The Problem State: A Cognitive Bottleneck in Multitasking

In which we present first support for a single-sized problem state resource in the form of three behavioral experiments.

This chapter was previously published as:

Borst, J.P., Taatgen, N.A., & Van Rijn, H. (2010). The Problem State: A Cognitive Bottleneck in Multitasking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 36*(2), 363–382.

2

Chapter

The Problem State: A Cognitive Bottleneck in Multitasking

2

Chapter

Abstract

The main challenge for theories of multitasking is to predict when and how tasks interfere. Here we focus on interference related to the problem state, a directly accessible intermediate representation of the current state of a task. On the basis of Salvucci and Taatgen's (2008) threaded cognition theory, we predict interference if two or more tasks require a problem state, but not when only one task requires one. This prediction was tested in a series of three experiments. In Experiment 1, a subtraction and text-entry task had to be carried out concurrently. Both tasks were presented in two versions: one that required maintaining a problem state and one that did not. A significant over-additive interaction effect was observed, showing that the interference between tasks was maximal when both tasks required a problem state. The other two experiments tested whether the interference was indeed due to a problem state bottleneck, instead of cognitive load (Experiment 2; an alternative subtraction and text entry experiment) or a phonological loop bottleneck (Experiment 3; a triple-task experiment that added phonological processing). Both experiments supported the problem state hypothesis. To account for the observed behavior, computational cognitive models were developed using threaded cognition within the context of the cognitive architecture ACT-R (Anderson, 2007). The models confirm that a problem state bottleneck can explain the observed interference.

Introduction

Some tasks can be performed together effortlessly, like walking and talking, while other tasks interfere with each other, like car driving and phoning, while again other combinations of tasks are nearly impossible to do concurrently, like writing a manuscript and talking to a colleague. Intuitively, it seems clear why some tasks interfere with each other and some do not: the more overlap in cognitive constructs between tasks, the more interference. For instance, writing a paper and talking to a colleague both use language faculties, resulting in major interference between the tasks.

Psychologists have been formally investigating multitasking behavior at least since the 1930s (e.g., Telford, 1931; see Meyer & Kieras, 1997a for an excellent review). Based on the large body of research collected since the 1930s, detailed cognitive models of multitasking have been developed, ranging from concurrent multitasking (e.g., Kieras et al., 2000; Salvucci, 2005) to task switching (e.g., Altmann & Gray, 2008; Gilbert & Shallice, 2002; Sohn & Anderson, 2001; see also Monsell, 2003) to sequential multitasking (e.g., Altmann & Trafton, 2007). These computational models make it possible to predict the amount of interference between tasks on a quantitative level. To unify several areas of multitasking, Salvucci and Taatgen (2008) recently proposed a new theory of multitasking behavior, threaded cognition, which accounts for concurrent multitasking as well as for sequential multitasking (see also Salvucci, Taatgen, et al., 2009). Threaded cognition was implemented in the cognitive architecture ACT-R (Anderson, 2007), enabling researchers to make formal models of multitasking behavior.

In threaded cognition, tasks can use several distinct cognitive resources, such as vision, manual operations, or memory. These resources can operate in parallel, but are themselves serial in nature (cf. ACT-R, Anderson, 2007; Byrne & Anderson, 2001). Because of this seriality, a resource can only be involved in one operation at a time, but multiple resources can be active at the same time. This within-resource seriality but between-resource parallelism holds regardless of whether the resources are recruited for a single task (e.g., physically moving a disc in a Towers of Hanoi problem while at the same time using memory to plan the next move) or whether the resources are recruited for different tasks (manually tuning the car-audio system while at the same time visually processing the road in front of the car). Thus, the key assumption related to multitasking in threaded cognition is that although several tasks can be active at the same time, a particular resource can only be used by a single task at a time. For instance, if two tasks want to use the visual system at the same time, only one of them can proceed and the other task will have to wait. In the case of the visual system this is quite obvious: we can only look at one object at a time. However, the same mechanism is assumed to hold for more central resources, like memory. For example, if two tasks want to retrieve a fact from memory at the same time, only one task can proceed; the other task will have to wait. On the other hand, no interference is predicted if one task wants to use the visual system, and one task wants to retrieve a fact from memory. Thus, as long as the resource requirements of the different tasks do not overlap in time, threaded cognition predicts no interference, but as soon as a particular resource is concurrently needed by two or more tasks, that resource will act as a bottleneck and delay the execution of the combined process. This aligns with the intuition that if two tasks require the same cognitive constructs, the tasks will interfere.

Salvucci and Taatgen (2008) discussed two peripheral bottlenecks (the visual and motor system) and two central cognitive bottlenecks (declarative and procedural memory; cf. "attentional limitations", Pashler & Johnston, 1998). In this article, we discuss a third central cognitive resource that can result in significant interference, both in terms of decreased speed and increased errors: the problem state. The problem state resource is used to maintain intermediate mental representations that are necessary for performing a task. For instance, while solving an algebra problem like $2x - 5 = 8$, the problem state can be used to store the intermediate solution $2x = 13$. The problem state resource is assumed to be limited to only one coherent 'chunk' of information (Anderson, 2005, 2007), and will therefore cause interference when multiple tasks concurrently require its use. However, not all tasks require the use of a problem state. If no intermediate results need to be stored (e.g., solving one step of the algebra problem $2x = 8$ immediately results in the required answer) or all necessary information is present in the world (e.g., if the intermediate steps can be selected from and are displayed on a computer screen), there is no need for maintaining a mental representation.

Previously, we have presented results (Borst & Taatgen, 2007) that illustrated the potential role of the problem state resource as a bottleneck in multitasking. In this study, participants had to type in an address in a simulated navigation device while driving a simulated car. The task required switching back and forth between driving and operating the navigation device. Both tasks had two versions: one that required maintaining intermediate results, and one in which there were no intermediate results. More specifically, in the driving task participants had to memorize the turns to take at the next intersections in one condition, while in the other condition arrows pointed out the route. In the navigation task, the two conditions differed in whether the participants had to memorize the full address before entering it, or whether the device would show what letter to press next. When both difficult conditions were combined, performance was much slower and more error-prone than could be explained by the difficulty of the separate tasks alone. That study suggested that combining certain tasks yield additional costs in terms of time and errors. However, the setup of the study was relatively under-constrained, making it difficult to derive precise conclusions.

In the current article, we investigate whether the problem state resource constitutes a bottleneck in a more constrained setting. In the first experiment, participants performed a complex dual-task. Data of this experiment are in line with predictions derived from a problem state bottleneck–based theory. However, to test whether the results of Experiment 1 were caused by cognitive load effects (e.g., Logan, 1979), Experiment 2 controls for cognitive load over the different conditions, while in Experiment 3 another possible explanation involving the phonological loop was investigated. Experiments 2 and 3 both provide corroborating support for a problem state bottleneck account. The experimental findings are supported by computational cognitive models, which show that a problem state bottleneck can explain the observed interference effects. Before we describe the experiments, we will introduce the problem state resource and the threaded cognition theory in more detail.

The Problem State Resource

In our terminology, the problem state resource is used for storing intermediate information that is necessary for performing a task. Information in the problem state resource is directly accessible for the task at hand, while it takes time to retrieve facts from declarative memory (cf. ACT-R, Anderson, 2007). For instance, while mentally solving an algebra problem like $3x - 12 = 0$, the problem state can be used to store the intermediate solution $3x = 12$; and when asking for directions, the problem state can be used to store at which street you should turn to arrive at your destination. If this information is present in the world, that is, if you work out an algebra problem on paper or follow road signs to a destination, it is not necessary to maintain a problem state.

The concept of the problem state stems from a series of neuroimaging experiments by Anderson and colleagues, who found BOLD activity in the posterior parietal cortex that correlates with the transformation of mental representations (e.g., Anderson, 2005; Anderson, Albert, & Fincham, 2005; Anderson, Qin, Sohn, Stenger, & Carter, 2003; Sohn et al., 2005). They concluded on this basis that a separate resource exists for maintaining and transforming mental representations.

The problem state construct is closely linked to mental states as used by Altmann et al. in their cognitive control model and memory for goals theory to explain task switching and task interruption behavior (Altmann & Gray, 2008; Altmann & Trafton, 2002, 2007). However, where in their case mental representations constitute both the goal and the problem state of a task, in threaded cognition (also in the current version of ACT-R, e.g., Anderson, 2005, 2007) these mental representations have been split into a goal state that only maintains the state of the current goal, and a problem state that maintains temporary intermediate information necessary for doing the task (but see Salvucci, Taatgen, et al., 2009, about how these theories can be reconciled). The problem state is also related to the 'episodic buffer' in Baddeley's (2000) extension of the classical working memory model of Baddeley and Hitch (1974). This buffer serves the function of a "limited capacity temporary storage system that is capable of integrating information from a variety of sources" (*p*. 421), which was previously part of the 'central executive' (Baddeley, 2003). This construct is very similar to ACT-R's problem state resource, in the sense that both systems can integrate information from different sources (perceptual and long-term memory) and temporarily store the outcome for further processing.

The Threaded Cognition Theory

Threaded cognition (Salvucci & Taatgen, 2008) is an integrated theory of human multitasking. In threaded cognition, every task is represented by a so-called cognitive thread. For instance, in the case of driving a car and operating a navigation device, one thread would represent steering the car and another thread would represent operating the navigation device. A thread is associated with the goal of a task, which serves as a key to mobilize associated task knowledge (e.g., declarative and procedural memory that is necessary for performing the task). Although multiple threads can be active at a time, only a single procedural processor is available; thus, although multiple threads are active in parallel, only one thread can use the procedural processor at a time (compare this to multiple programs running on a single CPU on a computer: while the CPU can only process one instruction at a time, programs act as if they were executed concurrently). Furthermore, if a thread needs to use a cognitive resource such as vision or memory, it can only be selected for execution if that resource is available. Thus, while the threads act in parallel and are not governed by any supervisory executive control structure, they are constrained by the available resources. (For a similar approach, but from a more mathematical point of view, see Liu, Feyen, & Tsimhoni, 2006.)

The threaded cognition theory 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 visual and aural modules 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, the timing module (Taatgen et al., 2007; Van Rijn & Taatgen, 2008) and the problem state module[1]. All modules operate in parallel, but each module in itself can only proceed serially (Byrne & Anderson, 2001). 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 (Anderson, 2007). 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 $8x = 12$. Thus, the goal module holds the current state of a task, while the problem state module 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 at a time.

Threaded cognition extends ACT-R by allowing for multiple parallel goals, and thus multiple tasks (threads), to be active. This translates into the assumption that the goal module in ACT-R can represent several goals at the same time. However, the other modules can still only do one thing at a time, which means that they can only be used by one thread at a time. The modules are shared on a first-come-first-served basis: a thread will 'greedily' use a module when it needs it, but also will 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 thread will have to wait for the other.

[1] *Sometimes referred to as 'imaginal module' or 'problem representation module'.*

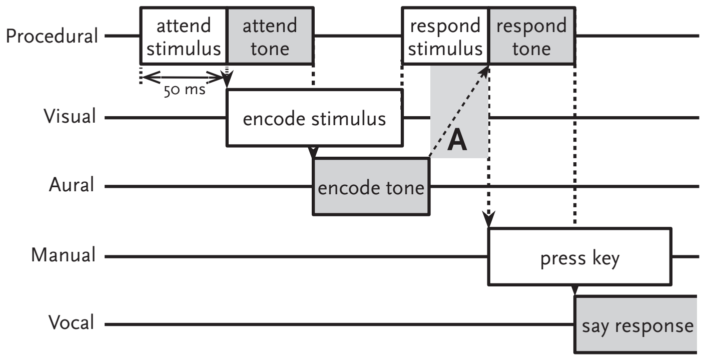

Figure 2.1 Example processing stream in threaded cognition. White boxes depict a visual–manual task, grey boxes an auditory–vocal task. The 'A' represents interference, caused by both threads needing the procedural resource at the same time.

