Bypassing the Problem State Bottleneck

In which we show that the problem state bottleneck causes an increase in mental workload and that it can be bypassed by presenting information in the environment.

The experimental data in this chapter was previously published in:

Buwalda, T.A., Borst, J.P., Taatgen, N.A., & Van Rijn, H. (2011). Evading a multitasking bottleneck: Presenting intermediate representations in the environment. In L. Carlson, C. Hölscher & T. Shipley (Eds.), *Proceedings of the 33st Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

3

Chapter

Abstract

Objective: *In this paper we investigate whether external support can prevent negative effects of the problem state bottleneck in human multitasking.*

Background: *Previously, it was shown that the problem state resource – a central element in working memory that maintains current task information – can only be used for one task at a time. When the problem state resource is required for multiple tasks concurrently, performance decreases.*

Method: *To see whether external support reduces the effects of the problem state bottleneck, we measured performance and pupil dilation (to assess mental workload) during an experiment that manipulated the use of the problem state resource.*

Results: *It was shown that the effects of the problem state bottleneck on response times and accuracy diminished when problem state information was presented externally. However, we did not find a difference in mental effort between the two conditions. A cognitive model was used to show that the participants behaved rationally: they only used the information in the environment when more than one problem state was required to do the task, thus only when it improved their performance.*

Conclusion: *We conclude that external support can be used to bypass the problem state bottleneck, but that external support is only beneficial when multiple problem states are required to do a task.*

Application: *These results should be taken into consideration when designing interfaces and tasks: users should at most need a single mental representation to carry out a task. Otherwise, response times and number of errors will increase.*

Introduction

Multitasking is all around us. González and Mark (2004) have shown that people switch on average every 3 minutes between tasks in a typical office environment. In addition, a recent study showed that every generation 'multitasks' more than the previous generation in their free time (Carrier et al., 2009). However, it is also well known that performance on individual tasks suffers from multitasking. In the field of sequential multitasking (i.e., switching between tasks, Salvucci, Taatgen, et al., 2009), theorists have focused on the disruptive effects of interruptions (e.g., Gillie & Broadbent, 1989; Monk et al., 2008). Likewise, the concurrent multitasking literature has identified several processing bottlenecks that lead to decreased performance when two tasks are performed at the same time (e.g., Broadbent, 1958; Keele, 1973; Pashler, 1994; Salvucci & Taatgen, 2008; Wickens, 1984, 2002). One cause of multitasking interference, both in concurrent and sequential multitasking, is the problem state bottleneck (Borst, Taatgen, & Van Rijn, 2010).

The problem state is defined as the element in working memory that can be used without any time cost (Anderson, 2005), unlike other elements in working memory (see e.g., McElree, 2001). It is used to represent intermediate information in a task, for example, '$3x = 15$' when solving '$3x - 5 = 10$'. Previously, we have shown that the problem state resource can contain at most one chunk of information, and therefore causes multitasking interference when required by multiple tasks at the same time (Borst, Taatgen, & Van Rijn, 2010). In a dual-task paradigm, participants needed a problem state for none, one, or both of the tasks. In the condition where subjects needed a problem state for both tasks, performance decreased considerably both in reaction times and accuracy as compared to the other conditions. Supported by a cognitive model, this was taken as an indication of a problem state bottleneck. Further evidence for a problem state bottleneck was provided by Salvucci and Bogunovich (2010), who showed that when subjects had to switch between an e-mail and a chat task, they chose switch points at which they did not have to maintain a problem state.

Given that the problem state bottleneck can lead to a decrease in performance, both in laboratory and real-life settings, we investigated how this bottleneck can be bypassed. In this article we describe an experiment in which participants had to perform two tasks at the same time. The first condition of the experiment is a replication of our previous study (Borst, Taatgen, & Van Rijn, 2010), and should result in problem state interference. In the other condition, we presented supporting information on the screen, thereby offloading possible internal representations to the environment (e.g., Kirsh, 1995). We hypothesized that the interference effects disappear in this condition. A second question that we address in this article is whether the problem state bottleneck causes an increase in mental workload, and whether this possible increase disappears with external support. To assess the level of mental workload during the experiment we measured pupil dilation (e.g., Beatty, 1982). Finally, to show that a problem state bottleneck can account for the observed behavior, we present a computational cognitive model. In the remainder of this article, we first describe the used methodology, followed by the results of the experiment, the model description and results, and a general discussion.

Method

In the experiment, based on earlier experiments by Borst, Taatgen, and Van Rijn (2010), 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 in which participants had to maintain a problem state from one response to the next. In the current paper we extended the original setup with a condition in which the problem state of the subtraction task is displayed on the screen (the support condition), reducing the need for mentally maintaining a problem state in the hard subtraction condition. Thus, the experiment had a 2 × 2 × 2 factorial within-subjects design (Subtraction Difficulty × Text-Entry Difficulty × Support).

