Journal of Memory and Language 76 (2014) 174–194

Contents lists available at ScienceDirect

Journal of Memory and Language

journal homepage: www.elsevier.com/locate/jml

Interference within and between declarative and procedural representations in working memory

Miriam Gade*, Michel D. Druey, Alessandra S. Souza, Klaus Oberauer

General Psychology (Cognition), Institute of Psychology, University of Zurich, Switzerland

ARTICLE INFO

Article history:
Received 28 August 2013
revision received 24 June 2014
Available online 1 August 2014

Keywords:
Working memory
Declarative and procedural representations
Attention
Interference

ABSTRACT

We investigate interference between declarative and procedural representations in working memory (WM). Declarative representations are objects of thought, whereas procedural representations provide the (cognitive) actions to work upon these objects. In eight dual-task experiments we varied the number of representations to be maintained in WM (memory load). In Experiments 1–4, we varied declarative and procedural load separately in the two tasks used. In Experiments 5–8, only declarative or procedural load was manipulated in both tasks employed. We measured how much performance in the currently relevant task was impaired by increasing the load in the currently irrelevant task. These cross-task load effects were larger for Experiment 5–8 compared to Experiment 1–4. Yet, in task-switch trials we also obtained cross-task load effects in Experiment 1–4. Our findings support the distinction of declarative and procedural WM as largely independent sub-systems or distinct representational spaces.

Introduction

In our everyday lives we need to have a rapidly fluctuating set of information temporarily available, such as PINs and passwords, or what to write next in a paper, together with action representations for how to perform intended actions (e.g., how to operate ticket vending machines, or how to use a computer keyboard). The cognitive device providing access to the mental representations most needed at any point in time is working memory (WM). WM holds currently relevant representations of the objects of thoughts (i.e., declarative representations), and of the currently intended (cognitive) actions to be performed on these objects (i.e., procedural representations). In this paper, we investigate interference between and within these two sets of representations in WM.

Dissociating declarative and procedural WM

The distinction between declarative and procedural representations is not new in cognitive psychology; it has a long tradition in research on long-term memory and skilled performance (Anderson, 1982; Anderson & Lebiere, 1998; Squire, 2004). For example, the ability to acquire declarative and procedural knowledge has been mapped to different neural substrates (Squire, 2004). However, such a distinction has rarely been made when considering representations maintained and manipulated over the short-term. To fill this gap, the distinction between declarative and procedural representations has been applied to WM in the context of a theoretical framework proposed by Oberauer (2009, 2010). Declarative WM holds representations of what is the case in a real or hypothetical situation, whereas procedural WM holds representations of mental or physical operations acting upon the situation. In this model, WM is conceptualized as an attentional system. This means that WM serves to select and make

---

* Corresponding author. Address: General Psychology (Cognition), Institute of Psychology, University of Zurich, Binzmuhlestr. 14/22, CH- 8050 Zurich, Switzerland.

E-mail address: m.gade@psychologie.uzh.ch (M. Gade).

http://dx.doi.org/10.1016/j.jml.2014.07.002
0749-596X/

available the most relevant declarative and procedural representations for ongoing goal-directed behavior (Oberauer, 2009, see also Cowan, 2001).

As the selection of relevant declarative representations and the selection of relevant procedural representations place analogous demands on working memory, the framework proposed by Oberauer (2009, 2010) included the assumption that analogous mechanisms are used to deal with both types of representations. As a consequence, analogous experimental manipulations should yield comparable effects regardless of the type of representation being processed. This prediction has been confirmed in two studies (Oberauer, Souza, Druey, & Gade, 2013; Souza, Oberauer, Gade, & Druey, 2012). There we have shown that the selection of a memory set, and of an object within this set, is akin to the selection of a task-set, and of a response within this set, respectively: These processes yield similar patterns of repetition benefits and costs, congruency effects, and set size effects.

That research, however, does not indicate whether it is meaningful to make a distinction between declarative and procedural WM. For example, one might question whether these two types of representations are kept separate in WM. Oberauer (2009, 2010) has proposed that declarative and procedural WM are separate and largely independent of each other. However, it is also conceivable that a unitary WM system deals with both types of representations. One way to gain insight into the relation between declarative and procedural WM is by means of a double-dissociation approach. The aim of our study was to test for a double dissociation between declarative and procedural representations in WM and thereby establishing the validity of the distinction between a declarative WM and a procedural WM.

Researchers have used double dissociations between sub-systems or representational domains to gain information about the structuring of WM since the seminal paper of Baddeley and Hitch (1974). They were the first to conceptualize WM as a non-unitary system consisting of multiple buffers (each with separate capacities) that hold different types of representations available for cognitive tasks. The distinction between WM (sub-) systems has built on the investigation of dual-task costs between and within representational domains. For example, the separation between visuospatial and verbal domains has relied on evidence that increasing the amount of visuospatial information to be held in WM (i.e., by increasing visuospatial WM load) interferes less with performance of a concurrent verbal task than with performance of a concurrent visuospatial task, whereas increasing verbal WM load interferes less with a visuospatial than a verbal task. This pattern of interference constitutes a double dissociation (Baddeley, 1986; Cocchini, Logie, Sala, MacPherson, & Baddeley, 2002). Here we followed a similar logic in testing for a double dissociation of declarative and procedural WM.

Finding such a double dissociation would be compatible with two theoretical interpretations: On the one hand, it would fit an interpretation in terms of two largely separate WM sub-systems (i.e., a declarative and a procedural one), an interpretation close to the one originally suggested by Oberauer (2009, 2010). On the other hand, such a result could also be interpreted in terms of separate representational domains within a unitary WM. Yet, a double dissociation is incompatible with a unitary WM that does not distinguish between the representations of objects to be acted upon (declarative content) and the (cognitive) actions itself (procedural content). Furthermore, a double dissociation rules out models of WM that have only one capacity limitation regardless of the content that has to be dealt with. We will return to these interpretational issues in Section 'General discussion'.

Overview of experiments

In our experiments, we always used two tasks that participants had to carry out in an unpredictable order. In the first four experiments, one of these tasks is used to vary declarative load on WM (further on referred to as *declarative load task*) and the other task is used to vary the load on procedural WM (further on referred to as *procedural load task*). In the last four experiments, both tasks loaded WM with the same kind of representation (two declarative load tasks or two procedural load tasks). We used an unpredictable order of tasks to enhance the chances that participants hold both sets of representations in WM concurrently to be optimally prepared for ongoing experimental demands. If this is the case, then representations from the same domain (declarative *or* procedural) should interfere with each other across tasks, such that performance on one task suffers from an increased load from the other task. These *cross-task load effects* are the main diagnostic tool in the present study: A double dissociation between declarative and procedural WM would be obtained if cross-task load effects were considerably reduced or even absent when the load manipulations in the two tasks affect different domains (Experiments 1–4) compared to when they affect the same domain (Experiments 5–8).

To foreshadow our results, we obtained evidence of cross-task load effects in all experiments: In Experiments 1–4, increasing the declarative load in one task reduced performance in a second task relying heavily on procedural representations, and increasing the procedural load of the second task impaired performance in a first task relying strongly on declarative representations. Yet, these cross-task load effects were considerably smaller than those in Experiments 5–8, in which both tasks depended mainly on declarative, or both depended mainly on procedural representations.