In Figure 2.1 an example processing stream of a dual-task in threaded cognition is shown: white boxes depict a task in which a key-press is required in response to a visual stimulus, and grey boxes depict a task in which a vocal response is required in response to an auditory stimulus. The *x*-axis represents time and boxes represent the period of time during which a resource is used. Both tasks start by activating production rules to initiate attending the respective stimuli, after which the encoding process starts in both the visual and the aural module. The grey area marked A indicates interference, caused by the concurrent request for the procedural module after the respective encoding steps. As the visual–manual task already uses the procedural module, the auditory–vocal task has to wait. Thus, if multiple threads require a resource at the same time, interference is observed.

Salvucci and Taatgen (2008) presented cognitive models that account well for dual-tasking in a number of different domains, ranging from simple Psychological Refractory Period (PRP) tasks to driving a car and using a cell phone concurrently. These models showed that bottlenecks in perceptual and motor resources in addition to bottlenecks in two more central cognitive resources (procedural and declarative memory) account for a wide range of multitasking interference phenomena (but see for a more detailed account of interference in the motor system e.g., Albert, Weigelt, Hazeltine, & Ivry, 2007; Diedrichsen, Hazeltine, Kennerley, & Ivry, 2001). Although multiple bottlenecks are identified, not all bottlenecks result in the same interference profiles. The severity of the interference depends on the particular resource: procedural memory is very fast and therefore only leads to delays in the order of 50 ms (but see Taatgen, Juvina, Schipper, Borst, & Martens, 2009, for an example where interference caused by the procedural resource explains counter-intuive results in an attentional blink dual-task). On the other hand, interference due to declarative memory and the visual and motor system leads to pronounced decreases in speed in the order of 200–500 ms.

Salvucci and Taatgen (2008) did not investigate the role of the problem state resource in multitasking. Because many tasks require the maintenance of intermediate representations and because this maintenance is required for relatively long periods of time (i.e., several seconds), we hypothesize that the problem state is a important source of interference in multitasking. We will now turn to three experiments that test this hypothesis.

Experiment 1: Subtraction & Text-Entry

In Experiment 1, 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. Thus, the experiment has a 2 × 2 factorial design (Subtraction Difficulty × Text-Entry Difficulty). As threaded cognition claims that the problem state resource can only be used by one task concurrently, we hypothesized that when a problem state is required in both tasks (the *hard–hard* condition), participants will be significantly slower or make more errors than in the other conditions. On the other hand, if just a single task requires a problem state, no interference is to be expected on behalf of the problem state. Thus, we expected an over-additive interaction effect of task difficulty.

Method

Participants

Fifteen 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 as approved by the Ethical Committee Psychology of the University of Groningen 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 (see Figure 2.2). Participants had to alternate between the two tasks: after a digit, the subtraction interface was disabled, forcing the participant to subsequently enter a letter. After entering a letter, the text-entry interface was disabled and the subtraction interface became available again.

The subtraction task is shown on the left side of Figure 2.2. Participants had to solve 10-column subtraction problems in standard right to left order; they had to enter the digits with their left hand using the keyboard. In the easy, no problem state 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 (as shown in Figure 2.2) required participants to borrow six times. The assumption is that participants use their problem state resource to keep track of whether a borrowing is in progress.

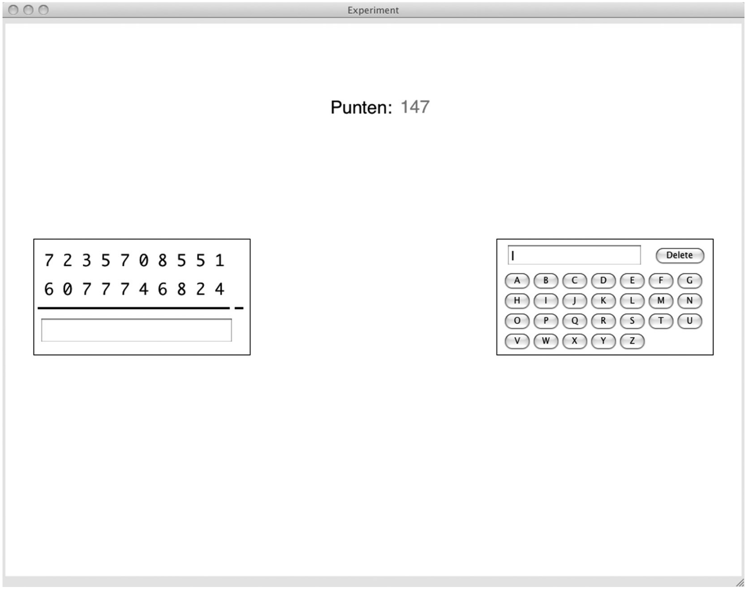

Figure 2.2 Screenshot of Experiment 1.

The second task in the experiment is text-entry. The interface is shown on the right in Figure 2.2: by clicking on the on-screen keypad 10-letter strings had to be entered. In the easy version of the text-entry task, the strings were presented one letter at a time. Participants saw one letter appear on the screen (for example the 'I' in Figure 2.2) and had to click the corresponding button on the keypad. As soon as a button was pressed, the text-entry keypad was disabled and the mouse pointer was hidden to prevent participants from putting the pointer on the next letter. When the text-entry task was re-enabled, the mouse pointer appeared again in the location where it had been hidden.[2] Participants could only enter the next letter after the next subtraction column was responded to. After 10 letters had been entered, the trial ended automatically. In the hard version, a 10-letter 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 (thus, participants could neither see what word they were entering, nor what they had entered, the text-entry screen remained blank until the end of the trial). Otherwise, both conditions were identical. In the hard version, we assume that participants need their problem state resource to keep track of what word they were entering and at which position they are (e.g., "*informatie*, 4th position").

As is shown in Figure 2.2, participants could earn points (*punten* in Dutch). Participants started out with 200 points. While performing the tasks, the counter at the top of the screen decreased by 2 points per second. For every correct letter or digit 10 points were added to the total (addition was done after finishing the complete trial).

[2] *Participants could have used the mouse to indicate what the last letter was that they entered. However, that would have made it harder to find our results, as that means that they would have maintained less information mentally (only the word, not the position within the word).*

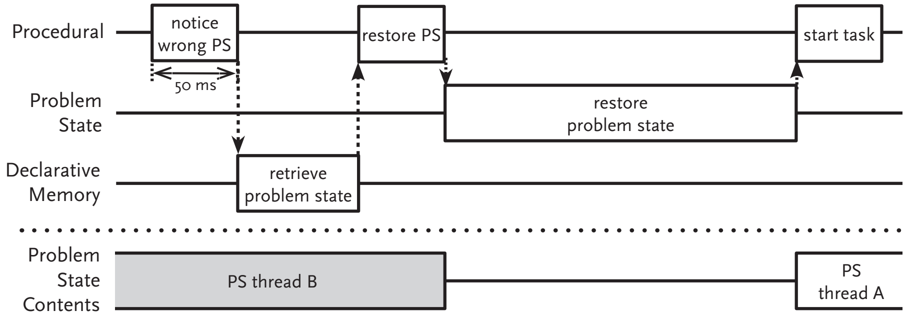

Figure 2.3 Processing stream of replacing a problem state. PS = problem state.

At the end of a trial a feedback display was shown to the participants, indicating how many points they gained per task in the current trial. In effect, to score a high amount of points participants had to act both quickly and accurately.

Stimuli and Apparatus

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, Baayen, Piepenbrock, & Van Rijn, 1993), 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 (under the constraint that a letter never appeared twice in a row). Thus, participants entered random sequences of letters. This did not introduce difficulties, because the participants never saw the complete letter-sequences but had to enter the letters one-by-one. By scrambling the words, we controlled for letter-based effects, while preventing the use of alternative strategies to predict the next letter.

The experiment was presented full screen on a 19-inch monitor. The width of both the subtraction interface and the text-entry interface measured 9 centimeters, while the space between the two tasks was 10 cm; the height of the interfaces was 4.8 cm (see also Figure 2.2). Participants were sitting at a normal viewing distance, about 75 cm from the screen.

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 required to alternate between the tasks. After the last response of a task within a trial a feedback display appeared, showing how many letters or digits had been 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 for the dual-task. 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, the participants had to perform 36 trials, presented semi-randomly. The complete experiment lasted approximately 45 minutes. Halfway the experiment participants could take a short break.

Model

We will first describe the computational cognitive model[3] that we developed for the task, after which the behavioral and modeling results will be presented side by side. The model was developed in the ACT-R cognitive architecture (Anderson, 2007; Anderson, Bothell, et al., 2004), using threaded cognition (Salvucci & Taatgen, 2008).

Of particular importance for the tasks at hand is ACT-R's problem state module. This module can hold a problem state, accessible at no time cost. However, changing a problem state takes 200 ms (Anderson, 2007). Because the problem state module can only hold one chunk of information, the module's contents have to be exchanged frequently when multiple tasks require a problem state. When the problem state is replaced, the previous problem state is automatically moved to declarative memory so that it can be restored when the other thread needs it. Figure 2.3 displays an example processing stream of problem state replacement. The white boxes represent Task A that requires the problem state resource, while the grey box represents the problem state of Task B, occupying the resource at the start of the example. First, the white task notes ("notice wrong PS") that the problem state resource does not contain its own associated problem state, and therefore initiates a process to retrieve this problem state from declarative memory. This retrieval takes a certain amount of time, after which a production rule ("restore PS") fires to start restoring the retrieved problem state to the problem state resource. This takes a fixed 200 ms. After this initialization process, the white task can start with its actual operation. The total time to replace the problem state resource is thus 200 ms plus the time for the retrieval plus 100 ms for the "notice wrong PS" and "restore PS" production rule executions. Thus, when multiple tasks need the problem state resource, the execution time of tasks is increased considerably per change of task. An additional effect of this exchange of problem states is that because problem states need to be retrieved from memory, it is possible that a task retrieves an older, and thus incorrect problem state from memory, resulting in behavioral errors.

The two tasks in the experiment were implemented as two threads: a subtraction thread and a text-entry thread. Both threads use the visual module to perceive the stimuli and the manual module to operate the mouse and the keyboard. In the easy condition of the subtraction task, the model perceives the digits, retrieves a fact from memory (e.g., 5 - 2 = 3) and enters the difference. In the hard condition, the general

[3] *Available for download at http://www.ai.rug.nl/~jpborst/models/.*

process is the same. However, if the model retrieves a fact from memory and notices that the 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). When the model encounters a negative subtraction outcome for the first time in a trial, it notes in its goal state that it is performing the hard version of the task ("subtraction – hard"). This ensures that the model checks for the appropriate problem state at the start of each subsequent response-sequence (as the problem state indicates whether a borrowing is in progress). If 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 know the target word and the current position within that word. Thus, it requires the problem state resource to store what word it is entering and at which position of the word it is ("*informatie*, 4th position"). If the model performs a trial in the hard condition, it will use the word and position in its problem state to come up with the next letter. To simulate the spelling processes required to come up with "letter 5 from the word *informatie*", we have assumed that an additional declarative retrieval is necessary that links the current position to the next letter. As spelling words is not the focus of this article, we did not model this in detail, but instead assumed an additional retrieval. After the model has determined the next letter, it clicks the appropriate button and updates its problem state to reflect that it is one position further in the word.

