plus tone counting condition had the lowest activity for all conditions involving 2-back, instead of the highest. The prediction for declarative memory represented the data marginally better. However, the low amount of predicted activity during tone counting does not fit with the neuroimaging data, and the predicted over-additive effect was not found.

Attempts to fit the parameters of Model 2-pre to better represent the data did not change the overall fit in a quantitative or qualitative way. Thus, Model 2-pre was not able to predict the BOLD signal for memory resources, despite the good behavioral fit for Experiments 1 and 2. The main difference lies in the tone-counting task and the interaction between tone counting and 2-back: In contrast to the model, the neuroimaging data indicated that the problem state is not used during tone counting. Furthermore, the data indicate that participants use declarative memory more during tone counting than our model predicted. Taken together, this indicates that participants used a different memory strategy to keep track of the current tone count than the model predicted.

Model 2-post

The fMRI data indicated that the memory strategies applied to the tasks in the paradigm are different from the monolithic system hypothesized in Model 1. For that reason Model 2-post uses a multi-component WM. Consequently, dual-task interference can originate from one or more systems that play a role in information retention. In particular, both 2-back and tone counting now use declarative memory and subvocalized rehearsal. This means that in Model 2-post, the 2-back thread uses the problem state in a slightly different manner to allow for interference in declarative memory: as shown in Figure 3-9A, the most recent letter is no longer explicitly stored in the problem state. Instead, all previously seen letters are kept in declarative memory, and the problem state itself contains only a reference to the chunk in declarative memory that contains the most recently presented letter. As in the Model 1-pre 2-back model, this is similar to the "time tags" used by Juvina and Taatgen (2007) in their n-back model. Retrieving this letter requires passing the reference to declarative memory, which will retrieve the associated chunk. The 2-back letter is determined by retrieving the two most active letter chunks in declarative memory. The chunk that is not equal to the reference stored in the problem state will typically be the 2-back (but possibly an older letter). Hence, the model iterates through the letters that are most active in memory, and picks the 2-back letter by excluding the one that has been tagged as the 1-back. After judging the most recently seen letter it can be considered the 1-back, and this new information is stored in declarative memory. To

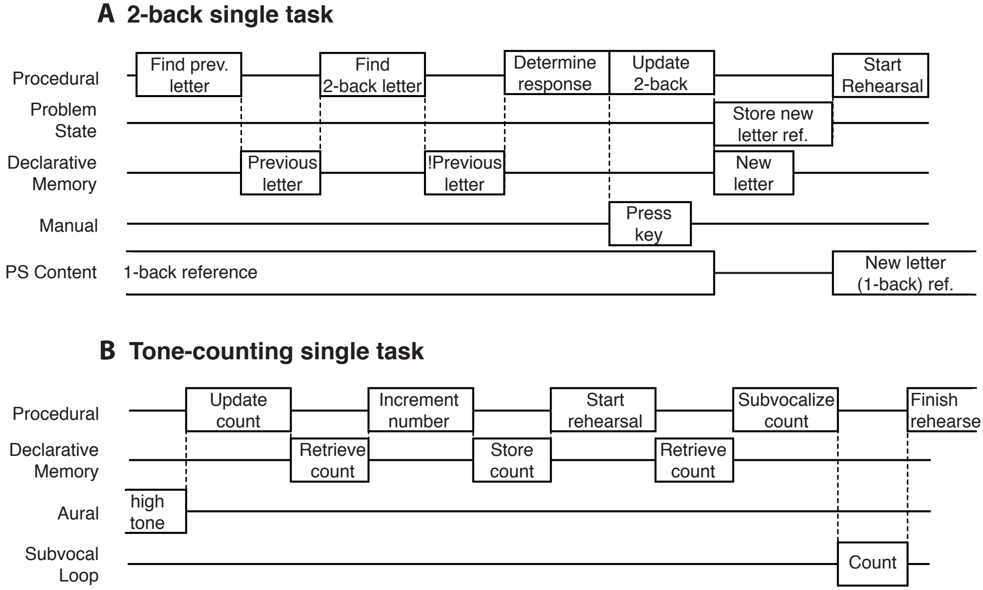

Figure 3-9. Process timelines of the implementations of the WM tasks in Model 2-post. *Panel A:* Updating of the tone count in the tone-counting task. *Panel B:* Deciding what to respond in the 2-back task.

keep declarative chunks active enough for retrieval, the model performs subvocalized rehearsal when there is time (i.e., when there is no new stimulus to process). During rehearsals, letter chunks are retrieved from declarative memory, and using the vocal buffer of ACT-R the thread repeats the letters back to itself. Rehearsals are alternated between the 1-back and 2-back letter in order to keep activation of both chunks high.