Pupil Dilation

To assess mental workload, pupil dilation was measured throughout the experiment. Since the 1960s, pupil size is known to reflect mental activity (e.g., Hess & Polt, 1964) and memory load (e.g, Kahneman & Beatty, 1966). In 1982, based on a large body of research, Beatty argued that pupil dilation could be used as a physiological measure of mental effort, because it reflects “within-task, between task, and between-individual variations in processing demands” (Beatty, 1982, p. 276; see for a more recent review, Steinhauer & Hakerem, 1992). From an applied perspective, Iqbal and colleagues used pupil dilation to study mental workload in a route planning and in a document editing task (Iqbal, Adamczyk, Zheng, & Bailey, 2005). The use of pupil dilation allowed them to track mental workload throughout the tasks, and identify opportunities to interrupt users on points of low workload. In the current task we measured pupil dilation to see if the decrease in performance when participants have to use multiple problem states concurrently is linked to an increase in mental effort, and if this possible increase disappeared when participants receive external support in the subtraction task.

Participants

Thirty-three students of the University of Groningen participated in the experiment for course credit or monetary compensation of €10. Four participants were rejected because they scored less than 75% correct where the other participants scored >95% correct. Two participants were rejected because they did not adhere to task instructions, and three because of recording problems of the eye tracker. This leaves 24 complete datasets (17 female, age range 18-43, mean age 20.5). 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.

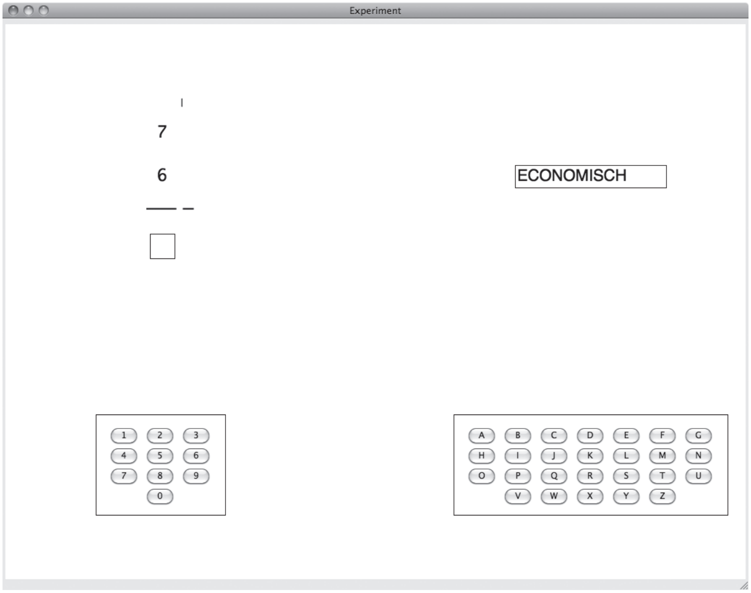

Figure 3.1 The experiment in the support condition. The '|' indicates that there is currently no carry, it will turn into a '1' when a carry has to be processed. Note that in the real experiment one of the tasks would be disabled at any given moment.

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 3.1). Participants had to alternate between the two tasks: after entering 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 3.1. Participants had to solve 10-column subtraction problems in standard right to left order. However, at each point in time, only a single column was visible. Although the problems were presented column by column, the participants were instructed that the separate columns in a trial were part of a 10-column subtraction problem (in the practice phase participants started out with a normal 10-column layout, only later they switched to solving the problems column by column). Participants had to enter the digits by clicking on the on-screen keypad with the mouse. In the easy, no problem state version, carrying was never needed because the upper digit was always larger or equal to the lower one. In contrast, the hard version required participants to carry six times out of 10 possible columns. The assumption is that participants use their problem state resource to store whether a carry is in progress.

The interface for the text-entry task is shown on the right in Figure 3.1. Participants had to enter 10-letter strings by clicking on the on-screen keyboard. In the easy version these strings were presented one letter at a time and participants had to click the corresponding button on the keyboard. In the hard version, a 10-letter word was presented once at the start of a trial. Once a participant clicked on the first letter, the word disappeared and the remaining letters had to be entered one at a time, without feedback. Thus, after the initial presentation of the word in the hard condition, participants could neither see what word they were entering, nor what they had already entered.

Because participants had to alternate between the two tasks after every response, they had to keep track of whether a carry was in progress for the subtraction task and what the word was for the text-entry task while performing the other task.

In the support condition a marker on the screen indicated whether a carry was in progress in the subtraction task. Figure 3.1 shows this condition. The '|' indicates that there is currently no carry in progress. However, as soon the previous column resulted in a carry (e.g., after a column like 3 - 4), the '|' turned into a '1'. Thus, in the support condition it was not necessary to keep track of the problem state mentally: when a '1' was shown on screen, there was a carry in the previous column.