The ACT-R theory predicts the time it takes to perceive a stimulus, to press a key and to move the mouse, and to retrieve facts from declarative memory, which makes it meaningful to incorporate these parts of the task in the model. These elements of ACT-R have been tested and validated separately, many examples can be found at http://act-r.psy.cmu.edu/. Instead of discussing all details here, we refer the reader to Anderson (2007) for more information.

Because the model requires two problem states that need to be exchanged at each trial in the *hard–hard* condition, and either zero (*easy–easy*) or one (*easy–hard*, *hard–easy*) in the other conditions, it predicts an over-additive effect of task difficulty on response times. Possibly, the number of errors will also increase, depending on whether older and incorrect problem states are retrieved frequently.

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 faster than 250 ms and slower than 9000 ms were removed from the data, after which we removed data exceeding 3 standard deviations from the mean per condition per participant (in total, 2.0% of the data was removed). All reported F- and p-values are from repeated-measure ANOVAs, all error bars depict standard errors, effects were judged significant if they reached a .05 significance level. Accuracy data were transformed using an arcsine transformation before performing ANOVAs. Figure 2.4 shows the main results, black bars depict experimental data, grey bars model data.

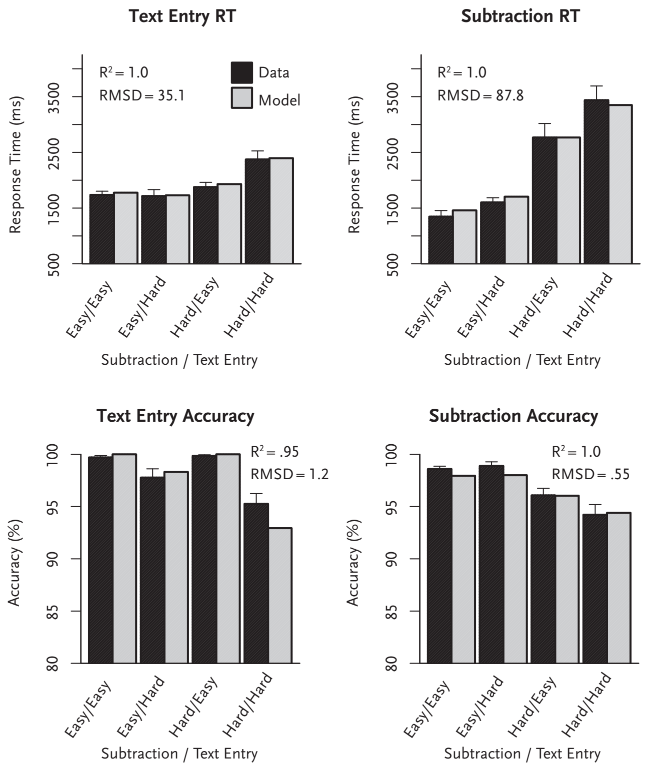

Figure 2.4 Results of Experiment 1. RMSD = root mean squared deviation.

Response Times

Response time on the text-entry task was defined as the time between entering a digit 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 2.4 shows the results. First, an interaction effect between Subtraction Difficulty and Text-Entry Difficulty ($F(1,12) = 22.15$, $p < .001$, $\eta_p^2 = .65$) was found. Next, we performed a simple effects analysis, showing an effect of Text-Entry Difficulty when subtraction was hard ($F(1,12) = 10.78$, $p < .01$, $\eta_p^2 = .47$), and an effect of Subtraction Difficulty when text-entry was hard ($F(1,12) = 47.16$, $p < .001$, $\eta_p^2 = .80$). The other simple effects did not reach significance: Text-Entry Difficulty when subtraction was easy ($F(1,12) = 1.88$, $p = .20$, $\eta_p^2 = .14$) and Subtraction Difficulty when text-entry was easy ($F(1,12) = 3.35$, $p = .09$, $\eta_p^2 = .22$). Thus, there was an over-additive interaction 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 2.4, 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 digit 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 ($F(1,12) = 6.24$, $p = .03$, $\eta_p^2 = .34$). A simple effects analysis revealed that all simple effects were significant: Subtraction Difficulty when text-entry was easy ($F(1,12) = 69.04$, $p < .001$, $\eta_p^2 = .85$), Subtraction Difficulty when text-entry was hard ($F(1,12) = 111.64$, $p < .001$, $\eta_p^2 = .90$), Text-Entry Difficulty when subtraction was easy ($F(1,12) = 11.65$, $p < .01$, $\eta_p^2 = .49$), and Text-Entry Difficulty when subtraction was hard ($F(1,12) = 11.81$, $p < .01$, $\eta_p^2 = .50$). Thus, the more difficult the tasks, the higher the response times, with an over-additive effect in the *hard–hard* condition, reflected by the interaction.

Accuracy

Figure 2.4, lower left panel, shows the accuracy on the text-entry task, in percentage correctly entered letters. Both main effects were significant: Subtraction Difficulty ($F(1,12) = 7.31$, $p = .02$, $\eta_p^2 = .38$) and Text-Entry Difficulty ($F(1,12) = 21.57$, $p < .001$, $\eta_p^2 = .64$). The interaction effect between Subtraction Difficulty and Text-Entry Difficulty shows a trend towards significance ($F(1,12) = 4.65$, $p = .052$, $\eta_p^2 = .28$). Thus, accuracy on the text-entry task decreased as a function of both Text-Entry Difficulty and Subtraction Difficulty, with a trend towards a stronger decrease when both tasks were hard.

In the lower right panel of Figure 2.4, the accuracy on the subtraction task is shown. Here, a significant interaction effect between Subtraction Difficulty and Text-Entry Difficulty was observed: $F(1,12) = 10.50$, $p < .01$, $\eta_p^2 = .47$. A simple effects analysis subsequently revealed that three simple effects reached significance: Text-Entry Difficulty when subtraction was hard ($F(1,12) = 6.68$, $p = .02$, $\eta_p^2 = .36$), Subtraction Difficulty when text-entry was easy ($F(1,12) = 7.17$, $p = .02$, $\eta_p^2 = .37$), and Subtraction Difficulty when text-entry was hard ($F(1,12) = 87.7$, $p < .001$, $\eta_p^2 = .88$). Text-Entry Difficulty when subtraction was easy did not reach significance ($F(1,12) = 3.64$, $p = .08$, $\eta_p^2 = .23$). Thus, when subtraction was hard accuracy was lower, but this effect was even stronger when text-entry was hard as well.

Model

The grey bars in Figure 2.4 show the results of the model. It resembles the empirical data closely (R^2- and Root Mean Squared Deviation-values are displayed in the graphs). The model shows the same interaction effects as the data, both in response times and accuracy. To fit the model, we estimated how long memory retrievals take[4] and how often incorrect memories are retrieved (i.e., retrieving problem states from declarative memory in the *hard–hard* condition, but also arithmetic errors like 9 - 6 resulting in 2 instead of 3). The incorrect retrievals were modeled in a similar fashion as in Anderson,

[4] *ACT-R's latency factor was set to .3 and activation noise to .1. Furthermore, subtraction facts were divided into two groups, one group of facts having a minuend under 10, and one group above 10. A third group was formed by the addition facts. The activation levels for those three groups of arithmetic facts were scaled to fit the participant group's behavior. The exact values of these parameters can be found in the model code online at http://www.ai.rug.nl/~jpborst/models/.*

Reder and Lebiere's (1996; see also Lebiere, 1999) model that accounts for arithmetic errors. All other parameters were kept at the default values of ACT-R 6.0 (Anderson, 2007; see also Anderson, Bothell, Lebiere, & Matessa, 1998).

As explained in detail above, the interaction effect in the model data is driven by the problem state bottleneck in the *hard–hard* condition. The model also accounts for the different reaction time patterns in the two tasks: in the subtraction task there is a large main effect of Subtraction Difficulty, while there is no such effect in the response times of the text-entry task. The model accounts for this by assuming that in the hard subtraction task, participants have to retrieve multiple facts from declarative memory to be able to enter a digit, as opposed to the easy subtraction task, in which only one fact has to be retrieved. In the text-entry task, on the other hand, there is no such difference between the easy and the hard task: in the easy version the model has to look at the display to see what letter it has to enter, while in the hard version it has to retrieve an order fact from memory and use information from its problem state to enter a letter. The timing of those processes is similar, resulting in the absence of a main effect of Text-Entry Difficulty on the response times in the text-entry task (cf. the upper left panel of Figure 2.4).

The model keeps track of the task condition in its goal state ("subtraction – hard", see the model description above). This state was set as soon as the thread noticed that it was performing a hard trial: initially it was always set to easy, but when the model came across a borrowing in the subtraction task or a complete word in the text-entry task, it would be set to hard. Did the participants also keep track of the task condition? We compared response times of subtraction columns from the hard condition in which no borrowing is in progress and in which no new borrowing is necessary (i.e., in every way comparable to columns in the easy condition, except that a borrowing has occurred more than one column back; for instance the left-most column of Figure 2.2), to columns of the easy subtraction condition. The difference in response time (2256.3 vs. 1466.3 ms) is significant (paired t-test, $t(12) = -10.10$, $p < .001$). This seems to indicate that participants were sensitive to the context of the current trial (i.e., the task condition): the task in these no-borrow columns in the hard subtraction conditions is exactly the same as in the easy subtraction task, only the context is different. This is consistent with the model's keeping-track account, which always checks whether a borrowing is in progress in the hard trials but not in the easy trials. For the model, this results in a difference in response times between hard responses that are comparable to the easy task and easy responses, although the difference is smaller (1762.4 vs. 1583.8 ms).

Discussion

The interaction effects in the data are in agreement with our model predictions: an over-additive effect of task difficulty on response times and error rates (a trend in the case of accuracy on the text-entry task). As described above, the model accounts for these interaction effects by proposing a problem state bottleneck that results in higher response times on the one hand (caused by constantly replacing the problem state) and higher error rates on the other (caused by retrieving older, incorrect 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, see Anderson et al., 1996; and Lebiere, 1999).

Another interesting observation is the effect of condition of one task on the other task. More specifically, there is a significant effect of Text-Entry Difficulty on the reaction times of the subtraction task when subtraction was easy, and a marginal significant effect (p = .09) of Subtraction Difficulty on reaction times of the text-entry task when text-entry was easy. As can be seen in Figure 2.4, the model captures these effects. In the model, these effects are due to the time costs associated with updating the problem state at the end of a step in the respective hard conditions. For instance, after entering a digit in the hard subtraction task, the model updates its problem state to indicate that it finished a step in the subtraction task. The text-entry task only starts when this problem state update is finished, causing a slight delay in the start of the text-entry task.

Alternative strategies

Except for an account based on a problem state bottleneck, there might be other possible explanations for the interaction effects. For example, participants might have employed different task strategies depending on the task condition. However, in the case of the text-entry task it is not easy to come up with alternative strategies because the task is so straightforward. In the easy condition, participants have to read a letter and click a button, which does not seem to allow for multiple strategies. In the hard text-entry condition, participants have to memorize the word, as they do not receive any feedback at all. Furthermore, they have to keep track of where they are within a word, for instance by memorizing the position or the last letter they entered. While the model does memorize the position, alternative strategies exist such as memorizing the last entered letter and reconstruct the position from that information. However, irrespective of which strategy was used, participants will have to keep track of the current position in some way, for which we assume they have to use their problem state.