Manipulation of memory load

Declarative load was varied through the number of item-position or item-color bindings that participants had to remember for a recognition or recall test. This variable, often referred to as memory-set size, has a reliable effect on retrieval times and accuracies (Oberauer & Kliegl, 2001; Sternberg, 1969). Procedural load was varied through the number of stimulus–response (SR) links participants had to prepare for and apply in a speeded choice task. This manipulation also has a well-established effect on response times and accuracy in choice-RT tasks (Hick, 1952). In either case, structurally comparable sets had to be held in WM: On the one hand, the declarative load comprised *memory sets* consisting of one-to-one mappings between retrieval cues and items; on the other hand, the procedural load comprised *task sets* consisting of one-to-one mappings between stimulus categories and responses.

For both types of tasks we manipulated load on two levels, comparing a low load condition (2 items or 2 SR-links, see Fig. 1, Panel A) to a high load condition (4 items or 4 SR-links, see Fig. 1, Panel B). This design resulted in four possible load combinations: 2 items declarative/2 SR-links procedural; 2 items declarative/4 SR-links procedural; 4 items declarative/2 SR-links procedural; 4 items declarative/4 SR-links procedural. To avoid that cross-task load effects arise because of peripheral (i.e., motor) overlap between the task pairs, we used two different response modalities (i.e., vocal and manual) for each task, which were counterbalanced between the declarative and procedural load tasks across experiments (McLeod, 1977).

Task sequence and task transition

To make sure that participants hold the cue-item bindings as well as the stimulus–response bindings available for immediate use, task sequence was random. Therefore, participants could not anticipate whether the next trial will be a trial of the declarative load task or the procedural load task. We refer to the WM content probed in each trial as currently relevant, while the non-probed WM content is currently irrelevant. For example, imagine a dual-task condition consisting of a declarative load task (i.e., memorize letter-position pairs) and a procedural load task (i.e., press a key in response to a digit). If in the current trial participants have to judge whether a letter-position probe was in the memory set, then the currently relevant task load is declarative. The currently irrelevant load is the number of SR-links in the procedural load task.

The use of a random task sequence encourages participants to keep both memory sets of the two potentially relevant tasks in WM and not outsource one of them to long-term memory. Outsourcing of currently irrelevant content from WM has been observed by Oberauer (2001, 2002, 2005): If one of two initially learnt lists was cued to be the relevant list in the current trial, participants could remove the irrelevant (not cued) list from WM to the activated part of long-term memory within 1–2 s, as indicated by the complete elimination of load effects of the irrelevant list.[1] Critically, at shorter post-cue times (<1 s), the load from both relevant and irrelevant lists affected RTs, showing that they were concurrently held in WM. In the present experiments there was no cue indicating ahead of time which memory set will be relevant on the next trial, therefore leaving no time for outsourcing the currently irrelevant set completely. From this we predicted that in case of a task switch (e.g., declarative load task on trial $n - 1$ and procedural load task on trial n, see Fig. 1), the now irrelevant WM load should still be in WM to some degree. Therefore, testing for cross-task load effects in task-switch trials provides the most incisive approach regarding the question whether declarative and procedural representations compete for the same capacity, or interfere with each other, when held in WM simultaneously.

Consistent vs. varied mappings

Another variable to consider is the variability of mappings in the memory sets and task sets. Research on declarative WM is mostly concerned with the storage and retrieval of newly learnt information (Baddeley, 1986), and therefore routinely relies on new memory sets on every run. In contrast, research on procedural WM primarily focuses on the selection and implementation of well-established SR mappings under conditions of competing action goals or response tendencies (Kiesel et al., 2010; Pashler, 1994). As a consequence, experimenters typically use task sets that remain constant across trials. Constant task sets are characterized by consistent mappings between stimuli and responses, whereas memory sets changing from run to run are characterized by varied mappings between cues (e.g., list positions) and items. In most of our experiments we follow the tradition in each literature, using varied cue-item mappings (i.e., memory sets varying from run to run) to implement declarative load, and constant SR mappings to implement procedural load (see Table 1). However, to investigate whether mapping variability influences the pattern of within-task and cross-task load effects, we used consistent letter-position pairs for the declarative task in Experiment 4.

We do not expect a qualitative difference in the effects of WM load by sets with consistent vs. varied mapping because we regard WM primarily as an attentional system for ongoing selections of goal-relevant representations, and not as a buffer to hold newly learnt arbitrary bindings (Cowan, 2001; Oberauer, 2009): Its capacity limit is a limit on how many cue-item bindings, or stimulus–response bindings, can be held available for ongoing cognition, regardless of whether these bindings are novel (as they are with varied mappings) or well learnt (as they soon become with consistent mappings). First evidence supporting this notion of comparable effects for consistent and varied mappings was found in a study by Risse and Oberauer (2010). In that study, the authors used consistent as well as variable memory sets in an investigation of declarative WM and observed comparable effects for both types of mappings. However, please note that we did not fully cross the mapping of the declarative memory set with the mapping of the procedural memory set across experiments for the following reason: A task instruction varying from trial to trial is likely to be encoded primarily by declarative representations of the instruction, rather than procedural S–R mappings. Therefore we were not confident that we could create a sufficiently pure procedural memory set with varied mappings.

Table 1 provides an overview of all eight experiments, describing the load tasks used, the type of mappings, and the response modalities for each load task.

---

[1] Evidence that people can selectively remove information from WM has been provided by Ecker, Oberauer, and Lewandowsky (2014).

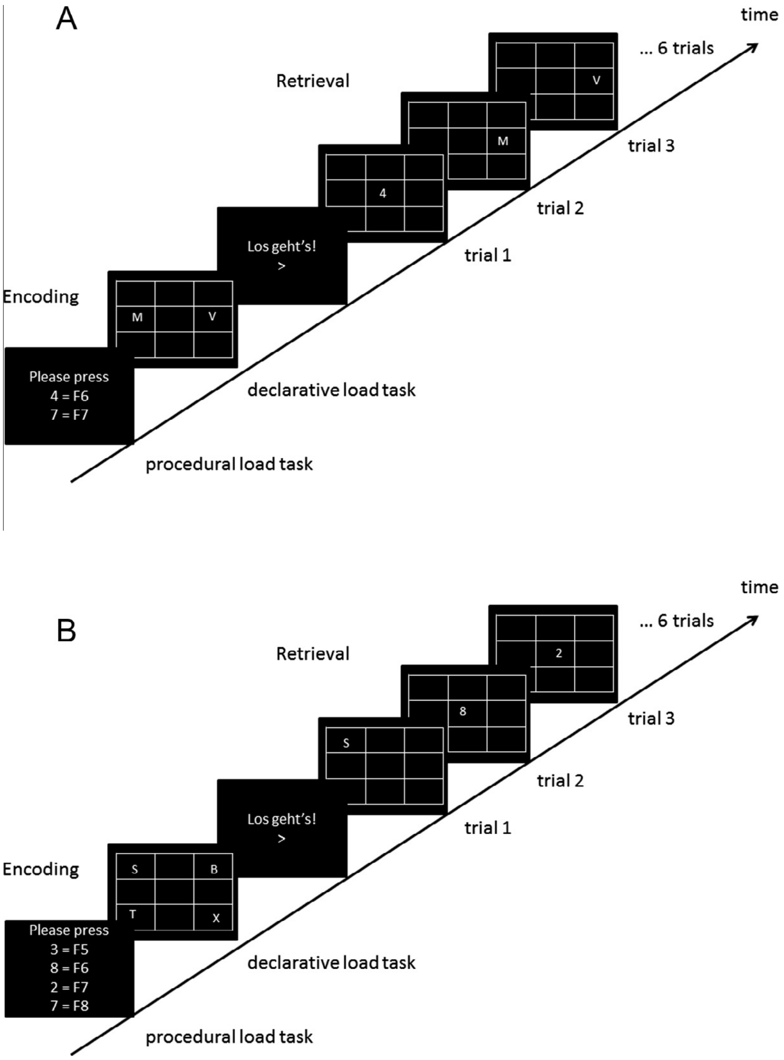

Fig. 1. Events during a run of trials in Experiment 1. Displayed is the declarative load 2/procedural load 2 condition (Panel A), and the declarative load 4/procedural load 4 condition (Panel B). In Panel A, the procedural load task has to be performed for trial 1 and the declarative load task for trial 2. In Panel B, the declarative load task has to be performed in trial 1 and the procedural load task in trial 2.