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 carries, 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-frequency Dutch words (CELEX database; Baayen et al., 1993) to ensure that similarities between words were kept at 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 were presented pseudo-random sequences of letters that they had to enter one-by-one in the easy condition. By scrambling the words, we controlled for letter-based effects, while preventing the use of strategies to predict the next letter.

The experiment was presented full screen on a 20.1" monitor. Participants were sitting at a normal viewing distance, about 70 cm from the screen. For recording pupil dilation an Eyelink 1000 table-mounted eye tracker of SR Research was used. We recorded one eye, with a sampling frequency of 500 Hz. To improve measurements, participants were seated with their heads positioned in a padded head- and chin-rest.

Procedure

Each trial started with the presentation of a calibration circle for the eye tracker. After the calibration circle a fixation cross was presented for 6 seconds, to allow pupil dilation to return to baseline. The fixation cross was followed by two horizontally aligned colored circles representing the tasks. The color of the circles indicated the difficulty levels of the tasks (on the left for the subtraction task, on the right for the text-entry task; green for easy, red for hard). The circles stayed on the screen for 1 second, followed by a fixation cross for 600 ms, after which the subtraction and text-entry tasks appeared. Participants had to begin with the subtraction task, and then alternate between the two tasks. After completing both tasks, a feedback screen was shown for 2 seconds, indicating how many letters and digits were entered correctly. Before the next trial started, a fixation screen was shown for 2 seconds.

The experiment consisted of a practice block and two experimental blocks. One of the experimental blocks contained the support condition; the order was counter-balanced over participants. The practice block consisted of 12 single task trials (4 subtraction trials with 10 columns visible, 4 subtraction trials with one column visible, and 4 text-entry trials), followed by a block of 4 multitasking trials: all combinations of subtraction and text-entry (*easy-easy*, *hard-easy*, *easy-hard*, and *hard-hard*). Both experimental blocks consisted of 28 multitasking trials. Before the second block the subtraction task was practiced again, to familiarize the participants with using the carry indicator if they did not use this in the first block, or with performing the task without the indicator in the other case. Subtraction and text-entry conditions were randomized within a block. The complete experiment consisted of 56 experimental trials, and lasted for about 90 minutes. In between blocks participants could take a short break. At the start of the multitask practice block and the two experimental blocks the eye tracker was (re) calibrated.

Results

We will discuss the results of the experiment on the basis of our experimental questions. First, we will discuss how the No-Support condition gives experimental support for a problem state bottleneck. We will then turn to the Support condition, to see if the effects of the problem state bottleneck disappear for the subtraction task when external support is provided. Finally, we will discuss mental workload.

We only analyzed the data from the experimental phase of the task. A response time in the subtraction task is defined as the time between a response in the text-entry task and a response in the subtraction task; a response time in the text-entry task as the time between a response in the subtraction task and a response in the text-entry task. First responses of each trial were removed. Outliers were removed from the data (RTs < 250 ms or > 10,000 ms), after which we removed data exceeding three standard deviations from the mean per condition per participant (in total 2.2% of the data was removed). All *F*- and *p*-values are obtained from repeated measure ANOVAs; all error bars depict standard errors. Accuracy data were transformed using an arcsine transformation before being submitted to the ANOVA. We will not discuss all effects in the text, but only the ones relevant to our questions. However, all ANOVA results are listed in Tables 3.1, 3.2, 3.3, 3.4, and 3.5.

The Problem State Bottleneck: Replication

The No-Support condition of the current experiment is a replication of Experiment 1 in Borst, Taatgen, & Van Rijn (2010), which was the first in a series of three experiments that we used to argue in favor of a problem state bottleneck. The current results replicate

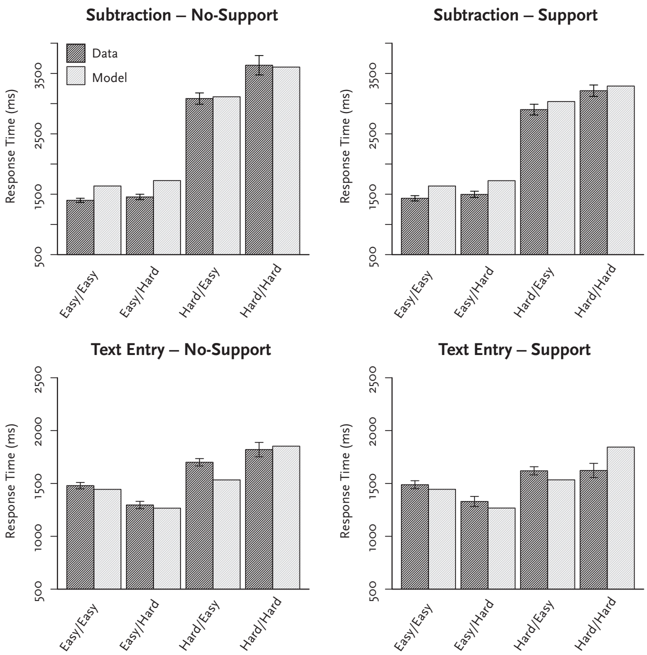

Figure 3.2 Response times. Easy/Hard means Easy Subtraction – Hard Text-Entry, etc. Error bars are standard errors.