Earlier research argues that focal working memory (and thus the problem state) is only needed when information needs to be updated while its being used (Oberauer, 2009). Otherwise it only needs to be kept active in long-term memory. This is reflected in the 2-back task, where several novel items need to be kept track of, and whose role within the task changes over time. Thus, there is considerable incentive to use focal working memory. However, tone counting is a very different type of memory task: the counting ability it requires is a very basic skill that has been extensively trained. Therefore, an alternative strategy to remember the tone count might be to increase the activation of numbers stored in declarative memory directly through rehearsal, thereby bypassing working memory. As such, the Model 2-post iteration of the tone-counting task shown in Figure 3-9B no longer stores the current count in the problem state. Instead, the count is stored in declarative memory. Rehearsals are performed in order to try and keep the correct count as the most active counting chunk. To summarize, instead of the general strategy that Borst et al. (2010) used, we assume a WM strategy that tries to move information that will no longer change to declarative memory as early as possible, thereby incidentally adhering to the politeness policy of

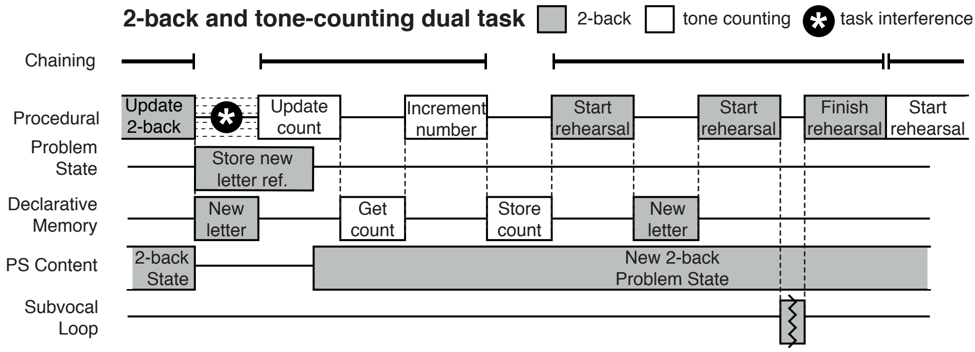

Figure 3-10. Process timeline of dual-task interference between 2-back and tone counting in Model 2-post: Updating the tone-counting count has to wait for the new letter of the 2-back task to be stored, and the 2-back rehearsal has to wait for the count update.

threaded cognition. This is also consistent with work by Lewis-Peacock et al. (2011) that indicates information is removed from WM when it is not needed anymore.

Model 2-post as described up to here was often able to interleave rules of two tasks perfectly, resulting in little to no interference. This can occur more easily compared to Model 1 because WM is now distributed over several systems that can work in parallel. Threaded cognition would predict perfect interleaving when tasks are trained extensively: training causes the initially learned task procedures to be compiled into more efficient and faster versions (Taatgen & Anderson, 2002). Consequently, when two tasks are performed concurrently, these faster procedures have a higher chance of being interleaved perfectly (Anderson et al., 2005; Salvucci & Taatgen, 2008). However, experiments in which this perfect interleaving was achieved, for example Schumacher et al. (2001), required up to five days of training. Therefore the participants in our experiments cannot be considered experts, as they had no experience with the paradigm beforehand. This is supported by the data, which does not show any ceiling effects for task accuracy. To mimic the novice models of Anderson et al. (2005), production rules of a task are executed in groups, as shown in Figure 3-10. This makes the organization of task rules more like the initially learned task procedures: Each group has a modular function, such as identifying a stimulus and determining the correct action, or retrieving a chunk and evaluating its content. These chained production rules lock the other threads out of the production system for the duration of the chain. Hence, interleaving of tasks happens between task procedures implemented as chained production rules, instead of individual task rules.