Predictions

We expected clear effects of the number of item-position bindings, or of S–R bindings, in the currently relevant task, regardless of mapping variability (Risse & Oberauer, 2010). We will refer to this as the within-task load effect. We argue that the tasks we used to vary declarative load – classical short-term memory tasks – rely critically on the integrity of declarative representations because performance on these tasks is limited by the rapid, reliable accessibility of item-position bindings. That said, these tasks also require procedural representations because no task can be performed without a task set. However, we kept the task sets of the declarative-load tasks very simple (e.g., simply saying "yes" or "no" to indicate that a recognition probe matches the memory set or not), so that we assume that performance is not limited as much by the integrity of the task set as by the integrity of the declarative memory set. Likewise, we assume that the tasks we used to manipulate procedural load – classical choice RT tasks – rely primarily on the integrity of procedural representations, without denying that doing these tasks also involves declarative representations. For instance, when making a speeded decision on a stimulus, people inadvertently encode the stimulus to some degree into declarative WM (Oberauer, Lewandowsky, Farrell, Jarrold, & Greaves, 2012).

Based on the above reasoning, we predict that an increase of declarative load in one task impairs performance in that task (i.e., a within-task load effect), and that it impairs performance in the other task to the extent that this other task also critically depends on declarative representations (i.e., a cross-task load effect). Conversely, we predict that an increase of procedural load in one task impairs performance in that task, and impairs performance in the other task to the extent that task also relies critically on the integrity of procedural representations.

Table 1
Overview of the tasks used in each experiment.

Exp.	Task 1				Task 2			
	Load	Task	Mapping variability	Response modality	Load	Task	Mapping variability	Response modality
1	Declarative	Letter-position recognition	Varied	Vocal	Procedural	CRT task on digits	Constant	Manual
2	Declarative	Letter-position recognition	Varied	Manual	Procedural	CRT task on digits	Constant	Vocal
3	Declarative	Symbol-position recognition	Varied	Manual	Procedural	CRT task on digits	Constant	Vocal
4	Declarative	Letter-position recognition	Constant	Vocal	Procedural	CRT task on digits	Constant	Manual
5	Declarative	Letter-position recognition	Varied	Vocal	Declarative	Digit recall	Varied	Manual
6	Declarative	Symbol-position recognition	Varied	Vocal	Declarative	Digit recall	Varied	Manual
7	Declarative	Letter-position recognition	Constant	Vocal	Declarative	Digit recall	Constant	Manual
8	Procedural	CRT task on filled cells	Constant	Vocal	Procedural	CRT task on digits	Constant	Manual

Note. CRT: choice reaction time.

Because no task relies exclusively on declarative WM, or exclusively on procedural WM (as explained above), we do not expect that there is no cross-task load effect at all between declarative and procedural load tasks, but only that it is smaller than the cross-task load effects between two declarative load tasks, or two procedural load tasks, which by their design draw on the same capacity limit or occupy the same representational space. If we find comparable cross-task load effects regardless of task combination, the dissociation between declarative and procedural representations would not hold.

Analytical approach

Data were analyzed using 2 × 2 × 2 × 2 repeated measures analyses of variance (ANOVA) with the factors task (task 1 vs. task 2, see Table 1), task transition (switch vs. repetition), within-task load (low vs. high), and cross-task load (low vs. high) for all experiments. Task was not a factor of theoretical interest. However, given that we used two tasks differing in several regards, including mapping variability and response modality (see Table 1), we assumed that our two tasks might differ in mean reaction time (RT) and sensitivity to manipulations of the relevant factors (i.e., transition, within-task load, and cross-task load). Therefore, we entered task as independent variable to control for this source of variance. Task as additional variable revealed no consistent influence and is theoretically uninformative, and therefore significant main effects and interactions involving task will not be discussed but can be found in Tables 2–5 and the Appendices A–H.

Our primary dependent variable was RT; accuracy data will be reported when differing qualitatively from RT results. Statistical results are reported jointly for those experiments in which the tasks load declarative and procedural WM differentially (Experiments 1–4: see Tables 2 and 3) and for those experiments in which only declarative or procedural WM was loaded by both tasks (Experiments 5–8: see Tables 4 and 5).

Data trimming was carried out as follows for all experiments: The complete first session was discarded as practice, as was the first run in each condition in the remaining sessions. The first trial in each run was omitted as it could not be classified as a switch or repetition trial. Furthermore, we discarded all trials in which not only the task but also the imperative stimulus (e.g., the recognition probe or the target stimulus for the choice task) was the same as in the preceding trial. This was done to avoid confounding within-task load effects with repetition priming effects, because stimulus repetitions were more frequent in low-load conditions. This procedure discarded 6.2% of all trials for the declarative load 2/procedural load 2 conditions, 3.1% of trials for the declarative load 2/procedural load 4 conditions, 1.6% of trials from the declarative load 4/procedural load 2 conditions, and 0.8% of trials from the declarative load 4/procedural load 4 conditions. The same numbers also apply to the respective load conditions in Experiments 5–8 in which both tasks loaded only declarative (Experiments 5–7) or only procedural (Experiment 8) WM.

All errors and all trials following an error were also excluded from the RT analyses, as were all RTs below 200 ms. For computing the mean RTs, we additionally removed from the raw RT data those RTs which were more than three standard deviations above the respective participant's mean in each cell of the design. This outlier-removal procedure resulted in discarding an additional 1–2% of the already trimmed data for all experiments. For the statistical analyses the RT data were *log*-transformed to better approximate a normal distribution, but effects are reported in the original scale. For accuracy data we performed a *probit* transformation to account for the underlying binomial distribution (Bartlett, 1947) when performing the same statistical tests as for RT data. A *probit* transformation on percentage correct compensates for ceiling effects and yields a distribution closer to normality. In each design cell, all means were based on more than 34 observations per participant, averaged across experiments.

Experiment 1

Method

Participants. Twelve participants (mean age 24.4 years, 11 women) took part in Experiment 1. For all experiments reported in the present paper, participants were compensated for their participation with either course credit or 60 Swiss Francs (CHF).

Stimuli and tasks. Participants performed two tasks: one declarative load task that loaded WM with declarative rep-

Table 2
Test statistics for RT analyses of variance for Experiments 1–4.