the effects in the original data. The left panels of Figure 3.2 show the response times in the No-Support condition (ANOVA results are listed in Table 3.1). On top the response times on the subtraction task are shown. First, we see a large increase in response times with Subtraction Difficulty: when subtraction was hard, response times were much higher than when subtraction was easy. More interestingly, when both tasks were hard, there was an additional increase in response times, as shown by a significant over-additive interaction effect between Subtraction Difficulty and Text-Entry Difficulty. This interaction effect was taken as an indication of a problem state bottleneck: when participants had to maintain a problem state for both tasks, response times increased considerably as compared to when they had to maintain a problem state for only one task.

A similar effect can be seen in the response times on the text-entry task (Figure 3.2, left side, lower panel). Here, response times were lower when text-entry was hard than when text-entry was easy (we discuss this effect in the model section below). However, response times increased when subtraction was hard as well: the *hard-hard* condition. Again, because an additional problem state is required in the other task, we see an increase in response times on the current task. Statistically, this is shown by a significant interaction effect between Subtraction Difficulty and Text-Entry Difficulty.

The accuracy data of both tasks also show this effect, as shown in Figure 3.3, left panels (ANOVA results in Table 3.2). While accuracy naturally decreases with task

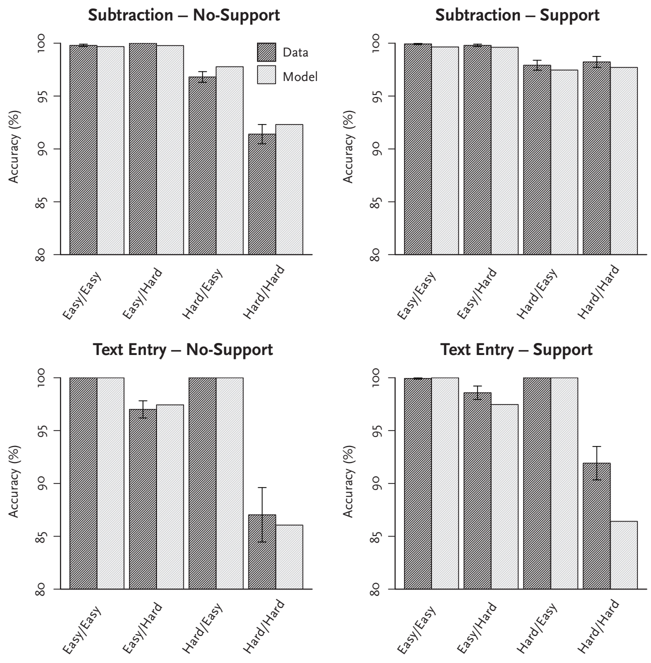

Figure 3.3 Accuracy. Easy/Hard means Easy Subtraction – Hard Text-Entry, etc. Error bars are standard errors.

difficulty of the task itself, it decreases even more when the other task is hard as well (shown by significant interaction effects between Subtraction Difficulty and Text-Entry Difficulty).

Summarizing, we see that response times increase and accuracy decreases when participants had to maintain two problem states as compared to zero or one. Previously, these interaction effects were taken as an indication of a problem state bottleneck. The question is now whether these interaction effects disappear in the subtraction task when external support is provided.

External Support: Bypassing the Bottleneck

The right panels of Figure 3.2 show the response times in the Support Condition. With respect to the response times on the subtraction task, a significant three-way interaction between Support, Subtraction Difficulty, and Text-Entry Difficulty (Table 3.3) shows that the two-way interaction between Subtraction Difficulty and Text-Entry is smaller in the Support condition than in the No-Support condition. Thus, participants were faster in the *hard-hard* condition with Support than without Support. However, as can be seen in Figure 3.2 and Table 3.1, the two-way interaction between Subtraction Difficulty and Text-Entry Difficulty was also significant with Support: even with external support

Table 3.1 ANOVA results of the response time data; separate for Support and No-Support.