Production chaining had the unintended side effect of reducing tracking performance when combined with 2-back, as less tracking input was produced. However, participant data showed similar rates of tracking input when compared to the tracking single-task condition. It is possible that participants were adapting to a

Measurement	Model 2-pre		Model 2-post	
	R^2	RMSE	R^2	RMSE
2-back error	.88	.11	.99	.01
2-back latency	.81	.11	1.0	.03
Tracking error	.99	.08	.97	.04
Tone-counting error	.99	.02	.99	.02

Table 3-4. Experiment 2 behavioral fit for all relevant models. RMSE = root mean square error.

Module	Model 2-pre		Model 2-post	
	R^2	RMSE	R^2	RMSE
Problem State	.36	.39	.89	.22
Declarative Memory	.73	.31	.87	.14

Table 3-5. Fit measures of the area-under-the-curve BOLD data for Model 1 and 2. RMSE = root mean square error.

different strategy for this task combination in which they were predicting the correct tracking input while focusing on the 2-back task. To approximate this behavior, the 2-back task model will periodically allow the tracking task to intrude, where it will generate motor responses in the direction the target was last seen moving without visually focusing on the tracking task. As visual focus is not shifted, this heuristic approach to tracking the target is used to relieve some of the constant visual switching between the tasks. This is a departure from a core concept of threaded cognition, where implicit interleaving of independent task models fully explains observed behavior: while interleaving is able to capture most of the behavior, it does not address possible adaptations people make to tasks in certain situations. This fits with our investigation of WM, which shows that the WM strategies employed depend on the task, and possibly the particular combination of tasks as well.

Model 2-post Evaluation

Model 2-post was fit to the data, and the dark-green bars of Figure 3-7 shows its behavioral results compared to the participant data. It is clear that Model 2-post shows the same qualitative prediction as Model 2-pre. Quantitatively Model 2-post has a better fit for the 2-back task, both in terms of accuracy and reaction times: the 2-back single-task reaction time is now estimated correctly (Table 3-4).

More importantly, Model 2-post accurately captures the BOLD activity for the problem state and declarative memory (Figure 3-8, dark-green bars; Table 3-5). The model no longer predicts a strong over-additive effect for the 2-back plus tone counting condition. Instead, when compared to the other dual-tasks, activity in the problem state drops slightly in the 2-back plus tone counting condition. In contrast, activity rises slightly in the declarative memory resource for this condition. Both of these predictions are in agreement with the neuroimaging data. Furthermore, the

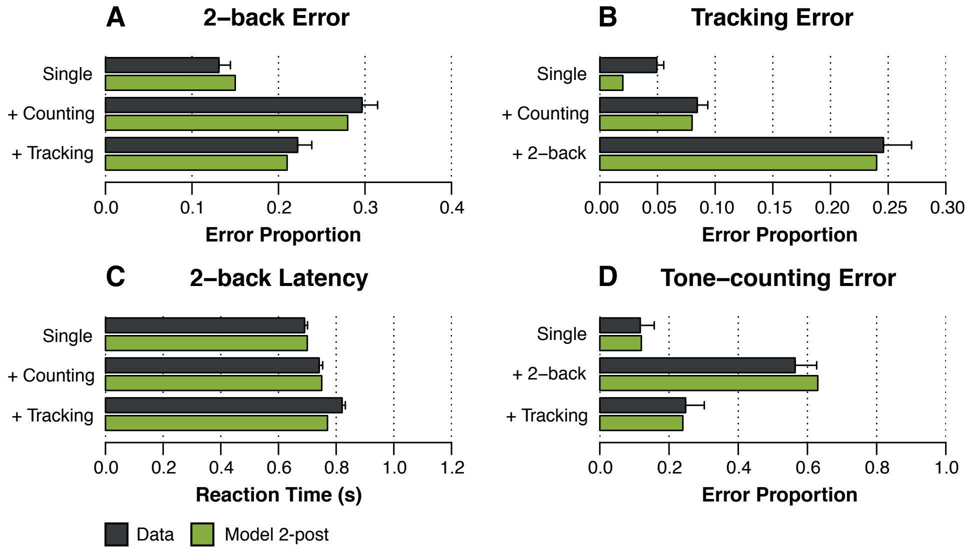

Figure 3-11. Model 2-post fit for the data of Experiment 1. Dark-green bars belong to Model 2-post of Experiment 1. Dark-grey bars represent the averages of the collected data. All model data was averaged over 1500 runs. See the text for a description of Model 2-post.