Effect	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
	F	p	η_p^2	F	p	η_p^2	F	p	η_p^2	F	p	η_p^2
Task (T)	**27.668**	**<.001**	**.716**	.182	.678	.016	.714	.416	.061	**355.422**	**<.001**	**.970**
Transition (TR)	**208.557**	**<.001**	**.950**	**148.292**	**<.001**	**.931**	**114.465**	**<.001**	**.912**	**110.994**	**<.001**	**.910**
Within-task load (WL)	**209.067**	**<.001**	**.950**	**95.839**	**<.001**	**.897**	**112.259**	**<.001**	**.911**	**78.666**	**<.001**	**.877**
Cross-task load (CL)	2.654	.132	.194	**15.493**	**.002**	**.585**	.001	.974	.000	**57.995**	**<.001**	**.841**[a]
T × TR	**18.468**	**.001**	**.627**	.088	.773	.008	.081	.781	.007	**65.601**	**<.001**	**.856**
T × WL	**8.296**	**.015**	**.430**	4.677	.053	.298	.265	.617	.024	**6.492**	**.027**	**.371**
TR × WL	**54.204**	**<.001**	**.831**	**17.395**	**.002**	**.613**	**58.181**	**<.001**	**.841**	1.779	.209	.139
T × TR × WL	.943	.352	.079	.115	.741	.010	1.196	.297	.098	1.487	.248	.119
T × CL	2.499	.142	.185	.214	.652	.019	.818	.385	.069	**56.070**	**<.001**	**.836**
TR × CL	**44.281**	**<.001**	**.801**	1.542	.240	.123	**4.933**	**.048**	**.310**	**17.417**	**.002**	**.613**
T × TR × CL	.010	.922	.001	.634	.443	.055	.071	.795	.006	**14.439**	**.003**	**.568**
WL × CL	2.055	.179	.157	.486	.500	.042	**24.280**	**<.001**	**.688**	**86.084**	**<.001**	**.887**
T × WL × CL	**5.286**	**.042**	**.325**	1.043	.329	.087	1.111	.315	.092	**27.242**	**<.001**	**.712**
TR × WL × CL	2.760	.125	.201	.561	.470	.049	.003	.957	.000	1.815	.205	.142
T × TR × WL × CL	2.702	.128	.197	3.344	.095	.233	.421	.530	.037	**6.681**	**.025**	**.378**

Note. Significant effects in bold. Degrees of freedom for all comparisons were $F(1,11)$.

[a] Please note that the effect went in the direction opposite to the predictions (see also Fig. 2, lower left panel, Experiment 4).

Table 3
Statistical values for accuracy data for Experiment 1–4.

Effect	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
	F	p	η_p^2	F	P	η_p^2	F	p	η_p^2	F	p	η_p^2
Task (T)	0.018	0.896	0.002	**92.988**	**<0.001**	**0.894**	**32.845**	**<0.001**	**0.749**	**48.765**	**<0.001**	**0.816**
Transition (TR)	4.756	0.052	0.302	**36.59**	**<0.001**	**0.769**	2.366	0.152	0.177	**7.813**	**0.017**	**0.415**
Within-task load (WL)	2.22	0.164	0.168	**12.41**	**0.005**	**0.53**	**14.827**	**0.003**	**0.574**	**76.024**	**<0.001**	**0.874**
Cross-task load (CL)	0.304	0.592	0.027	3.405	0.092	0.236	**6.604**	**0.026**	**0.375**	**15.119**	**0.003**	**0.579**
T × TR	0.393	0.543	0.035	3.034	0.109	0.216	**4.965**	**0.048**	**0.311**	1.268	0.284	0.103
T × WL	0.995	0.34	0.083	0.284	0.605	0.025	0.111	0.745	0.01	**56.241**	**<0.001**	**0.836**
TR × WL	3.927	0.073	0.263	**10.564**	**0.008**	**0.49**	**7.651**	**0.018**	**0.41**	**54.937**	**<0.001**	**0.833**
T × TR × WL	1.789	0.208	0.14	4.136	0.067	0.273	3.262	0.098	0.229	**8.674**	**0.013**	**0.441**
T × CL	0.96	0.348	0.08	2.489	0.143	0.185	3.766	0.078	0.255	0.783	0.395	0.066
TR × CL	0.006	0.938	0.001	0.004	0.95	0	0.735	0.41	0.063	2.23	0.163	0.169
T × TR × CL	**6.648**	**0.026**	**0.377**	0.249	0.628	0.022	0.474	0.505	0.041	1.016	0.335	0.085
WL × CL	0.15	0.706	0.013	0.032	0.862	0.003	1.971	0.188	0.152	**5.014**	**0.047**	**0.313**
T × WL × CL	0.887	0.367	0.075	0.399	0.541	0.035	0.132	0.723	0.012	4.206	0.065	0.277
TR × WL × CL	0.992	0.341	0.083	0.059	0.812	0.005	5.866	0.034	0.348	0.063	0.806	0.006
T × TR × WL × CL	0.097	0.761	0.009	**7.826**	**0.017**	**0.416**	0.208	0.657	0.019	0.448	0.517	0.039

Note. Significant effects in bold. Degrees of freedom for all comparisons were $F(1,11)$.

resentations, and one procedural load task loading WM with procedural representations. Participants performed both load tasks in randomized order in short runs of six trials (see Fig. 1). The declarative load task consisted of either two or four letters being presented in a 3×3 grid on the screen at the beginning of each run. Letters were drawn at random from the set of all consonants except "J" and "Y". Letters for the declarative task were chosen anew for each run of trials with the only constraint that letters never repeated within one memory set. In trials with load 2, the letters were presented in the left and the right cell of the middle row of the grid (Fig. 1, Panel A). In trials with load 4, the letters were presented in the same two cells plus the top and the bottom cell of the middle column (Fig. 1, Panel B). Recognition was probed by presenting a letter from the memory set in either the same position as in the memory set or in one of the other positions that were occupied by different letters in the memory set. Participants decided whether the letter-position conjunction of the probe stimulus matched the memory set. They gave a verbal yes/no response which was registered by a voice key and coded as correct or incorrect by the experimenter afterwards.

In the procedural task, participants were instructed to press a key in response to a digit presented centrally in the 3×3 grid. As response keys we used the F5 and F7 keys for the two-digit condition (Fig. 1, Panel A), and the F5, F6, F7, and F8 keys in the four-digit condition (Fig. 1, Panel B). To ease responding we turned the keyboard such that the function keys came to lie closer to the participants. There were four sets of digits for the two-digit conditions as well as for the four-digit conditions; the assignment of digits to conditions was counterbalanced across participants but remained constant within a participant. The six digits a participant encountered never overlapped. The assignments of digits to response keys were made such that all possible compatible mappings (e.g., small digits requiring

Table 4
Statistical values for RT data for Experiment 5–8.