Source	RT No-Support $F(1,23)$	RT No-Support p	RT No-Support η_p^2	RT Support $F(1,23)$	RT Support p	RT Support η_p^2
			Subtraction Task			
Subtraction Difficulty	357.9	< .001	.94	531.3	< .001	.96
Text-Entry Difficulty	22.0	< .001	.49	15.0	< .001	.40
Subtraction × Text-Entry	20.0	< .001	.47	14.5	< .001	.39
			Text-Entry Task			
Subtraction Difficulty	133.6	< .001	.85	46.4	< .001	.67
Text-Entry Difficulty	< 1	–	–	2.6	.12	.10
Subtraction × Text-Entry	26.5	< .001	.53	8.6	.007	.27

RT = response times.

Table 3.2 ANOVA results of the accuracy data; separate for Support and No-Support. Note that for the accuracy of the text-entry task we did not find any effects involving Support, which is why we collapsed over Support.

Source	Acc No-Support $F(1,23)$	Acc No-Support p	Acc No-Support η_p^2	Acc Support $F(1,23)$	Acc Support p	Acc Support η_p^2
			Subtraction Task			
Subtraction Difficulty	80.4	< .001	.77	36.8	< .001	.62
Text-Entry Difficulty	45.0	< .001	.66	< 1	-	-
Subtraction × Text-Entry	58.2	< .001	.72	1.1	.3	.05
			Text-Entry Task			
Subtraction Difficulty	25.9	< .001	.53			
Text-Entry Difficulty	173.0	< .001	.88		*Same as No-Support*	
Subtraction × Text-Entry	28.1	< .001	.55			

Acc = accuracy.

participants show an increase in response times in the *hard-hard* condition. Thus, the effect of the problem state bottleneck decreases, but does not fully disappear.

With respect to the response times of the text-entry task, we also observed a significant three-way interaction effect of Support, Subtraction Difficulty, and Text-Entry Difficulty. When external support was provided for the subtraction task, the effects of the problem state bottleneck also decreased in the text-entry task: participants were faster in the *hard-hard* condition. However, also here the two-way interaction effect between Subtraction Difficulty and Text-Entry Difficulty is still present with external support.

The right panels of Figure 3.3 show the accuracy data in the Support condition. Here we see that the two-way interaction effect completely disappears for the subtraction task (the three way interaction is significant, Table 3.3). Thus, participants no longer show the decrease in accuracy in the *hard-hard* condition: the effect of the problem state bottleneck disappeared. In the text-entry task there was no difference between the

Table 3.3 Overall ANOVA results, on the left for response times, on the right for accuracy.

	Response Times			Accuracy		
Source	$F(1,23)$	p	η_p^2	$F(1,23)$	p	η_p^2
			Subtraction Task			
Support	8.02	.009	.26	65.7	< .001	.74
Subtraction	484.2	< .001	.95	103.8	< .001	.82
Text-Entry	26.3	< .001	.53	18.78	< .001	.45
Support × Subtraction	27.6	< .001	.55	66.8	< .001	.74
Support × Text-Entry	4.22	.05	.15	12.7	.002	.36
Subtraction × Text-Entry	29.35	< .001	.56	24.7	< .001	.52
Support × Sub. × Text-Entry	5.05	.03	.18	21.4	< .001	.48
			Text-Entry Task			
Support	2.85	.10	.11	3.20	.09	.12
Subtraction	105.5	< .001	.82	25.7	< .001	.53
Text-Entry	1.17	.29	.05	149.4	< .001	.87
Support × Subtraction	32.7	< .001	.59	< 1	–	–
Support × Text-Entry	3.96	.06	.15	3.87	.06	.14
Subtraction × Text-Entry	19.7	< .001	.46	27.1	< .001	.54
Support × Sub. × Text-Entry	9.29	.006	.29	< 1	–	–

Subtraction = Subtraction Difficulty, Text-Entry = Text-Entry Difficulty.

Support and the No-Support conditions: in both conditions they make most mistakes in the *hard-hard* condition.

In summary, in the response times of the subtraction task the effect of the problem state bottleneck decreased, but did not disappear. In the accuracy data, on the other hand, performance in the subtraction task reached no-bottleneck levels with support. This indicates that external support can indeed help bypassing the problem state bottleneck, but does not bring performance fully back to normal levels. Below we will discuss how our model accounts for this. However, we will first describe the mental workload results.

Pupil Dilation: Mental Workload

Measuring pupil dilation served two goals: (1) investigating whether the problem state bottleneck leads to an increase in mental effort; and (2) seeing if the level of mental effort changes in the support condition. We calculated percentage change in pupil dilation as compared to the average dilation during the fixation screen before each trial; only data of stable fixations were taken into account. For each step in a trial (entering a digit or letter) the maximum pupil dilation was taken, which was then averaged per condition and participant. The results are shown in Figure 3.4, the ANOVA results reported in Table 3.4 (all conditions) and Table 3.5 (collapsed over Support). As we did