model no longer predicts problem state activity for the tone-counting task, and a substantial amount of activity for tone counting in declarative memory. Again, these results are consistent with participant data. The model does seem to overestimate the activity for the 2-back task. According to the model, 2-back single-task leads to higher activity in both the problem state and declarative memory when compared to 2-back plus tracking. In the neuroimaging data, however, the opposite is true. Possible explanations for this discrepancy are that participants are less engaged during the 2-back single-task than Model 2-post is, or that the 2-back plus tracking condition contains some minor additional processes that are not included in the model.

In order for Model 2-post to be considered as a good explanation of the observed behavior and fMRI data, it should generalize to other datasets. At the very least, it should be able to fit the behavioral data of Experiment 1. As seen in Figure 3-11 and Table 3-6, Model 2-post also provides an excellent fit for Experiment 1. The model parameters presented in Table 3-7 indicate show that the values for Experiment 1 and 2 are surprisingly similar, given that both the participant groups and the environment in which the study was performed were different.

To summarize, we see that Model 2-post is more in line with the neuroimaging data than Model 2-pre. Quantitatively, the behavioral prediction of Model 2-post has improved as well: the 2-back RT pattern is much closer to the participant data than before. Thus, all results suggest that Model 2-post is a better explanation of the data than Model 2-pre, and generalizes over both experiments.

Measurement	R^2	RMSE
2-back error	.99	.02
2-back latency	.87	.03
Tracking error	.99	.02
Tone-counting error	1.0	.04

Table 3-6. Experiment 1 behavioral fit for Model 2-post. RMSE = root mean square error.

Parameter	Experiment 1	Experiment 2
Retrieval threshold τ	0.168	0.142
Activation noise ε_i	0.042	0.043
Latency-scaling factor F	0.155	0.1
Sound decay time Sd	1.5	1.5

Table 3-7. ACT-R parameters of Model 2-post for Experiment 1 and 2.

Discussion

In Experiment 2 we used fMRI data to further investigate working memory interference during dual tasking: Together, Experiment 1 and 2 show that the observed task interference in our paradigm is robust and consistent. Results showed that Model 2-pre was unable to accurately predict the neuroimaging data: The problem state as singular WM bottleneck, as used by Model 2-pre, could not explain the BOLD response observed between the 2-back and tone-counting tasks. Consequently, we created Model 2-post with a distributed approach to WM, which was able to fit both behavioral and neuroimaging data. Thus, the neuroimaging data proved to be an important tool to evaluate the plausibility of our model (Borst et al., 2015). The neuroimaging data were also invaluable in constraining our modeling space during construction of Model 2-post: activity in participants indicated that declarative memory and rehearsal played a larger role than we anticipated. By using these constraints we were able to obtain a good fit for both behavioral and neuroimaging data.

Although declarative memory seems to play a larger role, the BOLD response for 2-back with tone counting in Figure 3-8 shows a much lower value in Model 2-post compared to Model 2-pre. This is due to the problem state switching mechanism of Model 2-pre: When two tasks both need to use the problem state, this results in the content of the problem state constantly being switched in and out. The information that is put in the problem state comes from declarative memory, which is therefore used very intensively. Model 2-post does not use this mechanism, so no additional declarative memory activity is generated when interference between the two tasks occurs. In effect, having the tasks rely more on declarative memory causes less total activity in the declarative module.

The neuroimaging constraints furthermore led to the use of production chaining. Production chaining seems to show that in working-memory tasks, rule interleaving is more complicated than threaded cognition would predict: There is additional control over interleaving, causing rules of a task to be executed as a group before control is given over to the other task. Production chaining could be indicative of novice behavior: Because the participants were not given much training, the task rules have not been proceduralized, resulting in inefficient (and slow) task execution (Taatgen & Lee, 2003).