Effect	Experiment 5			Experiment 6			Experiment 7			Experiment 8		
	F	*P*	η_p^2	*F*	*p*	η_p^2	*F*	*p*	η_p^2	*F*	*p*	η_p^2
Task (T)	3.022	.110	.216	.408	.536	.036	**21.277**	**.001**	**.659**	**19.273**	**.001**	**.562**
Transition (TR)	**24.616**	**<.001**	**.691**	**23.093**	**.001**	**.677**	**118.872**	**.000**	**.915**	**84.101**	**<.001**	**.849**
Within-task load (WL)	**338.776**	**<.001**	**.969**	**151.915**	**<.001**	**.932**	**222.940**	**.000**	**.953**	**230.851**	**<.001**	**.939**
Cross-task load (CL)	**20.366**	**.001**	**.649**	**5.495**	**.039**	**.333**	**12.273**	**.005**	**.527**	**18.877**	**.001**	**.557**
T × TR	**25.560**	**<.001**	**.699**	**5.382**	**.041**	**.329**	**35.431**	**.000**	**.763**	1.952	.183	.115
T × WL	**61.233**	**<.001**	**.848**	**28.403**	**<.001**	**.721**	**36.426**	**.000**	**.768**	1.827	.197	.109
TR × WL	**9.666**	**.010**	**.468**	**12.059**	**.005**	**.523**	**8.431**	**.014**	**.434**	**30.930**	**<.001**	**.673**
T × TR × WL	**17.956**	**.001**	**.620**	.014	.908	.001	**25.217**	**.000**	**.696**	.863	.368	.054
T × CL	**5.182**	**.044**	**.320**	.427	.527	.037	3.372	.093	.235	3.962	.065	.209
TR × CL	**21.808**	**.001**	**.665**	**39.185**	**<.001**	**.781**	.742	.407	.063	**18.204**	**.001**	**.548**
T × TR × CL	3.949	.072	.264	**27.912**	**<.001**	**.717**	.051	.825	.005	.143	.710	.009
WL × CL	.461	.511	.040	.296	.597	.026	.081	.781	.007	.980	.338	.061
T × WL × CL	.242	.632	.022	2.571	.137	.189	.566	.468	.049	2.554	.131	.145
TR × WL × CL	.818	.385	.069	**47.667**	**<.001**	**.813**	.009	.927	.001	1.134	.304	.070
T × TR × WL × CL	.404	.538	.035	**14.957**	**.003**	**.576**	.743	.407	.063	.782	.390	.050

Note. Significant effects in bold. Degrees of freedom for all comparisons were *F*(1, 11) in Experiment 5–7, for Experiment 8 degrees of freedom were *F*(1, 15).

Table 5
Statistical values for accuracy data for Experiment 5–8.

Effect	Experiment 5			Experiment 6			Experiment 7			Experiment 8		
	F	*P*	η_p^2	*F*	*p*	η_p^2	*F*	*p*	η_p^2	*F*	*p*	η_p^2
Task (T)	2.02	0.183	0.155	**8.736**	**0.013**	**0.443**	**131.469**	**<0.001**	**0.923**	0.778	0.392	0.049
Transition (TR)	**60.48**	**<0.001**	**0.846**	**66.765**	**<0.001**	**0.859**	**145.945**	**<0.001**	**0.93**	**22.568**	**<0.001**	**0.601**
Within-task load (WL)	**141.333**	**<0.001**	**0.928**	**68.129**	**<0.001**	**0.861**	**90.867**	**<0.001**	**0.892**	**15.894**	**0.001**	**0.514**
Cross-task load (CL)	**13.689**	**0.004**	**0.554**	**15.814**	**0.002**	**0.59**	0.002	0.963	0	2.069	0.171	0.121
T × TR	0.21	0.656	0.019	**6.114**	**0.031**	**0.357**	**5.735**	**0.036**	**0.343**	0.136	0.718	0.009
T × WL	**35.712**	**<0.001**	**0.765**	**14.849**	**0.003**	**0.574**	**18.39**	**0.001**	**0.626**	0.008	0.931	0.001
TR × WL	**10.69**	**0.007**	**0.493**	1.443	0.255	0.116	**16.864**	**0.002**	**0.605**	**7.981**	**0.013**	**0.347**
T × TR × WL	4.546	0.056	0.292	**9.506**	**0.01**	**0.464**	**23.786**	**<0.001**	**0.684**	1.311	0.27	0.08
T × CL	**15.406**	**0.002**	**0.583**	3.807	0.077	0.257	0.885	0.367	0.074	0.467	0.505	0.03
TR × CL	1.632	0.228	0.129	**11.139**	**0.007**	**0.503**	0.017	0.899	0.002	3.971	0.065	0.209
T × TR × CL	0.103	0.755	0.009	4.209	0.065	0.277	3.002	0.111	0.214	0.022	0.884	0.001
WL × CL	0.22	0.648	0.02	0.888	0.366	0.075	0.197	0.666	0.018	2.605	0.127	0.148
T × WL × CL	0.033	0.86	0.003	0.011	0.92	0.001	5.002	0.047	0.313	0.046	0.833	0.003
TR × WL × CL	0	0.992	0	5.612	0.037	0.338	0.609	0.452	0.052	1.629	0.221	0.098
T × TR × WL × CL	0.231	0.64	0.021	0.376	0.552	0.033	**10.13**	**0.009**	**0.479**	0.039	0.847	0.003

Note. Significant effects in bold. Degrees of freedom for all comparisons were *F*(1, 11) in Experiment 5–7, for Experiment 8 degrees of freedom were *F*(1, 15).

a left key-press response) were excluded to make sure that participants had to keep the instructed mapping in WM within a run of trials, rather than using SR associations in long-term memory (Risse & Oberauer, 2010).

Procedure. The experiments were programmed in C using the Tscope library functions (Stevens, Lammertyn, Verbruggen, & Vandierendonck, 2006). Participants were tested for four sessions, each of which lasted about 50 min. The sessions were separated by at least 12 h and, at most, 7 days.

In each of the four sessions, participants worked through the four conditions generated by crossing the load manipulations in each task (2 letters/2 digits vs. 2 letters/4 digits vs. 4 letters/2 digits vs. 4 letters/4 digits) in a fixed sequence counterbalanced across participants. Conditions were thus blocked and participants were informed about the actual load conditions by the experimenter at the beginning of each condition. Participants were invited to take a short break between blocks, during which the experimenter prepared the next block. In each session there were 20 runs of six trials per condition, comprising a total of 480 trials per session. The first run of each condition was considered as warm-up and discarded. At the beginning of each run, the items of the memory set for the declarative task were presented simultaneously with duration of 1 s per item (i.e., 2 s for two letters, 4 s for four letters). Next, a reminder of the stimulus–response mapping for the procedural WM task (i.e., the assignment of digits to keys) was displayed (again, 1 s for every digit-key assignment to be encoded). Then a start announcement in German was displayed ("Los geht's!") and participants could start the run by clicking on the arrow below the message (see Fig. 1).

Within a run of trials, the two tasks occurred in a random order. The response-stimulus interval (RSI) was fixed to 500 ms. Within each condition (but not necessarily within each run), each task occurred equally often. Participants received no feedback on their performance. Maximal RT was limited to 10 s; if no response was recorded until then, the trial counted as an error and the next trial started.

Results

RT data. The results of the ANOVA on RTs are shown in Table 2; the mean RTs for each condition are given in Appendix A. Fig. 2 depicts proportional load effects (i.e., mean load effect/mean RT) for the within-task and cross-task load effects split up by task transition and task for Experiments 1–4, the data of Experiment 1 are in the upper left panel. Participants were 194 ms slower in the declarative load task compared to the procedural load task. On task-switch trials participants were 136 ms slower than on task-repetition trials. They were also 282 ms slower with high within-task load compared to low within-task load. Cross-task load failed to yield a significant main effect. Both within-task and cross-task load effects were moderated by task transitions. While the effect of within-task load was 324 ms for task-repetition trials, it was 238 ms for task-switch trials. For cross-task load, we observed the reversed pattern: Whereas the effect of cross-task load was −14 ms for task-repetition trials (not significantly different from zero, $t(11) = 1.52$, $p = .157$), it increased to 46 ms for task-switch trials (significantly different from zero, $t(11) = 3.80$, $p = .003$).