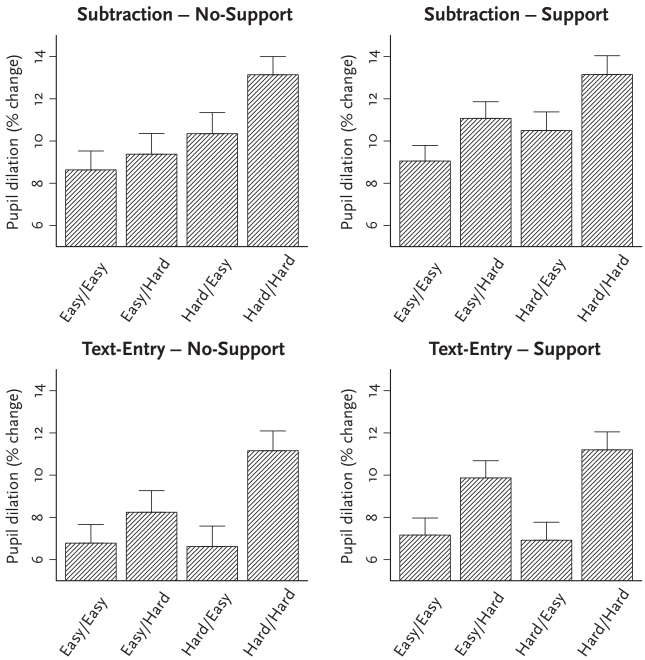

Figure 3.4 Pupil dilation results. Easy/Hard means Easy Subtraction – Hard Text-Entry, etc. Error bars are standard errors.

not find a difference in pupil dilation between the support and the no-support condition for either task (no effects involving Support were significant, Table 3.4), we collapsed over Support for both tasks.

The top row of Figure 3.4 shows pupil dilation in the subtraction task. While it seems as if the interaction effect between Subtraction Difficulty and Text-Entry Difficulty is somewhat smaller with external support, this effect did not reach significance. Overall, the two-way interaction between Subtraction Difficulty and Text-Entry Difficulty was significant (see Table 3.5), as were the main effects of Subtraction Difficulty and Text-Entry Difficulty. Thus, pupil dilation increased with task difficulty, and increased even more when both tasks were hard. This replicates the effects that were found in the response time and accuracy data.

The bottom row of Figure 3.4 shows pupil dilation during the text-entry task. The overall interaction effect between Subtraction Difficulty and Text-Entry Difficulty was even more pronounced for the text-entry task than for subtraction: highest pupil dilation levels were observed in the *hard-hard* condition (Figure 3.4). However, again we did not find a significant difference between the support and the no-support condition. In addition to the two-way interaction effect, the main effect of Text-Entry Difficulty was also significant. Thus, in the text-entry task pupil dilation increased considerably when text-entry was hard, but not when subtraction was hard. However, when both tasks were hard an additional increase in pupil dilation was observed.

Table 3.4 **Overall ANOVA results for the pupil dilation data.**

Source	Pupil Dilation $F(1,23)$	p	η_p^2
	Subtraction Task		
Support	1.86	.19	.07
Subtraction	15.73	< .001	.41
Text-Entry	25.13	< .001	.52
Support × Subtraction	1.24	.28	.05
Support × Text-Entry	< 1	–	–
Subtraction × Text-Entry	7.25	.01	.24
Support × Sub. × Text-Entry	1.47	.24	.06
	Text-Entry Task		
Support	2.98	.10	.11
Subtraction	3.51	.08	.13
Text-Entry	56.14	< .001	.71
Support × Subtraction	1.05	.32	.04
Support × Text-Entry	< 1	–	–
Subtraction × Text-Entry	24.05	< .001	.51
Support × Sub. × Text-Entry	2.16	.15	.09

Subtraction = Subtraction Difficulty, Text-Entry = Text-Entry Difficulty.

Table 3.5 **Overall ANOVA results for the pupil dilation data.**

Source	Pupil Dilation $F(1,23)$	p	η_p^2
	Subtraction Task		
Subtraction Difficulty	15.79	< .001	.41
Text-Entry Difficulty	24.76	< .001	.52
Subtraction × Text-Entry	7.33	.01	.24
	Text-Entry Task		
Subtraction Difficulty	3.56	.07	.13
Text-Entry Difficulty	55.86	< .001	.71
Subtraction × Text-Entry	24.32	< .001	.51

In summary, we found strong interaction effects between Subtraction Difficulty and Text-Entry Difficulty. These indicate that the performance decrease caused by the problem state bottleneck is also reflected in an increase in mental workload. When external support was provided, this increase in mental workload did not disappear for either task.