In the case of the subtraction task there is at least one possible alternative strategy. Participants could have used the display to determine whether or not a borrowing is in progress instead of maintaining a problem state (i.e., looking at the previous subtraction column, if the lower term is higher than the upper term, a borrowing is in progress). If this had been the overall strategy, it would have had the same impact on both the *hard subtraction – easy text-entry* and *hard subtraction – hard text-entry* conditions. In that case, one would not expect to find an interaction effect, as the problem state is not used for the subtraction task. However, it is possible that participants only switched to this strategy in the *hard–hard* condition: thus using a problem state strategy as long as text-entry is easy, and switching to an interface strategy when text-entry became hard. This would incur a time cost in the *hard–hard* condition, and would thus have resulted in a similar interaction effect as we found. To rule out this alternative explanation, we controlled for this in Experiment 3 by masking previous columns, yielding, as we will see, the same results. Obviously, alternative strategies also exist for solving a borrow-in-progress. For instance, one could subtract one from the upper term of the next column, or add one to the lower term, giving the same results. However, in both cases it is necessary to keep track of whether a borrowing is in progress, resulting in similar latency predictions.

Is a problem state bottleneck necessary?

As the threaded cognition theory already proposes a number of bottlenecks, is an additional problem state bottleneck necessary to account for the observed interaction? The over-additive interaction is caused by a resource that is required in both hard conditions, but not in the other conditions. As the hard conditions require additional information to be kept available, a bottleneck should be related to this additional information maintenance. The bottleneck associated with production rule execution cannot offer an explanation for the found interactions, because production rule activity cannot store information without using another resource. A possible alternative explanation is that 'problem states' are stored as declarative memory chunks and are retrieved when needed, instead of having a separate problem state resource. In such a model, however, one would not expect to find an interaction effect because declarative memory is never concurrently required by the two tasks, as the participants have to alternate between the tasks. Thus, in that case the first task would retrieve its problem state from declarative memory and give a response, after which the second task would retrieve its own problem state from declarative memory and give a response, etc. Because declarative memory is in that case never required by both tasks at the same time, it cannot explain the effect of one task on the other task. Thus, we would predict a simple additive effect of conditions, not an interaction effect. As the two peripheral bottlenecks cannot be used to store information, we argue that a problem state bottleneck is the most plausible option to account for the human data.

Cognitive load effects

While we argued above that a problem state bottleneck is the most plausible account within the ACT-R-based threaded cognition theory, there is an extensive psychological literature on cognitive load that can also explain the results of Experiment 1. For instance, it is shown that memory load causes an increase in reaction time in tasks as simple as visual search (e.g., Logan, 1979; Woodman, Vogel, & Luck, 2001) and tone classification (Jolicœur & Dell'Acqua, 1999). Thus, in that sense it is not surprising that maintaining an additional memory load (problem state) influences another task with a memory load, resulting in the over-additive interaction effect. To rule out the possibility of cognitive load causing the interaction effect, Experiment 2 was designed. The dual-task setup of Experiment 1 was slightly modified by requiring the participants to switch tasks only after every two responses in each task. Thus, Experiment 2 also includes responses where no problem state switch is required, but where a memory load representing the state of the other task still has to be maintained (see Figure 2.5). This means that the cognitive load is equal (the memory load of the other task) on both responses, while the problem state only has to be switched for the first response and is still available for the second response. According to a cognitive load account, the interaction effect should be present on both responses, but according to a problem state bottleneck account, the interaction effect should only be present on the first response, and disappear on the second.

Experiment 2: Subtraction & Text-Entry – Two Responses Per Switch

Experiment 2 was designed to test whether the problem state bottleneck can be observed when controlling for cognitive load effects. The design of the experiment was the same as Experiment 1, except that participants now had to give two responses on each task before switching to the other task. Thus, the new experiment has a 2 × 2 × 2 design (Subtraction Difficulty × Text-Entry Difficulty × Switch). Switch responses are the first responses on a task, directly after switching from the other task; non-switch responses are the second responses on a task, following a response in the same task (cf. task switching). Figure 2.5 shows the experimental setup, detailing when a memory / cognitive load is present, and when problem state changes are required in the *hard–hard* condition. On the basis of the problem state bottleneck hypothesis and the outcome of Experiment 1, we predict an over-additive interaction effect in the switch condition (because the problem state has to be replaced for each response in the *hard–hard* condition), but simple additive main effects in the non-switch condition (because the problem state does not have to be replaced in any condition, as the previous response was given in the same task). Because the memory load is the same on switch and non-switch responses (whether a borrowing is in progress for subtraction / what the word and position are for text-entry), a cognitive load account would predict identical effects for both switch and non-switch responses. Thus, we did not introduce additional cognitive load, but merely removed problem state changes on the non-switch responses, enabling the comparison between a cognitive load account and a problem state account.

Method

Participants

Fifteen students of the University of Groningen who did not take part in Experiment 1 participated in the experiment for course credit (9 female, age range 18–23, mean age 19.8). All participants had normal or corrected-to-normal visual acuity. Informed consent as approved by the Ethical Committee Psychology of the University of Groningen was obtained before testing.

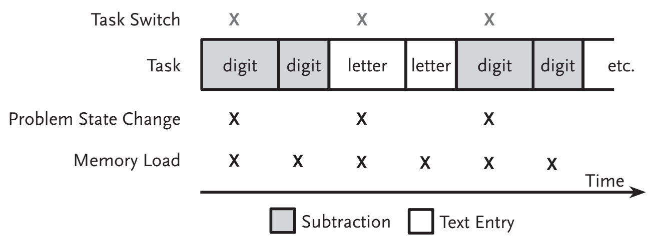

Figure 2.5 Experimental setup of Experiment 2. Grey boxes represent the subtraction task, white boxes the Text Entry task. The black Xs show the problem state and memory load in the hard–hard condition: on the switch responses there is both a problem state change and a memory load, while on the non-switch responses only a memory load is present.

Design, Stimuli, & Procedure

Design, stimuli and procedure were identical to Experiment 1, except that participants were now required to alternate after every two responses, thus they had to enter two digits, two letters, two digits, et cetera.

Model

The model of Experiment 1 was extended to enable it to respond in the situation where a response directly followed a response within the same task.[5] Furthermore, we scaled retrieval times of declarative facts and number of incorrect retrievals to match the new participant group's cognitive arithmetic ability, as we did in Experiment 1.[6]

Results

Only the data of the experimental phase were analyzed. One participant did not adhere to task instructions and was removed from the data set. The same exclusion criteria were used as in Experiment 1 (3.8% of the data was rejected). If not noted otherwise, analyses were the same as in Experiment 1. Figures 2.6 and 2.7 show the main results for response times and accuracy.

Response Times

In line with our hypothesis, ANOVAs on response times showed significant three-way interactions of Subtraction Difficulty × Text-Entry Difficulty × Switch on both the response times of the text-entry task ($F(1,13) = 29.99$, $p = .001$, $\eta_p^2 = .70$) and the subtraction task ($F(1,13) = 5.96$, $p = .03$, $\eta_p^2 = .31$). Therefore, the following analyses were performed separately on the switch and non-switch data.

[5] *While the model was extended, we could have used this new model for Experiment 1 without affecting the results; the situation in which a response can be followed by a response on the same task just never occurs in Experiment 1.*

[6] *ACT-R's latency factor and activation noise were not changed (respectively .3 and .1). The activation levels of the three groups of arithmetic chunks of Footnote 4 were adjusted for the new group of participants.*

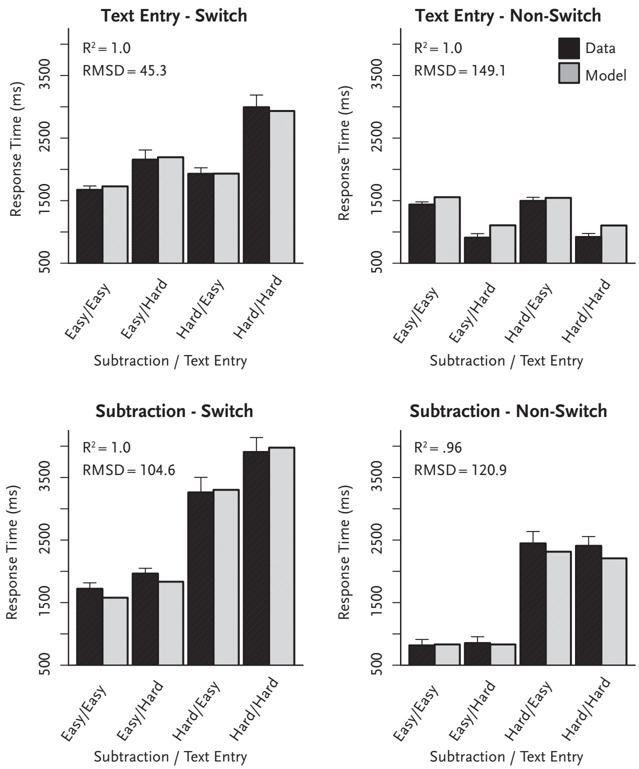

Figure 2.6 Response time data of Experiment 2. RMSD = root mean squared deviation.

The upper panels of Figure 2.6 show the response times on the text-entry task. On the left the switch data are shown, on the right the non-switch data. As predicted, an interaction effect of Subtraction Difficulty and Text-Entry Difficulty was found on the switch trials ($F(1,13) = 19.8$, $p < .001$, $\eta_p^2 = .60$). A simple effects analysis subsequently revealed significant effects of Text-Entry Difficulty when subtraction was easy ($F(1,13) = 27.6$, $p < .01$, $\eta_p^2 = .68$), Text-Entry Difficulty when subtraction was hard ($F(1,13) = 59.2$, $p < .001$, $\eta_p^2 = .82$), Subtraction Difficulty when text-entry was easy ($F(1,13) = 13.2$, $p < .01$, $\eta_p^2 = .50$), and Subtraction Difficulty when text-entry was hard ($F(1,13) = 135.1$, $p < .001$, $\eta_p^2 = .91$). Thus, response times on the switch responses of the text-entry task increased with task difficulty, with an over-additive interaction effect when both tasks were hard. An analysis of the non-switch responses of the text-entry task (upper right panel) showed that only the main effect of Text-Entry Difficulty reached significance, ($F(1,13) = 377.53$, $p < .001$, $\eta_p^2 = .97$). The main effect of Subtraction Difficulty ($F < 1$) and the interaction effect between Subtraction Difficulty and Text-Entry Difficulty ($F(1,13) = 2.4$, $p = .15$, $\eta_p^2 = .16$) were not significant. Note that response times decreased with Text-Entry Difficulty, instead of increasing.