Accuracy data. The ANOVA for the accuracy data is shown in Table 3, and the mean accuracy rates for each condition can be found in Appendix A. Mean accuracy was 97.8%. None of the effects of theoretical interest (task transition, within-task load, cross-task load, and their interactions) was significant (see Appendix A).

To balance the assignment of vocal and manual responses to declarative and procedural tasks, we ran Experiment 2 with one change relative to Experiment 1: We asked participants in Experiment 2 to respond vocally in the procedural load task and manually in the declarative load task. Because of the similarity of the experiments, discussion of the results of Experiment 1 is deferred to after Experiment 2.

Experiment 2

Method

Twelve new participants took part in Experiment 2 (mean age 29.0 years; 8 women). The stimuli, tasks, and procedure were the same as in Experiment 1 with the only exception that participants now responded to the declarative load task with the F6 and F7 key of a standard Swiss German Keyboard (again, turned around to facilitate responding) to indicate “yes” and “no” responses, respectively. Conversely, they spoke the syllables “bu”, “da”, “ki” and “to” in response to the digits in the procedural load task. Again, the stimulus–response mapping in the procedural task (i.e., the digit-to-syllable assignment) was kept constant throughout the experiment for any given participant, but was varied between participants. The memory set of the declarative load task varied across runs for all participants.

Results

RT data. The ANOVA for the RTs is listed in Appendix B, and the mean RTs for each condition are presented in Appendix B. Fig. 2, upper right panel shows the proportional load effects (within-task and cross-task effects) for each task and task transition. As in Experiment 1, participants were 166 ms slower in task-switch trials than in

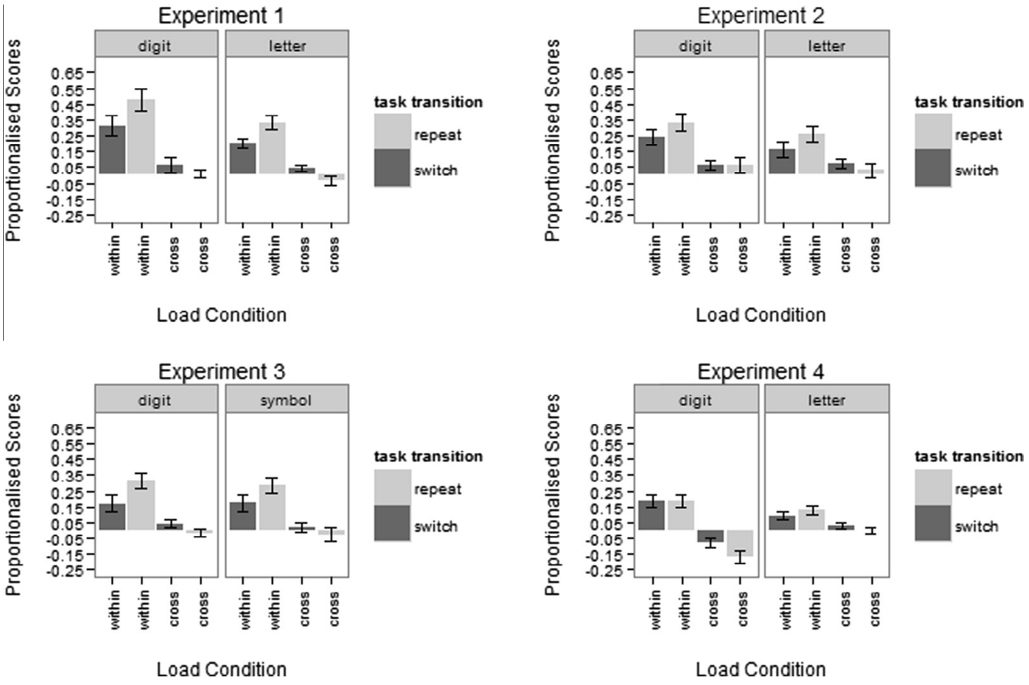

Fig. 2. Proportional within-task (within) and cross-task (cross) load effects by task transition (repeat vs. switch trials) and task (digit task = procedural load task; letter or symbol task = declarative load task) for Experiments 1–4. Error bars denote 95% within-subjects confidence intervals.

task-repetition trials. Moreover, participants were 249 ms faster when within-task load was low compared to when within-task load was high. Contrary to Experiment 1, cross-task load effects were significant: Participants were 60 ms slower when cross-task load was high compared to when cross-task load was low. Within-task load effects were different for switch and repetition trials: The effect of within-task load was 66 ms higher in task-repetition trials than in task-switch trials. Cross-task load effects did not interact with task transition (see Fig. 2, upper right panel).

Accuracy data. The ANOVA for the accuracy is shown in Table 3, and the mean accuracy rates for each condition are presented in Appendix B. Overall, mean accuracy was high (95.7%). Participants were 2.5% less accurate in case of a task switch compared to a task repetition. Within-task load effects mirrored RT data and decreased performance by 6% in case of high within-task load. Furthermore, within-task load interacted with task transition: Participants had higher switch costs (3.6%) with low within-task load compared to high within-task load (1.4%, see Appendix B). The cross-task load effect was not significant.

Discussion of Experiments 1 and 2

Combining one declarative and one procedural load task revealed transition costs between these tasks in RTs, and sometimes also accuracy. As shown in Fig. 2, upper panels, performance in each task was also impaired by increasing the load within the task: Responding in the declarative load task was slower when participants had to hold 4 items than when they had to hold 2 items. The same pattern of within-task load effect was observed for the procedural load task. Of most diagnostic value for our question, however, are the cross-task load effects. Experiment 1 confirmed our hypothesis that declarative and procedural representations do not interfere much in WM, as indicated by the lack of a main effect of cross-task load. When participant had to switch between tasks, however, cross-task load effects emerged, showing a condition in which declarative and procedural WM are more likely to interfere with each other.

Contrary to Experiment 1, cross-task load effects were observed independently of task transition in Experiment 2, showing a higher degree of interference between declarative and procedural WM than in Experiment 1. The only difference between Experiments 1 and 2 was that we reversed the response modalities between the two tasks, such that in Experiment 2 the procedural load task required vocal responses, and the declarative load task required manual responses. One explanation of how this could have led to a larger cross-task load effect is that the vocal responses in the procedural load task produced phonological interference. It is known that short-term memory for verbal materials is disrupted by concurrent speaking, and the disruption is larger when the speech output is more variable. This effect is predicted from interference theories of WM on the assumption that speech output is inadvertently encoded into declarative WM (Lewandowsky, Geiger, Morrell, & Oberauer, 2010; Lewandowsky, Geiger, & Oberauer, 2008). The procedural load task with high load involved four digits linked to four different syllables as responses, whereas the low-load condition involved only two syllables. Therefore, the sequence of syllables to be spoken under high procedural load was more variable than that under low procedural load. This could have produced more disruption of memory for the letters in the condition with high procedural load. Conversely, rehearsal of the letters in the declarative load task could have created motor competition with the vocal responses in the procedural task. If participants felt compelled to rehearse 4-letter lists more often than 2-letter lists, this could explain the small cross-task load effect from declarative load on the procedural load task in Experiment 2.