General Discussion

Over the course of two experiments we tested and refined a cognitive model of dual tasking that focuses on the way in which working memory is used in multitasking situations. In particular, we were interested in how interference can occur when multiple working memory tasks are performed concurrently, and what mechanisms are involved.

For Experiment 1, our model predicted that a 2-back task combined with a tone-counting task would result in the largest increase in errors, while tone counting combined with tracking would have the lowest increase. This strong interference between 2-back and tone counting in the model was due to contention for the problem state resource, which stores the information required to perform either memory task. Behavioral data from Experiment 1 supported the model prediction, and the fitted Model 1-post showed an excellent quantitative fit to the data, with the exception of the 2-back single-task reaction time. However, when we tested fMRI predictions of this model in Experiment 2, it turned out that the predicted strong over-additive effect for the problem state resource was absent in the data. Instead, the data indicated a much stronger declarative-memory component when compared to the prediction. To match both the neuroimaging and the behavioral data, we changed the model's working memory from a strong focus on a single resource to a collaboration of several resources with distinct roles. Interference between 2-back and tone counting occurred as contention for declarative memory and rehearsal opportunities. The good fit of Model 2 to both behavioral and neuroimaging data indicated that WM interference in dual tasks is an interaction between several resources that together form the working-memory strategy.

Creating and adapting a model for working-memory interference in dual-tasking situations has provided valuable insight into the recruitment and function of WM resources. In particular, examining WM models of different complexity allowed us to form a description of the circumstances required to recruit the problem state and declarative memory in a working-memory task. This can be seen in Table 3-8, which details what components are used by which model iteration. The model revisions highlight the importance of having sufficient constraints when building a cognitive model. While the interactions between three separate tasks by itself imposes strong

Task	Property	Models			
2-back		**1-pre**	**1-post**	**2-pre**	**2-post**
	Problem state	✔	✔	✔	✔
	Declarative memory	✔	✔	✔	✔
	Production chaining				✔
	Rehearsal				✔
Tone counting					
	Problem state	✔	✔	✔	
	Declarative memory	✔	✔	✔	✔
	Production chaining				✔
	Rehearsal				✔

Table 3-8. Resources and mechanisms used by working memory tasks, evaluated across models.

constraints on the modeling space, behavioral data points alone did not prove sufficient to accurately validate the model. This was evident from Model 2-pre, which fit the behavioral data, but not the fMRI results: The neuroimaging data proved to be invaluable for model construction, by giving additional insight into the resources recruited by a task over time.

The modeling work indicates that working memory is a bottleneck, but not necessarily in the same way as the classic response-selection bottleneck (Pashler, 1994): Working memory consists of multiple parts, each of which may be considered a serial bottleneck in itself. The particular working-memory parts used by a task can be adapted according to task requirements, and the particular working memory mechanisms used by a task play an important role in how working memory interference affects performance. We have found three components involved in working memory during dual tasking: the problem state, declarative memory, and subvocalized rehearsal. Based on Experiment 1 and 2, we hypothesize that the problem state resource is exclusively engaged when information needs to be manipulated. Furthermore, declarative memory is recruited when the information that needs to be maintained is simple, requires no manipulation, or exceeds the capacity of the problem state. Subvocalized rehearsal functions as an additional support system for declarative memory, as it counters the activation decay of the relevant stored items.

While the exact organization and functionality of working memory is still debated (Baddeley, 2012), the working-memory system used in our model can be considered a fully functional system according to the definition by Oberauer (2002): The three working-memory components used by our model are very similar to the three functionally-distinct components outlined in Oberauers' (2002) concentric model. As mentioned, the problem state is very similar to the focus of attention, which holds a single item that can be used to perform cognitive operations. The declarative-memory module can be considered an activated long-term memory, which contains recently activated items that are not directly available for further cognitive processing

(Cowan, 1995; Oberauer, 2002). The third component proposed by Oberauer is the direct-access region, which holds a small set of items that can be immediately used by other cognitive operations. While ACT-R does not contain a direct equivalent to this, our model implements a similar functionality through subvocalized rehearsal: both systems have a small capacity that is limited by competition for memory retrievals. The direct-access region is substantially more complex than the subvocalized rehearsal however, as it is able to integrate distinct elements into a single structure. In our model, this would occur in the problem state, where new structures can be constructed. While the problem state and the focus of attention are very similar, the latter functions solely as a direct access point to a specific element. Taken together, these differences in functionality point out that the specific functionalities attributed to each working-memory component are distributed differently in our dual-task model when compared to the Oberauer (2002) concentric WM model.