The two lower panels of Figure 2.6 show response times on the subtraction task. The left panel shows the switch responses. An ANOVA revealed a significant interaction effect of Subtraction Difficulty and Text-Entry Difficulty ($F(1,13) = 6.9$,

$p = .02$, $\eta_p^2 = .35$). Subsequent simple effects analyses showed significant effects of Text-Entry Difficulty when subtraction was easy ($F(1,13) = 19.1$, $p < .001$, $\eta_p^2 = .59$), Text-Entry Difficulty when subtraction was hard ($F(1,13) = 14.7$, $p < .01$, $\eta_p^2 = .53$), Subtraction Difficulty when text-entry was easy ($F(1,13) = 104.9$, $p < .001$, $\eta_p^2 = .89$), and Subtraction Difficulty when text-entry was hard ($F(1,13) = 185.5$, $p < .001$, $\eta_p^2 = .93$). Thus, response times on the switch responses of the subtraction task increase with task difficulty, with an over-additive interaction effect, resulting in the highest response times in the *hard–hard* condition. The non-switch response times are shown in the lower right panel of Figure 2.6. Only the main effect of Subtraction Difficulty was significant ($F(1,13) = 305.2$, $p < .001$, $\eta_p^2 = .96$), the main effect of Text-Entry Difficulty and the interaction effect were not significant, Fs < 1. Thus, non-switch response times were lower when the subtraction task was easy.

Accuracy

Figure 2.7 shows the accuracy data of Experiment 2. An ANOVA on the text-entry data shows only a significant main effect of Text-Entry Difficulty ($F(1,13) = 9.7$, $p < .01$, $\eta_p^2 = .43$). The main effect of Switch ($F(1,13) = 2.23$, $p = .16$, $\eta_p^2 = .15$) and Subtraction Difficulty ($F(1,13) = 1.46$, $p = .25$, $\eta_p^2 = .10$) were not significant, neither were the interaction effects between Switch and Subtraction Difficulty (F < 1), Switch and Text-Entry Difficulty ($F(1,13) = 3.29$, $p = .09$, $\eta_p^2 = .20$), Subtraction and Text-Entry Difficulty (F < 1), and the three-way interaction between Switch, Subtraction Difficulty and Text-Entry Difficulty ($F(1,13) = 3.52$, $p = .08$, $\eta_p^2 = .21$). Thus, accuracy on the text-entry task was lower when text-entry was hard.

Along the same lines, an analysis of the subtraction data only revealed a significant main effect of Subtraction Difficulty ($F(1,13) = 40.7$, $p < .001$, $\eta_p^2 = .76$). The main effects of Switch ($F(1,13) = 2.55$, $p = .13$, $\eta_p^2 = .16$) and Text-Entry Difficulty ($F < 1$) did not reach significance, neither did the interaction effects of Switch and Subtraction Difficulty ($F < 1$), Switch and Text-Entry Difficulty ($F < 1$), Subtraction Difficulty and Text-Entry Difficulty ($F(1,13) = 1.53$, $p = .24$, $\eta_p^2 = .11$), or the three-way interaction between Switch, Subtraction Difficulty, and Text-Entry Difficulty ($F < 1$). Again, subtraction accuracy only decreased when the subtraction task became hard.

Model

The model fits well to the response time data (Figure 2.6, grey bars, R^2- and RMSD-values are shown in the graphs). It shows on the one hand the interaction effects in the switch responses, caused by the problem state replacements for each response, and on the other hand no interaction effects in the non-switch responses. Furthermore, it reflects the decrease in response times on the text-entry task non-switch responses, when text-entry was hard (the reason why the model shows these effects is discussed below). The model also follows the accuracy data closely (Figure 2.7): In general capturing the (non-significant) interaction effects in the *hard–hard* conditions, but slightly over-estimating these effects in the text-entry task.

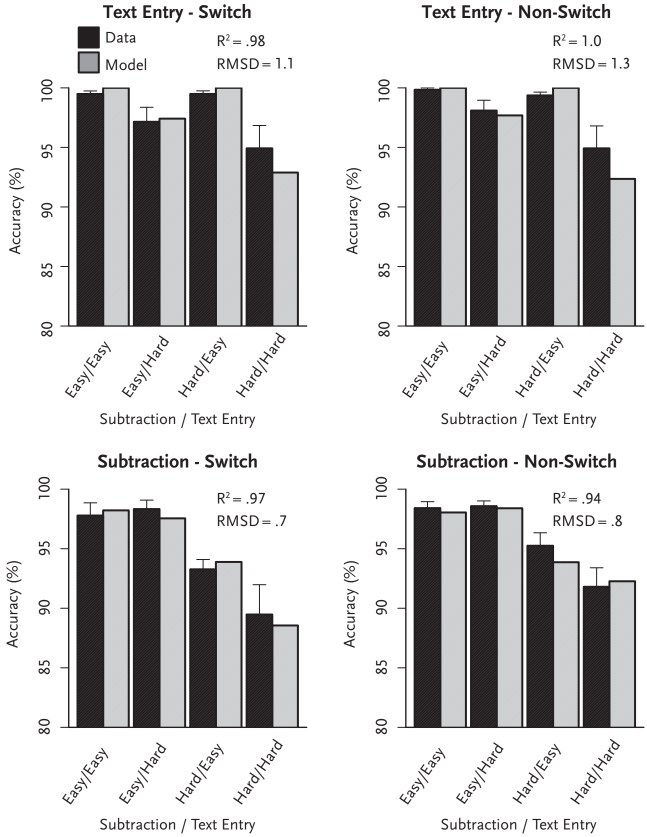

Figure 2.7 Accuracy data of Experiment 2. RMSD = root mean squared deviation.

Discussion

As predicted by the model, an over-additive interaction effect was found on the switch response times in both tasks, but not on the non-switch response times. The model explains this by assuming a problem state bottleneck, requiring the replacement of the problem state in the *hard–hard* condition of the switch responses. In the non-switch responses, on the other hand, the problem state never has to be switched: it is still present from the previous response. A cognitive load account would predict an interaction effect both in the switch and the non-switch responses, because the memory load of the other task is present in both cases (see also Figure 2.5). However, as no interaction effect was observed on the non-switch trials, cognitive load of the other task does not seem to have caused the effects observed in Experiment 1 and in the switch trials in Experiment 2. On the other hand, the model fit shows that a problem state bottleneck account accounts well for the data. Note that we equated cognitive load here with memory load (as for example, Logan, 1979), while there is no consensus in the literature to what exactly constitutes cognitive load. Nonetheless, irrespective of the operationalization of cognitive load, Experiment 2 still gives additional support to a problem state account: When the problem state does not have to be changed, the interaction effect that shows problem state interference disappears.

A second interesting effect is the lower average response time of the hard non-switch text-entry responses, as compared to the easy non-switch responses (upper right panel of Figure 2.6). The model explains this decrease by the fact that in the hard condition it is already known what word has to be entered, and thus also what the next letter is that has to be clicked. Therefore, the model does not have to look at the display of the text-entry task to see what it has to enter, as in the easy version, but can directly search for the correct button and click it. For the switch responses, this decrease in response time is not present, because in that case the model starts the hard text-entry task by retrieving spelling information from declarative memory to determine which letter it has to enter next. On the non-switch responses, the model already initiates the retrieval of the spelling information while clicking the mouse for the previous response, enabling faster responses.

Furthermore, participants were also in general faster on the non-switch responses than on the switch responses. This effect can be explained by the fact that it is necessary to redirect vision and attention to the other task on the other side of the screen on the switch responses, while this is not necessary on the non-switch responses (cf. task switching).

Phonological Loop

Experiment 2 has shown that a memory load probably did not cause the interference effects in the data. However, another possible explanation is that the problem state information in the hard tasks was verbally mediated, and that the phonological loop (e.g., Baddeley & Hitch, 1974) acted as a bottleneck, instead of the problem state resource. That is, if problem state information is rehearsed in the phonological loop, it is possible that there is some overhead in retrieving information when more information has to be rehearsed in the *hard–hard* condition. This alternative account would result in an interaction effect. To test whether the phonological loop is used for storing the problem state information, a third experiment was performed. While Experiment 2 was aimed at the maintenance of the information in working memory without rehearsal, Experiment 3 specifically targets possible rehearsal of the information. In this experiment a listening comprehension task was added to the subtraction and text-entry dual-task, overloading the phonological loop.

Experiment 3: Triple-tasking

For Experiment 3, a listening comprehension task was added to the subtraction and text-entry task: in half of the trials participants had to listen to short stories while performing the other tasks. At the end of a trial, participants had to answer a multiple-choice question about these stories. The experiment has a 2 × 2 × 2 design (Subtraction Difficulty × Text-Entry Difficulty × Listening). Adding a continuous listening task results in the phonological loop being constantly filled with verbal information. If a phonological loop bottleneck was the reason for the interaction effects in Experiments 1 and 2, adding the listening task should produce similar effects in the *easy–hard* and *hard–easy* conditions as we previously saw in the *hard–hard* condition, because now it is also in use by multiple tasks in these conditions. Thus, if the problem state is maintained in the phonological loop, we should now find interference effects as soon as one problem state is stored alongside the information of the listening task (in the *easy–hard* and *hard–easy* conditions). If, on the other hand, the interaction effects were caused by a problem state bottleneck, one would expect the same patterns in the data as found in Experiment 1, with possibly higher response times and error rates over all conditions due to increased cognitive load. Furthermore, our model proposes that, as long as no additional use of the problem state resource is introduced, the problem state bottleneck is independent of the number of tasks and of the amount of cognitive load. Therefore, adding the listening task to the experiment should not influence the results we found previously. To measure baseline performance, the listening task was also tested separately.

Method

Participants

Twenty-three students of the University of Groningen who did not participate in Experiments 1 and 2 participated in Experiment 3 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). A different set of 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 as approved by the Ethical Committee Psychology of the University of Groningen 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 (see the Discussion of Experiment 1). 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, but 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 (M = 30.4, SD = 10.9). The multiple-choice questions consisted of three options. Two example questions are:

> You would like to buy a new washing machine. When will you get a discount?
> A. If you pay cash.
> B. If you buy an extended warranty.
> C. If you buy a dryer as well.

> You are visiting a laboratory with colleagues. What should you do with your lab coat when you leave?
> A. Put it in the yellow container.
> B. Put it in the green container.
> C. Reuse it.

These questions can be answered without making inferences, but do require attention for the complete duration of the story (i.e., the color of the container in the second question is only said once; participants only see the question after they heard the text).

Procedure

The procedure was identical to Experiment 1 if not noted otherwise. In this experiment, participants had to start each trial with the subtraction task. In the listening condition, playback of the story was initiated simultaneously with the presentation of the subtraction task. Thus, the listening task had to be performed concurrently with the subtraction and text-entry tasks. The multiple-choice 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.

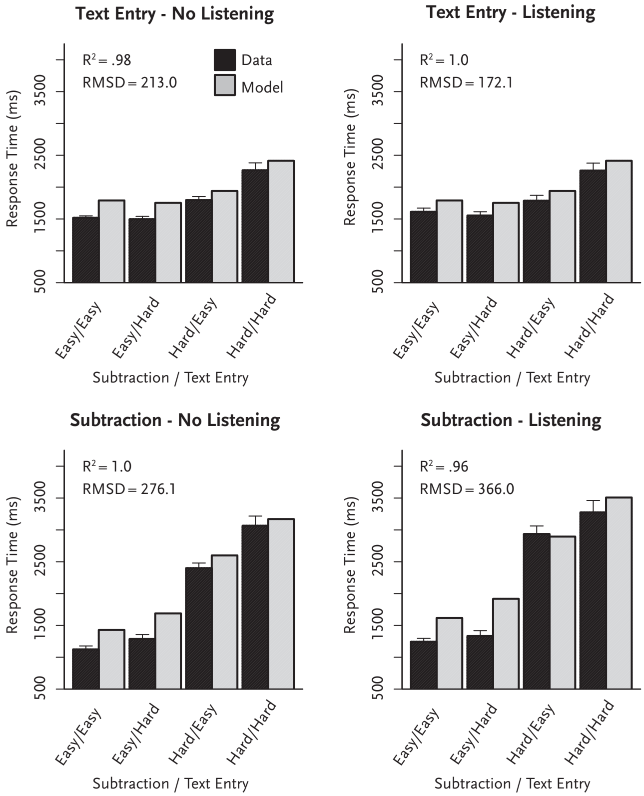

Figure 2.8 Response time data of Experiment 3. RMSD = root mean squared deviation.