If either of these explanations is correct, then the effect of cross-task load should be abolished again when using non-verbal stimuli for the declarative load task. Therefore, we re-ran Experiment 2, this time using non-verbalizable stimuli as material for the declarative WM task. If the cross-task load effect reflected some process specific to the combination of verbal declarative memory sets with vocal responses, then we should expect no such cross-task load effects in Experiment 3. In contrast, if the cross-task load effect in Experiment 2 reflected the general case, and Experiment 1 was an exception, then we should again obtain a cross-task load effect in Experiment 3.

Experiment 3

Method

Twelve new participants (mean age 24.5 years, 10 women) took part in Experiment 3. We used the stimulus material from Ricker, Cowan, and Morey (2010), for which the authors showed that maintenance is not impaired by articulatory suppression, suggesting that these memoranda were not encoded into WM phonologically. However, as many of them were letters in either Greek or Russian script and therefore we could not be sure that they would not be named by participants, we ran a pilot experiment on the whole stimulus set of Ricker et al. (2010) in which we asked participants ($n = 6$) to try to name all the stimuli. For Experiment 3 we used only those stimuli that were named by only 50% or less of our pilot sample. By this criterion we selected 18 of the original 50 symbols (see Fig. 3). In all other respects, the tasks, procedure and design were the same as in Experiment 2, only the RSI was increased to 750 ms because verbal responses were coded online by the experimenter in this experiment.

Results

RT data. The ANOVA for the RT data is listed in Table 2, and the mean RTs for each cell of the design can be found in Appendix C. Proportional load effects can be found in the lower left panel of Fig. 2. Participants were 241 ms slower when they had to switch the task compared to a task repetition. As in Experiments 1 and 2, participants were slower (by 262 ms) when within-task load was high compared to when within-task load was low. As predicted,

Fig. 3. Nonverbalizable stimuli used in Experiments 3 and 6.

there was no main effect of cross-task load. Within-task load was moderated by task transition in that the within-task load effects were 306 ms when the task repeated, and 217 ms when the task switched. Cross-task load also interacted with task transition, reflecting a larger cross-task load effect for task-switch trials (37 ms, significantly different from zero, $t(11) = 2.52$, $p = .029$), than for task-repetition trials (−23 ms, not different from zero, $t(11) = -1.2$, $p = .25$). Cross-task load interacted with within-task load: The effect of cross-task load was 38 ms for low within-task load, and −25 ms for high within-task load.

Accuracy data. The ANOVA for accuracy is shown in Table 3, and the mean accuracy rates for each condition are listed in Appendix C. Mean accuracy was 95.9%. Participants were more accurate in the procedural load task (98.7%) than in the declarative load task (93.1%). Task transition failed to yield a main effect. Contrary to the RT data, there was a main effect of cross-task load. Participants were less accurate with high cross-task load (95.2%) than with low cross-task load (96.7%). Task transition interacted with within-task load such that the within-task load effect was 1.8% for task repetitions and decreased to 1.5% for task switch trials.

Discussion

As predicted, using non-verbalizable stimuli for the declarative task abolished the main cross-task load effect in the RTs. Contrary to our predictions we still obtained a cross-task load effect in the accuracy data. Thus, it appears that the cross-task load effect in Experiment 2 was not entirely abolished in Experiment 3.

In all other regards the results of Experiment 3 were comparable to those of Experiments 1 and 2. Again, we obtained large within-task load effects, which were larger in task-repetition trials than in task-switch trials. The cross-task load effect on RTs was also moderated by task transition, but in the opposite direction, such that the cross-task load effect was positive in task-switch trials, and slightly negative in task-repetition trials, replicating Experiment 1.

In all three experiments so far we used varied memory sets for our declarative load tasks but consistent SR mappings for the procedural load tasks. To make the two tasks more comparable in this regard, we used constant memory sets for the declarative load task in Experiment 4.

Experiment 4

Method

Twelve new participants took part in Experiment 4 (mean age 24.8 years, 7 women). The stimuli, tasks, procedure and design were the same as in Experiment 1 with the only exception that participants now worked through the entire experiment on constant letter sets in the declarative load task. Letters for the two-letter condition were always different from the four-letter condition. Recognition decisions were given vocally and coded online by the experimenter as in Experiment 3, using an RSI of 750 ms. Responses in the procedural load task were again entered using the F5 to F8 keys of a standard Swiss German keyboard.

Results

RT data. The ANOVA for the RTs can be found in Table 2, and the mean RTs for each condition are given in Appendix D. Proportional load effects are shown in the bottom right panel of Fig. 2. Participants were 89 ms slower when they had to switch the task compared to when the task repeated. Furthermore, they were 114 ms slower with a high within-task load compared to a low within-task load. The significant effect of cross-task load was in the opposite direction of what would be expected: Participants were 34 ms faster with high cross-task load. Cross-task load interacted with task transition, again reflecting larger (in this case, less negative) cross-task load effects in case of a task switch [−16 ms, $t(11) = -2.25$, $p = .046$] compared to a task repetition [−52 ms, $t(11) = -6.31$, $p < .001$]. Furthermore, cross-task load interacted with within-task load. The cross-task load effect was 25 ms for the low within-task load condition and −93 ms for the high within-task load condition (see Appendix D and Fig. 2, lower right panel).

Accuracy data. The ANOVA for accuracy data is listed in Table 3, and the mean accuracies for each condition are presented in Appendix D. Mean overall accuracy was 95.3%. Contrary to the RT data, the main effect of within-task load went in the opposite direction. Participants were more accurate (97.4%) when within-task load was high compared to when it was low (93.1%). Furthermore, we obtained a significant effect of cross-task load. Participants were 0.9% less accurate when cross-task load was high.

Task transition moderated within-task load such that within-task load effect reversed from repetition to switch trials (+0.2% for repetition and −8.3% for switch trials).

Discussion

In Experiment 4 we used constant letter sets for the declarative load task next to consistent digit-response pairs for the procedural load task. We replicated the effects of within-task load and task transition, although with smaller effects than in Experiments 1–3. The cross-task load effect was reversed in RT data, actually showing a benefit with high cross-task load when compared to low cross-task load. However, there was a cost of higher cross-task load in the accuracy data, suggesting a speed-accuracy trade-off.

Summary of Experiments 1–4

Across the first four experiments, in which we independently loaded declarative and procedural WM across two tasks, we found no effect (Experiments 1 and 3), fairly small effects (Experiment 2), or even reversed effects (Experiment 4) of cross-task load on RTs. In any case the effects of cross-task load were much smaller than the effects of within-task load (see Figs. 2 and 5 for an overview). Furthermore, for two of the four experiments cross-task load effects were more pronounced in task-switch trials than in task-repetition trials (see Fig. 5 for all experiments).

We tentatively conclude that declarative and procedural representations interfere very little in WM, which would be consistent with the assumption that these representations draw largely on separate capacity limits. The next step is now to investigate whether there is a larger cross-task load effect if both tasks in our dual-task paradigm load only declarative or only procedural WM. To this end in Experiments 5–8 we used either two declarative load tasks or two procedural load tasks. Experiments 5–7 combined two declarative load tasks, whereas Experiment 8 combined two procedural load tasks. Materials and responses in Experiments 5–8 were chosen such that each experiment of the following series matches at least one experiment of the previous series (Experiments 1–4). In this way, the comparison of Experiments 1–4 with Experiments 5–8 is not confounded by differences in the materials involved. We expected to see considerably larger cross-task load effects than in the first four experiments, thus showing that performance of two different tasks suffers when these tasks load WM in the same representational domain.