Cognitive Model

To account for the observed data, we adapted our computational model of the problem state bottleneck (Borst, Taatgen, Stocco, & Van Rijn, 2010; Borst, Taatgen, & Van Rijn, 2010). This model was developed in the ACT-R cognitive architecture (e.g., Anderson, 2007), and uses threaded cognition theory to account for the multitasking aspects of the task (Salvucci & Taatgen, 2008, 2011). Using a cognitive architecture ensures that the components of the model have been validated earlier, which makes it meaningful to take for instance the memory, visual, and motor components of the task into account (e.g., Cooper, 2007; Newell, 1990). We do not describe the complete model here, but refer the interested reader instead to Borst, Taatgen, & Van Rijn (2010).

For the current paper, the problem state component is our main interest. The assumption of the model is that the problem state resource can only maintain one chunk of information at a time. Thus, as long as at most one of the tasks is hard, the model can do the task without a problem – because then at most one problem state is required – but when both tasks are hard the model can only maintain one problem state, which results in interference. The model assumes that in the *hard-hard* condition, on each step in a trial the problem state resource is swapped out. That is, problem state information of the now current task is restored to the problem state resource, while problem state information of the previous task is moved to declarative memory. Thus, when the model switches to the other task, it first retrieves the necessary problem state information from declarative memory, restores this to the problem state resource, and only then performs the task. This takes time (a memory retrieval and 200 ms problem state restoration costs; Anderson, 2005), which results in increased response times in the *hard-hard* condition. Furthermore, incorrect problem states are sometimes retrieved from memory, resulting in lower accuracy scores in the *hard-hard* condition.

The grey bars in the left panels of Figure 3.2 and 3.3 show the fit of this model to the original task[1]. As can be observed, the model accounts well for the interaction effects in both response times and accuracy data, and also matches quite well to the absolute response times and accuracy data of the task (R^2- and RMSD-values are reported in Table 3.6). For instance, while we did not add this explicitly to the model, response times are lower in the hard text-entry condition than in the easy text-entry condition. This is caused by the fact that in the easy condition the model has to read which letter it has to enter before it can search for a button and click on it, while in the hard condition the model (and participants) already knows what word it is entering. This saves visual perception time, and thus results in lower response times in the hard text-entry condition.

We extended the model to also perform the subtraction task in the support condition. There were two basic options: either the model always uses the support indicator on the screen, which would result in equal response times in the *hard subtraction – easy text-entry* and the *hard-hard* condition, or it only uses the indicator when it cannot use its problem state, in the *hard-hard* condition. The latter option seems to be the most rational one: using the problem state to remember whether a carry is in progress

[1] *We fit the model to the data in the no-support condition by estimating retrieval times and retrieval errors from declarative memory, and mouse- and eye movements. The model code is available from http://www.jelmerborst.nl/models/.*

Table 3.6 **Model fit.**

	Measurement	R^2	RMSD
RT	Subtraction No-Support	1.0	181 ms
	Subtraction Support	1.0	171 ms
	Text-Entry No-Support	0.88	88 ms
	Text-Entry Support	0.72	124 ms
Acc	Subtraction No-Support	0.99	.68 %
	Subtraction Support	1.0	.38 %
	Text-Entry No-Support	1.0	.53 %
	Text-Entry Support	1.0	2.8 %

RT = Response Times, Acc = accuracy, RMSD = root mean squared deviation.

takes less time than having to look at the indicator on each step of a trial (cf. the lower response times in the hard text-entry condition in the previous paragraph). Thus in total it will take less time to do the task when the support indicator is only used in the *hard-hard* condition. When we implemented this strategy in the model, that is, using the problem state resource in the *hard-easy* condition, but the support indicator in the *hard-hard* condition, this led to a good model-data fit (see Figure 3.2 and 3.3; please note that we did not make any additional changes to the model, all parameters were kept at the same values as for the no-support condition). On the one hand, implementing this strategy resulted in a small interaction effect in response times in the support condition (in the *hard-hard* condition the support indicator has to be processed, this takes more time than doing the task mentally in the *hard-easy* condition). On the other hand, it also results in a complete absence of the interaction effect in the accuracy data (as the model does not make mistakes anymore because of retrieving incorrect problem states). Thus, it seems that participants use the externally presented support only when it helps them to do the task faster than a mental strategy would allow.

It should be noted that the model always uses its problem state resource to *process carries* in the subtraction task, also in the *hard-hard* condition *with* support. Thus, when it has to process a carry, it will use its problem state to represent the intermediate solution (e.g., when solving ‘6 - 4’ with a carry, it will use the problem state to represent ‘5 - 4’). This is why the model predicts no changes to the interaction effects for the text-entry task when external support is presented: It always has to retrieve the text-entry problem state from declarative memory and restore it to the problem state resource before it can start the text-entry task. However, the data show a small decrease of the interaction effects in the support condition in the text-entry task. A simple explanation could be that participants do not need to overwrite their text-entry problem state when using the support indicator for the subtraction task. This should lead to a complete absence of the interaction effect though, both in response times and accuracy. While we see a decrease, the interaction effect is still present. As we have no strong hypothesis about what happens, we decided against making post-hoc changes to the model to fit this.