Model

The same model as for Experiment 1 was used for the subtraction and text-entry tasks, adjusted for the differences in arithmetic skills between participant groups as was done to calibrate the model to the skill level of the participants in Experiment 2. That is, we adjusted retrieval times of declarative facts and number of incorrect retrievals to match the new group of participants.[7]

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 by Salvucci and Taatgen (2008) to model the classical reading and dictation study by Spelke, Hirst, and Neisser (1976), and by Van Rij, Hendriks, Spenader, and Van Rijn (2009) and Hendriks, Van Rijn and Valkenier (2007) to account for developmental patterns in children's ability to process pronouns. This model is a simplified version of Lewis and Vasishth's model of sentence processing (2005; Lewis, Vasishth, & Van Dyke,

[7] *ACT-R's latency factor and activation noise were again left unchanged (respectively .3 and .1). The activation levels of the three groups of arithmetic chunks of Footnote 4 were adjusted for the new group of participants.*

2006) that constructs syntactic trees for sentence processing. For the current model that kind of linguistic detail is unnecessary, as we are mostly interested in how the tasks influence one another. Thus, it suffices to account for the use of procedural and declarative memory in the listening task.

For each word, the aural module processes the word, four procedural rules fire, and two facts are retrieved from memory, which results in about 320 ms processing time per word, which is fast enough to keep up with the speaking rate of 372 ms per word on average.[8] Because ACT-R's aural module is used to perceive the words, using a phonological loop-based strategy is prevented as this strategy is implemented in ACT-R as a combination of the aural and vocal modules (Huss & Byrne, 2003). No control or executive mechanisms were added to the model: the interleaving of the tasks was left to threaded cognition. Answering the multiple-choice questions was not modeled, as this would have required linguistic processing at a level of complexity that is beyond the scope of this paper.

Results

The same exclusion criteria were used as in Experiment 1 (2.4% of the data was rejected). One question from the listening task was removed, as it was consistently answered incorrectly. If not noted otherwise, analyses were the same as in Experiment 1. Because the stories did not always last for the complete trials of the subtraction and text-entry task, some responses on these tasks were made without participants listening to a story. Therefore, we only took responses into account that were made while the story was present.[9]

Response Times

Figure 2.8, 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 main effect of Listening, nor any interaction effects involving Listening (all *F*s < 1, except for the interaction between Listening and Subtraction Difficulty: $F(1,21) = 1.9$, $p = .18$, $\eta_p^2 = .08$), we collapsed over Listening. The interaction between Text-Entry Difficulty and Subtraction Difficulty was significant ($F(1,21) = 38.78$, $p < .001$, $\eta_p^2 = .65$); a simple effects analysis showed effects of Text-Entry Difficulty when subtraction was hard ($F(1,21) = 37.17$, $p < .001$, $\eta_p^2 = .64$), Subtraction Difficulty when text-entry was easy ($F(1,21) = 30.89$, $p < .001$, $\eta_p^2 = .60$), and Subtraction Difficulty when text-entry was hard ($F(1,21) = 80.60$, $p < .001$, $\eta_p^2 = .79$). Text-Entry Difficulty when subtraction was easy did not reach significance ($F(1,21) = 3.0$, $p = .10$, $\eta_p^2 = .13$). Thus, there was no effect from the listening task on the response times of the text-entry task. Irrespective of the listening task, response times increased

8 *Note that the model is capable of listening to speech faster than 320 ms/word, because the audio module can already start processing the next word while the current word is processed.*

9 *Because this results in an unequal number of observations per cell, we also fitted linear mixed effects models (Baayen et al., 2008). The linear mixed effect models confirmed the ANOVA results. For reasons of consistency, we decided against reporting these additional statistics in the main text, but refer the reader to the Appendix for more details.*

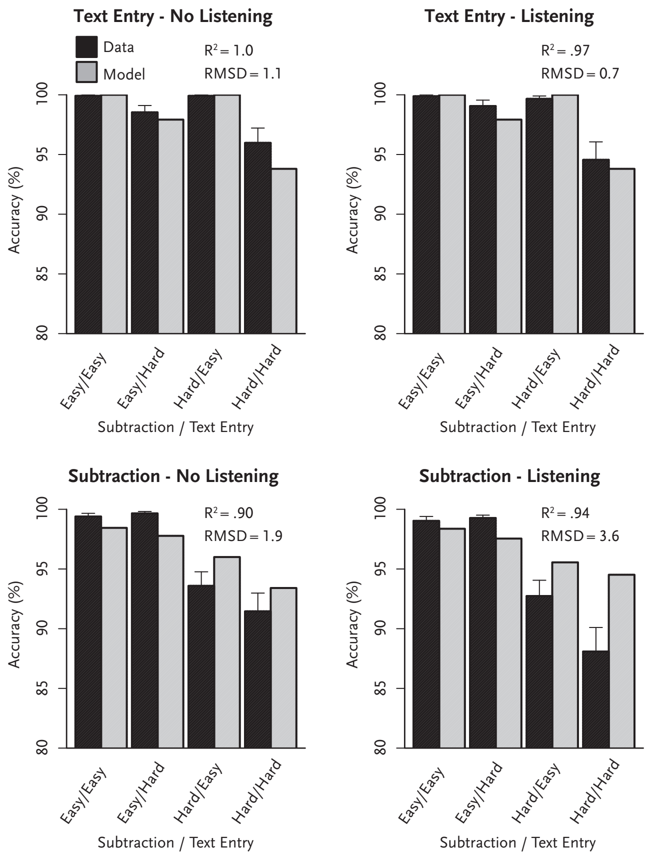

Figure 2.9 Accuracy data of Experiment 3. RMSD = root mean squared deviation.

when subtraction was hard, with an additional increase when text-entry was also hard, resulting in the interaction effect.

The lower panel of Figure 2.8 shows response times on the subtraction task, on the left without, and on the right in combination with the listening task. An ANOVA showed that the three-way interaction between Listening, Subtraction Difficulty, and Text-Entry Difficulty did not reach significance ($F(1,21) = 3.2$, $p = .09$, $\eta_p^2 = .13$), but the main effect of Listening did ($F(1,21) = 4.97$, $p = .04$, $\eta_p^2 = .19$). Furthermore, all two-way interactions reached significance: between Listening and Subtraction Difficulty ($F(1,21) = 9.33$, $p < .01$, $\eta_p^2 = .31$), between Listening and Text-Entry Difficulty ($F(1,21) = 5.98$, $p = .02$, $\eta_p^2 = .22$), and between Subtraction Difficulty and Text-Entry Difficulty ($F(1,21) = 14.3$, $p < .01$, $\eta_p^2 = .40$). A subsequent simple effects analysis of Subtraction Difficulty and Text-Entry Difficulty when the listening task had to be performed revealed significant effects of Text-Entry Difficulty when subtraction was hard ($F(1,21) = 7.12$, $p = .01$, $\eta_p^2 = .25$), Subtraction Difficulty when text-entry was easy

($F(1,21) = 347.1$, $p < .001$, $\eta_p^2 = .94$), and Subtraction Difficulty when text-entry was hard ($F(1,21) = 175.3$, $p < .001$, $\eta_p^2 = .89$). Text-Entry Difficulty when subtraction was easy did not reach significance ($F(1,21) = 3.8$, $p = .07$, $\eta_p^2 = .15$). When the listening task did not have to be performed, all simple effects were significant: Text-Entry Difficulty when subtraction was easy ($F(1,21) = 26.9$, $p < .001$, $\eta_p^2 = .56$), Text-Entry Difficulty when subtraction was hard ($F(1,21) = 27.1$, $p < .001$, $\eta_p^2 = .56$), Subtraction Difficulty when text-entry was easy ($F(1,21) = 337.2$, $p < .001$, $\eta_p^2 = .94$), and Subtraction Difficulty when text-entry was hard ($F(1,21) = 226.7$, $p < .001$, $\eta_p^2 = .92$). Furthermore, Listening had a significant effect when both subtraction and text-entry were easy ($F(1,21) = 4.37$, $p = .05$, $\eta_p^2 = .17$) and when subtraction was hard and text-entry easy ($F(1,21) = 10.5$, $p < .01$, $\eta_p^2 = .33$), but not when subtraction was easy and text-entry was hard or when both tasks were hard (Fs < 1). To summarize, response times on the subtraction task increased when the listening task had to be performed and with task difficulty of the subtraction and text-entry tasks. Furthermore, the effects of Text-Entry Difficulty were smaller when the listening task had to be performed, while the effects of Subtraction Difficulty were larger when the listening task had to be performed (as shown by the two-way interaction effects). An over-additive interaction effect of Subtraction Difficulty and Text-Entry Difficulty was present, both when the listening task had to be performed and when it did not have to be performed.

Accuracy

In Figure 2.9 the accuracy data of Experiment 3 is displayed. The upper panels show the accuracy on the Text-Entry task. As there was neither an effect of Listening, nor any interaction effects involving Listening (all Fs < 1), we collapsed over Listening. The subsequent ANOVA showed an interaction effect of Subtraction Difficulty and Text-Entry Difficulty ($F(1,21) = 6.55$, $p = .02$, $\eta_p^2 = .24$). Three of the four simple effects were significant: Text-Entry Difficulty when subtraction was easy ($F(1,21) = 7.81$, $p = .01$, $\eta_p^2 = .27$), Text-Entry Difficulty when subtraction was hard ($F(1,21) = 33.1$, $p < .001$, $\eta_p^2 = .61$), and Subtraction Difficulty when text-entry was hard ($F(1,21) = 16.0$, $p < .001$, $\eta_p^2 = .43$). Subtraction Difficulty when text-entry was easy did not reach significance ($F < 1$). Thus, accuracy on the text-entry task was lower when text-entry was hard, with an over-additive effect when subtraction was hard as well.

The lower panels of Figure 2.9 show the accuracy data on the subtraction task. Again, there were no significant effects involving Listening (all Fs < 1, except for the main effect of Listening: $F(1,21) = 1.91$, $p = .18$, $\eta_p^2 = .08$), thus we collapsed over Listening. The ANOVA showed a significant interaction effect of Subtraction Difficulty and Text-Entry Difficulty ($F(1,21) = 6.6$, $p = .02$, $\eta_p^2 = .24$). Three simple effects reached significance: Subtraction Difficulty when text-entry was easy ($F(1,21) = 47.2$, $p < .001$, $\eta_p^2 = .69$), Subtraction Difficulty when text-entry was hard ($F(1,21) = 127.4$, $p < .001$, $\eta_p^2 = .86$), and Text-Entry Difficulty when subtraction was hard ($F(1,21) = 10.9$, $p < .01$, $\eta_p^2 = .34$). Text-Entry Difficulty when subtraction was easy was not significant ($F < 1$). Thus, accuracy on the subtraction task was lower when subtraction was hard, and even lower when text-entry was hard as well.