The particular working-memory components recruited by the tasks depended on task complexity: If the information that needs to be maintained is very simple, such as highly trained memory items (letters, numbers), then the problem-state component of working memory can be omitted. Instead, the items can be rehearsed directly from declarative memory to keep them sufficiently active for immediate recall. However, when a task requires new information chunks to be constructed, and the content of these chunks can change over time, the problem state will be used.

The multi-component WM system we present can help address the issues that EPIC and ACT-R have in regards to WM use during multitasking: EPIC has no constraints at all on working memory, and does not explain how strategies for novel task combinations are formed. The greedy-polite policy of threaded cognition, on the other hand, is troublesome for WM: Assuming that only one task can use WM at any given time results in a conflict when multiple tasks require WM at the same time. Furthermore, a task cannot keep WM occupied indefinitely (Borst et al., 2010). Thus, when performing two WM tasks, control of WM by one task must be released at some point to allow the other task(s) access to it. However, ACT-R/threaded cognition has no method to determine when this release of WM occurs, since that would require knowing beforehand how long the information needs to be available in WM. Our model shows that a mix of both the EPIC and ACT-R approaches can be used: individual resources of working memory behave as serial bottlenecks, much as threaded cognition would predict: only one task has access to the problem state or declarative memory at any given time. However, the interleaving of operators in a working-memory dual task seems to share some similarities with EPIC. The production rule chaining used in Model 2-post indicates that some form of strategy plays a role, at least at the novice level: operators (or rules) that belong to the same action of a task are executed as a single block before control is given to the other task. Production chaining has led to a possible solution for determining when control of the problem state is released in threaded cognition. Namely, control of the problem state is released when the action performed on the item currently loaded into the problem state has been resolved. These actions can be extensive: such as the retrieval of an item from declarative memory, altering the content of this item, and returning the item to declarative memory.

The ability to revise our model to include things such as production chaining, while remaining in a constrained modeling space, is a strength of our modeling approach: In methods such as CBR (Howes et al., 2009) the model space that can be searched is determined beforehand. This works well with relatively simple paradigms where this space can be covered to a significant degree, but is difficult with more complex tasks. Because certain model variations of complex tasks are unlikely to be considered by the modeler (such as production chaining in our case), the selected model space might not include the optimal model. By making strong initial predictions, and ensuring that models generalize over all datasets, we attain more flexibility in model selection by integrating findings from model validation into subsequent model iterations. We believe an even more strenuous model selection process would use both methods: using prediction and generalization to generate several model candidates, from which the optimal model is selected using CBR.

The difference in resource use between 2-back and tone counting leads to an interesting question: what are the circumstances that cause a specific memory-related resource to be utilized? The models we have constructed suggest that there is substantial strategic freedom within the constraints of the cognitive architecture. In the 2-back task, there is strategic choice whether to keep the last item in the problem state, or to consolidate it in declarative memory as soon as possible. In the tone-counting task, the count can be maintained by either storing it in the problem state, or to use a combination of subvocalization and declarative memory. In both cases the evidence suggests that the dominant choice is to use the problem state resource as little and briefly as possible. This strategy is consistent with the "politeness" policy of threaded cognition, and makes it easier to support a smooth alternation between multiple tasks that need working memory.

To conclude, both the model and the data indicated that WM interference is the result of an interaction between multiple resources as opposed to a single bottleneck. Furthermore, what particular resources are used in the working-memory strategy is strongly task dependent. The consequence of this finding is that estimating the performance of a WM dual task is more complicated than previously thought: if WM is not a singular bottleneck, then parallel processing can occur between the different systems that make up WM as a whole, and it is hard to predict the resulting dual-task performance.