Summarizing, the model accounts well for the main effects in the data. While we might have expected to find no interaction effects at all in the support condition for the subtraction task, the model shows that rational behavior does lead to an interaction in the support condition, albeit a smaller one than in the no-support condition. Thus, it seems that it is possible to use problem state information in the environment, but that people only do so when an environment-based strategy is faster than a mental strategy.

General Discussion

In this article we investigated a major bottleneck in human multitasking: the problem state (Borst, Taatgen, & Van Rijn, 2010). It was previously shown that this bottleneck can have considerable influence on performance, both in the lab and in more real-world tasks (Salvucci & Bogunovich, 2010). We therefore looked at how we can design tasks in such a way that the problem state bottleneck can be bypassed. In a dual-task experiment, it was shown that if problem state information is presented on the screen, the negative effects of the problem state bottleneck diminish. Accuracy levels came back to non-bottleneck levels, while response times improved, but did not reach non-bottleneck levels. The computational model showed that this is rational behavior: Participants performed the task as fast as possible, which in the single problem state case meant that they did it mentally, while they used the external support when a problem state was required for both tasks. These results seem to indicate that the problem state bottleneck can be avoided by presenting information in the environment, but that users will only use this information when it leads to faster and less error-prone behavior. Furthermore, the model showed that participants still use their problem state resource for subtraction in the support condition, because otherwise the interaction effects in the text-entry task should have disappeared.

It is not surprising that presenting external support improves performance on a task; the beneficial effects of offloading mental representations to the environment have been described before (e.g., Hollan, Hutchins, & Kirsh, 2000; Kirsh, 1995; Wickens, 1992). However, the current experiment shows that presenting information in the environment only helps in certain cases. Using the model, it is possible to predict exactly when external support is helpful, and when not. In general, we can conclude that it only helps to present external support when users need more than one problem state to carry out a task. While in other cases it might still be used as a memory aid, there are limits on presenting information in the environment. The current research indicates that it is often not necessary to present external information, and that it will not even be used when a mental strategy is faster. Moreover, while the current very simple interface already shows that the costs of visually processing external support have an influence on the task, this is much more important with a real-life interface. When multiple sources of external support are present (as is often the case in real-life systems), this will increase the costs of actually using it, making it important to only present external support when it improves performance.

As we just described, the current research shows that external support is only helpful when multiple problem states are required for performing a task. However, on the other hand it indicates that already with two concurrent problem states it can be profitable to present information externally. This runs counter to classical ideas that we only have to offload internal representations if we cross a threshold of about 5 items. For instance, based on the classical idea of a working memory capacity of 7 ± 2 items (Miller, 1956), Wickens states that "The 7 ± 2 limit is a critically important one in system design." (1992, p. 222). Based on the current research we would argue for a much lower limit of only one item. However, this is also dependent on the costs of processing the external support: naturally, it is only helpful to present support when the gains are higher than the costs.

Besides behavioral measurements, we also recorded pupil dilation during the experiment. Pupil dilation is assumed to reflect mental effort in a task (e.g., Beatty, 1982; Steinhauer & Hakerem, 1992). Where we previously reported interaction effects in response times and accuracy, we now show that the problem state bottleneck also leads to an over-additive increase in mental effort: we observed higher dilation in the *hard-hard* condition than would be expected based on the separate hard conditions. This is not simply a reflection of increased response times: in the *easy subtraction – hard text-entry* condition we see for example faster response times (Figure 3.2), but higher pupil dilation than in the *easy-easy* condition (Figure 3.4). Interestingly, we did not find a significant difference in mental workload between the support and the no-support condition. This could indicate that while participants offload problem state processing to the screen, it still leads to an increase in mental effort (at least as indicated by pupil dilation). However, as there seems to be a slight difference between the conditions (Figure 3.4), additional experiments are necessary to make this clear.

Given that an objective measure of mental workload, pupil dilation, does not show a difference between the support and no-support conditions, it is likely that also a subjective measure of workload (i.e. questionnaires) would not yield a difference. However, we see in the behavioral data that performance does improve significantly when external support is provided. This indicates that asking users if a certain task environment is a useful improvement is not sufficient, but that detailed measures like response times and low-level errors have to be taken into account when designing user interfaces. Using these measures, cognitive models can then help in identifying bottlenecks in behavior, and how these can be bypassed(see also Gray, 2008, on the use of cognitive architectures in human factors). As shown, a model can for instance be used to predict when external support will be useful to the users of the task, and when the visual processing costs are too high for it to be useful.

We conclude that it is possible to bypass the behavioral effects of the problem state bottleneck by presenting external support. However, when giving external support, it should be taken into account that it is only useful when users need more than one problem state to perform a task, and when the processing costs of the support are not higher than the behavioral gains.