Figure 2.10, left panel, shows the accuracy data of the listening task. The leftmost bar shows the results of the listening baseline experiment (i.e., participants only performed the listening task): 89% correct. Adding the other tasks had little effect, except when both the subtraction and the text-entry task were hard. The interaction between Subtraction Difficulty and Text-Entry Difficulty was significant ($F(1,21) = 7.42$, $p = .01$, $\eta_p^2 = .26$); as were the simple effects of Text-Entry Difficulty when subtraction was hard ($F(1,21) = 9.18$, $p < .01$, $\eta_p^2 = .30$) and Subtraction Difficulty when text-entry was hard ($F(1,21) = 14.75$, $p < .001$, $\eta_p^2 = .41$), driving the interaction effect. The simple effects of Text-Entry Difficulty when subtraction was easy ($F(1,21) = 1.73$, $p = .20$, $\eta_p^2 = .08$) and Subtraction Difficulty when text-entry was easy ($F < 1$) were not significant.

Model

As can be seen in Figure 2.8, the response times of the cognitive model fit well to the human data, both in combination with and without the listening task (R^2 and RMSD values are shown in the graphs). The accuracy data in Figure 2.9 is also accounted for, especially in the text-entry task the model follows the data closely. For the subtraction task the effects are slightly under-predicted: the effects in the data are larger, especially when the listening task is present.

The right panel of Figure 2.10 shows the percentage of words processed by the model. The model can only process words when declarative memory is available. Thus, when words are presented while declarative memory is in use by the other tasks, words cannot be processed, and will be substituted by new words entering the auditory buffer. This happens most often in the *hard–hard* condition, as problem states have to be retrieved from declarative memory for the other tasks on each step of a trial, blocking the resource. Obviously, a percentage of processed words cannot be translated directly into number of correctly answered questions, but the model shows a similar pattern of performance ($R^2 = .68$).

Discussion

In Experiment 3 we added a listening comprehension task to the two tasks used in the previous experiments. The same interaction effects were found as in Experiments 1 and 2, both when the listening task was present and when it was not. Experiment 3 was designed to test whether a problem state bottleneck caused the interference effects, as opposed to a phonological loop bottleneck. If it was a phonological loop bottleneck that caused the interference, overloading the phonological loop by adding the listening task should cause interference effects not only in the *hard–hard* condition, but also in the *hard–easy* and *easy–hard* conditions of the other tasks. The only effect we found that pointed in this direction was the increase of reaction times of the subtraction task when subtraction was hard and the listening task had to be performed (Figure 2.8, lower panel). However, this effect is accounted for in the model by declarative memory interference instead of phonological loop or problem state interference (see below). As the other three conditions did not increase in reaction times, this implies that the

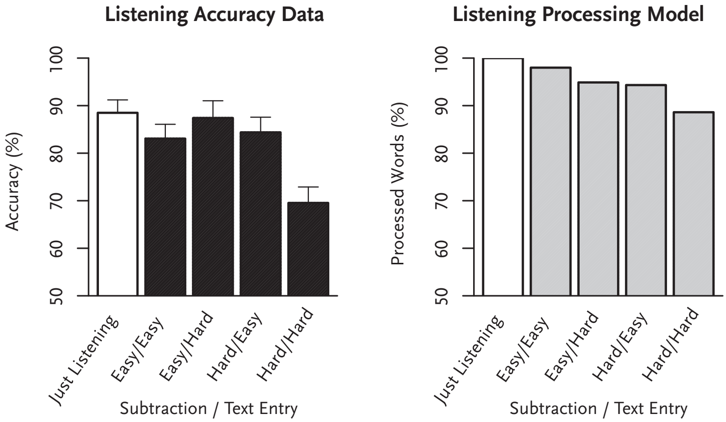

Figure 2.10 Accuracy on the listening task of Experiment 3, and the percentage of processed words by the model per condition.

phonological loop did not cause the interference in Experiment 1 and 2, and provides additional support to a problem state account.

Generally speaking, the listening task had surprisingly little influence on the subtraction and text-entry tasks: the response times only increase by a small amount in the subtraction task. Interestingly, while we did not think about this effect beforehand, and did not model it explicitly afterwards, this increase in response times emerged naturally from our model. A close inspection of the model revealed that it is caused by the continuous use of declarative memory by the listening task. Threaded cognition causes the tasks to be closely interleaved, which means that most of the time there is little interference. However, when the subtraction task needs to use declarative memory when it is in use by the listening task, this will cause a slight delay in execution, causing the increase in response times. This effect is more pronounced when the subtraction task was hard, as shown by the two-way interaction between Listening and Subtraction Difficulty. The model explains this by the need for more declarative retrievals in the hard subtraction task as compared to the easy subtraction task, which leads to more interference with the declarative retrievals of the listening task. For the text-entry task a similar effect would be expected, except for the fact that the text-entry task is much less memory intensive (i.e., less memory retrievals have to be performed) than the subtraction task. That is why the model does not predict an increase in reaction times for the text-entry task, which is consistent with the human data.

The listening task was involved in one more effect: the interaction effect between Listening and Text-Entry Difficulty on the response times of the subtraction task (Figure 2.8, lower panel). That is, the effect of Text-Entry Difficulty was smaller when the listening task had to be performed. An opposite effect would have been expected when the interference effects of Experiment 1 and 2 were caused by a phonological loop bottleneck, because in that case, the phonological loop would have caused interference in combination with only one hard task, as explained above. The current model does not account for the interaction between Listening and Text-Entry Difficulty that was observed in the data. However, as this paper focuses on the effects of the problem state manipulations, and the purpose of the experiment was to rule out any auditory loop related explanation (which would have caused an opposite effect), we decided against adding a post-hoc explanation to the model.

The effects of the subtraction and text-entry task on listening comprehension were also surprisingly small: a decrease in listening accuracy scores was observed only when both other tasks were hard. The model explains this finding by the assumption that declarative memory is in high demand by the subtraction and text-entry tasks when both these tasks are hard, because problem states have to be retrieved from declarative memory on each step of a trial. Therefore, a word is sometimes replaced by the next presented word in the auditory buffer before it is processed using declarative memory. As there is not sufficient time in the *hard–hard* condition to process all words, this will presumably result in more mistakes on the listening comprehension task in this condition.

In conclusion, our threaded cognition model proposed that adding this particular third task should not influence the results of the other tasks dramatically. This turned out to be the case, even while the continuous listening task is, arguably, quite demanding. The patterns in the data were comparable to the data of the previous experiments, while the small increase in response times was explained by the increased use of declarative memory.

General Discussion

In this paper, we tested the hypothesis that the problem state resource acts as a bottleneck in multitasking. Experiment 1 consisted of two tasks that had to be carried out concurrently, both with and without a problem state. This resulted in an over-additive interaction effect of task difficulty (i.e., the requirement of two problem states led to higher response times), confirming the hypothesis. In Experiments 2 and 3, we tested whether this interaction effect was due to cognitive load or to a phonological loop bottleneck, respectively, instead of to a problem state bottleneck. Experiment 2 showed that the interaction effect is not due to a simple memory load effect, but instead is related to a switch of task context. This corroborates the problem state hypothesis. In Experiment 3, the phonological loop was overloaded by adding a story comprehension task. This did not have a major influence on the effects found in Experiment 1, lending additional support to a problem state bottleneck account of the data. Based on these three experiments and general ACT-R assumptions about memory, modularity and performance, we conclude that the problem state resource indeed acts as a bottleneck when it has to be used by multiple tasks concurrently.

Nevertheless, it should be possible to formulate alternative models explaining these data sets, and we therefore cannot claim the data prove the existence of a problem state bottleneck. The strength of the current account over any post-hoc fit of the data is that we tested an a priori prediction made by the threaded cognition theory before running the experiment. First, we ran Experiment 1 to test a qualitative prediction of a problem state bottleneck. Without having to add additional assumptions to the model, it accounted for the interference effects that we found. Subsequently, we tested the two most plausible alternative accounts of the data in Experiments 2 and 3. With the same basic model as used for Experiment 1, we were able to account for the data of these experiments. Thus, based on a theory-driven model we were able to predict the effects of Experiment 1, and could subsequently account for data of two related experiments.

The Problem State and Working Memory

There is a relation between the problem state and the classical notion of working memory (e.g., Baddeley, 1986; Baddeley & Hitch, 1974): both are used for temporarily maintaining mental representations. In the ACT-R architecture, working memory does not exist as a separate system. Instead, working memory is represented by a combination of (a) the contents of the declarative memory buffer and the problem state buffer and (b) highly active chunks in declarative memory (Anderson, 2005; Daily, Lovett, & Reder, 2001; Lewis & Vasishth, 2005; Lovett, Daily, & Reder, 2000). In this scheme, the buffer contents are accessible at no time cost, and thus constitute directly accessible 'true working memory': the 'focus of attention' (e.g., Cowan, 1995; Garavan, 1998; Oberauer, 2002). With a size of two this is comparable to theories positing an extremely limited working memory size (e.g., Garavan, 1998; McElree, 2001). On the other hand, highly active chunks in declarative memory are accessible, but at a small time cost. If these items have just been added, as in common working memory or immediate memory experiments, the number that can be reliably retrieved is around 4 to 9 (e.g., Anderson et al., 1998; Oberauer & Lewandowsky, 2008). This is more comparable to theories with a working memory size of 4 to 9 (e.g., Cowan, 2000; Miller, 1956; Morey & Cowan, 2004). The combination of (a) having a small amount of directly accessible items and (b) a number of easily accessible items at a small time cost, is in line with a number of recent theories (e.g., Jonides et al., 2008; McElree, 2001; Oberauer, 2002).

The problem state acts in this framework as the location where new information is stored. This new information can either originate from perceptual processes (the to-be-entered word in the text-entry task), from a result of processing existing information (the carry-flag in the subtraction task), or from changing existing information (for example, processing $2x + 5 = 8$ to $2x = 3$). This is potentially important for many dual- or multitask situations, as it is often necessary to maintain new information. As we have argued that the problem state acts as a bottleneck, this could have a considerable influence on tasks in which multiple sets of information have to be maintained at the same time.

Threaded Cognition

While our hypothesis was inspired by the threaded cognition theory, one could wonder whether threaded cognition is a necessary part of the models. It is indeed possible to think of a way of modeling our experiments using only ACT-R. The merit of threaded cognition, though, is that we were not forced to come up with a supervisory control structure to model the tasks.[10] This would have been possible, as is best shown by existing multitasking models without threaded cognition (e.g., Anderson, Taatgen, & Byrne, 2005; Salvucci, 2006; Taatgen, 2005). However, these models represent all tasks in a single goal representation. This is hard to defend if the tasks in the experiment are tasks that the participants are already proficient in, like driving or multi-column subtraction. The importance of using threaded cognition is that the existing threads can be reused if, for instance, the subtraction task would be combined with driving. This seems to be the way humans would handle this: our participants would seemingly not have to learn a new supervisory control structure if they had to solve subtraction problems while driving (see, for a more in-depth discussion of this issue, Kieras et al., 2000; Salvucci, 2005; Salvucci & Taatgen, 2008). Case in point is our Experiment 3, in which we were able to add an additional thread for the listening task, without having to change anything in the existing model. The interaction of the three threads, without any supervisory control, turned out to be a good predictor of the participants' behavior: Even the slight increase in reaction times in the listening condition was accounted for by the model, while not predicted by the authors beforehand.

One potential criticism of threaded cognition is that it allows for an unlimited set of goals that is not susceptible to decay. Altmann and Trafton (2002) have successfully argued against the construct of the goal stack, which had the same problem. Instead, Altmann et al. (Altmann & Gray, 2008; Altmann & Trafton, 2002) have proposed that only a single goal can be active at a time, and multiple goals have to be handled by swapping out the current goal with goals retained in declarative memory. Salvucci and Taatgen (2008), however, found that such a procedure would be too slow to account for certain Psychological Refractory Period experiments. The model we have presented here is consistent with both the Salvucci and Taatgen approach, as well as with the Altmann et al. approach. Instead of swapping out the goal as such, though, the contents of the problem state resource are swapped out. The main difference is our assumption that not all tasks require a problem state. Although we have not applied the strategic encoding strategies that Altmann et al. use in their models, this would certainly be possible if a task would necessitate it (see also Salvucci, Taatgen, et al., 2009).

Single vs. Multiple Bottlenecks

In this article we have introduced the notion of multiple bottlenecks, and it is therefore useful to contrast it with a single bottleneck approach (e.g., Pashler, 1994). Explaining multitasking interference with multiple bottlenecks can be considered as a refinement of a single bottleneck account. Single-bottleneck accounts consider central cognition as a uniform system that can only be engaged in a single action at a time. Although this offers accurate accounts of many combinations of simple tasks, the more complex tasks discussed in this article need a more refined theory. Multiple-bottleneck models

[10] *One could argue that executive control plays a role in threaded cognition, the threads act in a 'greedy and polite' way after all. However, this is a task-unspecific form of executive control, not customized for the tasks at hand, and not influencing the interleaving of the tasks directly.*

allow parallel processing of certain combinations of tasks, as long as they use different central resources. In Experiment 3, for example, the listening task almost continuously engages central cognition, but because central cognition is subdivided into separate resources it is still possible to do the other tasks by properly interleaving them, resulting in only a minor impact on performance.

The three bottlenecks or resources we focus on in this article have different impacts on performance, which is mainly due to the time scale on which they operate. Interference in the fastest system, the procedural resource is usually very limited and in the order of tens of milliseconds, and therefore hardly noticeable in the experiments discussed here (although it is in perfect time sharing experiments, Salvucci & Taatgen, 2008, and attentional blink experiments, Taatgen et al., 2009). Interference in the declarative memory resource is usually limited to the maximum duration of a memory retrieval, which is never more than a couple of hundreds of milliseconds in our experiments. This produces small amounts of interference, especially noticeable in the listening task in Experiment 3. The problem state resource, finally, can produce considerable interference, because threads need this resource over longer periods of time. Using threaded cognition it is possible to predict quantitatively how much two tasks will interfere with each other. Single bottleneck models usually do not deal with experiments in which a problem state needs to be maintained over longer periods of time, but nevertheless the problem state behaves like a bottleneck in the same way as procedural bottlenecks in for example perfect time-sharing experiments. We therefore do not see multiple bottlenecks as a refutation of the single bottleneck theory, but rather as a refinement in the details and an extension in time scale.

Implications of a Problem State Bottleneck

Why is the problem state bottleneck important for real life situations? A clear example can be found in our previous research, in which participants had to steer a simulated car and operate a navigation device at the same time (Borst & Taatgen, 2007). It was shown that as soon as participants had to use a problem state for both tasks, their performance decreased considerably. This can be tied back to real life: as soon as information is not readily available in the world, performance levels will decrease if two tasks require the maintenance of intermediate information. Thus, it is preferable to have at most one task that requires the use of a problem state in a multitasking situation. As a design guideline this means that, for example in cars, a secondary device should present its information to the user, instead of requiring the user to maintain intermediate representations.

However, if there is an ongoing task that requires the use of problem representations, and it is known that it will be interrupted (including self-interruptions), human-computer interface designers should try to ensure that the task is interrupted at a point without a problem state. If that is not possible, the user should at least be given the opportunity to rehearse the problem state before the task is suspended. For example, when your work is interrupted by a phone call, most people would let the telephone ring a couple of times before picking it up, and only interrupt their work at a point where it is easy to resume it afterwards. Trafton, Altmann, Brock, and Mintz (2003) formally showed this effect: if users were warned 8 seconds before their task was interrupted, they were significantly faster in resuming the original task than users who were interrupted without a warning. According to our problem state bottleneck theory, the warning gave users the opportunity to rehearse their problem state before being interrupted, while that was impossible in the non-warning condition, enabling faster resumptions after the interruptions (see Salvucci, Taatgen, et al., 2009, for simulations of this experiment).

Conclusion

In summary, the three experiments showed that the problem state resource acts as bottleneck in multitasking. Because the intermediate representations that are stored as a problem state often have to be maintained for several seconds or more, this bottleneck can result in considerable interference between tasks, and therefore has to be taken into account when designing environments for multitasking.

Appendix: More Detailed Analysis of Experiment 3

In this appendix we discuss an alternative analysis of the data of Experiment 3. Because the stories in Experiment 3 did not always last for the complete trials of the subtraction and text-entry task, some responses on these tasks were made without participants listening to a story. Therefore, we only took responses into account that were made while the story was present. Because this results in an unequal number of observations per cell, we also fitted linear mixed effects models to analyze the data (e.g., Baayen, Davidson, & Bates, 2008). As the results of the linear mixed effect models are very similar to those of the ANOVAs, we have included the ANOVA results in the body of the manuscript for reasons of consistency. The results of the linear mixed effects models are reported here.

Response Times

Figure 2.8, upper panels, shows response times on the text-entry task, on the left without and on the right in combination with the listening task. A linear mixed effects model was fitted to the response time data, with Listening, Subtraction Difficulty, and Text-Entry Difficulty as fixed effects and Subject as a random effect. The model shows significant contributions of Listening ($\beta = 88.83$, $t(7836) = 2.84$, $p < 0.01$), Subtraction Difficulty ($\beta = 278.1$, $t(7836) = 9.30$, $p < 0.001$), the Listening × Subtraction Difficulty interaction ($\beta = -142.7$, $t(7836) = -3.03$, $p < 0.01$), and of the Subtraction Difficulty × Text-Entry Difficulty interaction ($\beta = 489.7$, $t(7836) = 11.57$, $p < 0.001$). Comparing this model to a model without the interaction between Subtraction Difficulty and Text-Entry Difficulty shows that the first model is to be preferred ($\chi^2(2) = 206.1$, $p < 0.001$), indicating a significant contribution of the interaction term. Thus, response times were higher when the listening task had to be performed and when the subtraction task was hard, while the combination of the listening and the hard subtraction task caused response times to decrease. Most importantly, an over-additive interaction effect of Subtraction Difficulty and Text-Entry Difficulty was found, irrespective of the listening task. The difference with the ANOVA results reported in the main text is the small influence of the listening task on response times. Although this would argue against collapsing over the Listening conditions for the ANOVA, collapsing over Listening does not change the main outcome of the analysis, nor does it influence our conclusions. Therefore, we opted to keep the main text a consistent whole, and collapsed over Listening.

The lower panels of Figure 2.8 show response times on the subtraction task. Again, a linear mixed effects model was fitted to the data. Listening, Subtraction Difficulty, and Text-Entry Difficulty were added as fixed effects, while Subject was entered as random effect. All main effects contributed significantly to the response times: Listening ($\beta = 112.5$, $t(7854) = 2.51$, $p = 0.01$), Subtraction Difficulty ($\beta = 1276$, $t(7854) = 29.37$, $p < 0.001$), and Text-Entry Difficulty ($\beta = 164.8$, $t(7854) = 3.82$, $p < 0.001$), as did all interaction effects except Listening × Text-Entry Difficulty: Listening × Subtraction Difficulty ($\beta = 344.0$, $t(7854) = 5.1$, $p < 0.001$), Subtraction Difficulty ×

Text-Entry Difficulty ($\beta = 494.8$, $t(7854) = 8.0$, $p < 0.001$), and the three-way interaction of Listening × Subtraction Difficulty × Text-Entry Difficulty ($\beta = -356.9$, $t(7854) = -3.68$, $p < 0.001$). A comparison between this model and a model without the Subtraction Difficulty × Text-Entry Difficulty interaction showed that the first model fits better to the data ($\chi^2(2) = 67.34$, $p < 0.001$), indicating a significant contribution of the Subtraction Difficulty × Text-Entry Difficulty interaction. Comparing this model to a model without the three-way interaction showed that the first model again fits the data better ($\chi^2(1) = 13.1$, $p < 0.001$), thus, also the three-way interaction contributes significantly to the model. This means that response times were higher when the listening task had to be performed, when the subtraction task was hard, and when the text-entry task was hard; and that the effect of subtraction difficulty is larger in the presence of the listening task. Furthermore, there is a significant interaction between Subtraction Difficulty and Text-Entry Difficulty, which is larger without the listening task than with the listening task. The main difference with the results of the ANOVA is the significant three-way interaction between Listening, Subtraction Difficulty, and Text-Entry Difficulty. The three-way interaction shows that the effect of the interaction between Subtraction Difficulty and Text-Entry Difficulty is smaller when the listening task has to be performed. However, even in the presence of the three-way interaction, the two tasks still interact, which is in accordance with our modeling results.

Accuracy

In Figure 2.9 the accuracy data of Experiment 3 is displayed. The upper panel shows the accuracy on the text-entry task. A binomial linear mixed effects model was fitted to the data with Listening, Subtraction Difficulty, and Text-Entry Difficulty as fixed effects, and Subject as a random effect. It shows only a significant effect of Text-Entry Difficulty ($\beta = -2.9$, $z(7836) = -2.74$, $p < 0.01$). Thus, accuracy on the text-entry task was lower when the text-entry task was difficult. The ANOVA reported in the main text also found a significant interaction between Subtraction Difficulty and Text-Entry Difficulty.

The lower panels of Figure 2.9 show the accuracy data on the subtraction task. A binomial mixed effects model with Listening, Subtraction Difficulty, and Text-Entry Difficulty as fixed effects and Subject as a random effect showed that only Subtraction Difficulty contributes significantly to the model ($\beta = -2.4$, $z(7854) = -6.1$, $p < 0.001$). Thus, accuracy on the subtraction task decreased with Subtraction Difficulty. Again, the ANOVA also found a significant interaction effect of Subtraction Difficulty and Text-Entry Difficulty.

Figure 2.10, left panel, shows the accuracy data of the listening task. The leftmost bar shows the results of the listening baseline experiment (i.e., participants only performed the listening task): 89% correct. Adding the other tasks had little effect, except when both the subtraction and the text-entry task were hard. Fitting a binomial linear mixed effects model with Subtraction Difficulty and Text-Entry Difficulty as fixed effects and subject as random effect, shows a significant interaction effect between Subtraction Difficulty and Text-Entry Difficulty ($\beta = -1.2$, $z(507) = -2.55$, $p = 0.01$). However, because the stories lasted sometimes longer than the other two tasks, parts of the stories were attended without performing the other tasks. If we add the proportion overlap between the stories and the other tasks to the linear model, this does not significantly improve the first model ($\chi^2(4) = 3.74$, $p = .44$). Thus, adding the overlap did not change the outcome of the analysis, leaving only a significant decrease in accuracy when both the subtraction and the text-entry task were hard.