Experiment 5

Experiment 5 combined the letter-position recognition task (declarative load task) of Experiments 1, 2 and 4 with a second declarative load task, using the same type of stimuli (i.e., digits) as the procedural load task of those experiments.

Method

Twelve new participants (mean age 24.1 years 12 women) took part under the same conditions as in the previous experiments. The first declarative load task was identical to the letter-recognition task of Experiments 1, 2, and 4. Our second declarative load task was a digit-recall task. At the beginning of each run, participants first encoded two or four letters (as in Experiments 1 and 2), then two or four digits. Each digit was presented in a different color, and all digits were presented simultaneously in a row in the center of the screen. Encoding of the two memory sets was followed by a run of six trials in which either the letter-recognition task or the digit-recall task had to be performed in random order. Letter recognition was probed as in Experiment 1. Digit recall was probed by filling the central cell of the grid with one of the digits' color, and participants had to type the digit that had been presented in that color. Participants were presented with new letters and digits on every run (i.e., both memory sets used variable mappings). They were tested for four sessions, each of which included the same four conditions as the first four experiments. The sequence of conditions within a session was counterbalanced as before across participants but kept constant for each participant.

Results

RT data. The ANOVA of the RT data is shown in Table 4, mean RTs for each condition are shown in Appendix E. Fig. 4 depicts proportional within-task and cross-task load effects split by task transition and task for Experiments 5–8. We found a main effect of task transition, observing significant task-switch costs of 184 ms. Participants showed a within-task load effect of 429 ms. As predicted, we also obtained a significant effect of cross-task load. Participants were 113 ms slower when the currently irrelevant task had a high load (4 items) compared to a low load (2 items). Task transition interacted with both within-task load as well as cross-task load (see Appendix E). In line with the effects observed in Experiments 1–4, the cross-task load effect increased from 64 ms in task-repetition trials to 161 ms in task-switch trials.

Accuracy data. The ANOVA of the accuracy data can be found in Table 5, and mean accuracy rates for each condition are listed in Appendix E. Mean overall accuracy was 87.6%. We observed main effects of task transition (93.2% for repeat compared to 83.8% for switch trials) as well as for within-task load and cross-task load. Participants were more accurate when within-task load was low (95.8%) than when it was high (81.1%). Furthermore, we found a main effect of cross-task load. Again, participants were more accurate for low (90.1%) than for high cross-task load (86.6%). Task transition interacted with within-task load such that participants' accuracy decreased by 7.7% in case of task repetition and by 21.7% in case of a task switch. For cross-task load, the accuracy pattern went in the same direction as in RT data.

Discussion

The results of the current experiment showed the expected cross-task load effect when both tasks loaded WM with declarative representations. This pattern of results adds important support to the conclusions drawn from Experiments 1–4. Namely, declarative and procedural representations in WM do not interfere as much with each other than do two sets of declarative representations.

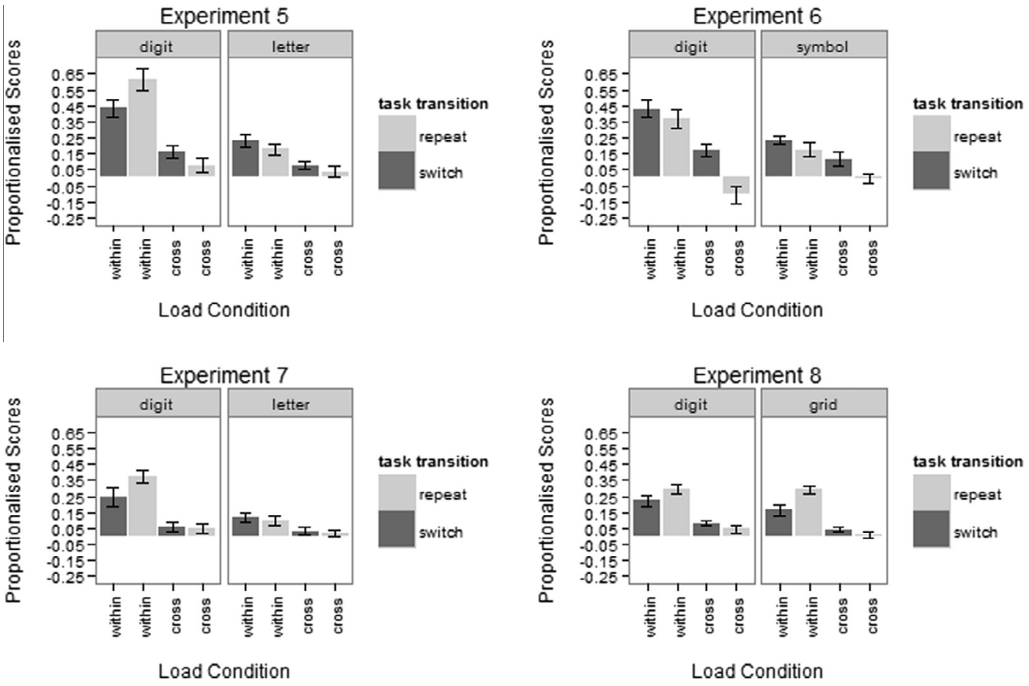

Fig. 4. Proportional within-task (within) and cross-task (cross) load effects by task transition (repeat vs. switch trials) and task (digit task or grid task = procedural load task; letter or symbol task = declarative load task) for Experiments 5–8. Error bars denote 95% within-subjects confidence intervals.

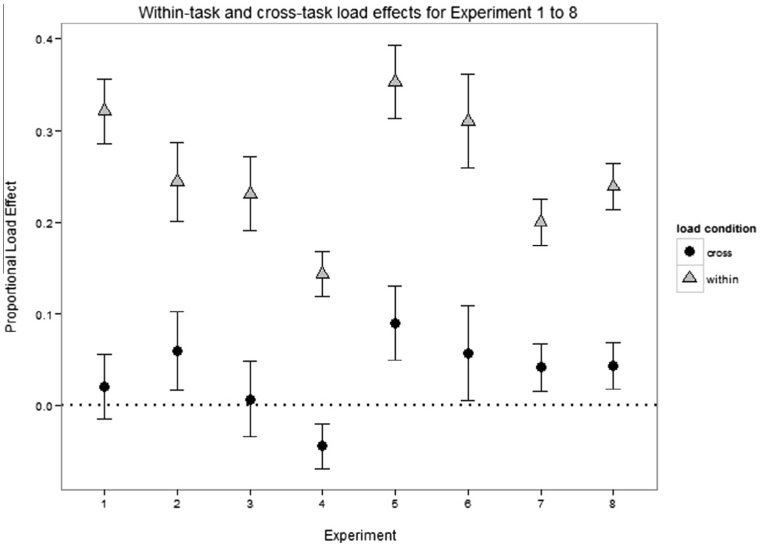

Fig. 5. Proportional within-task and cross-task load effects in Experiments 1–8. Error bars denote 95% within-subjects confidence intervals.

Experiment 6

In Experiment 5 we used the same sets of stimuli (i.e., letters and digits) as in Experiments 1, 2 and 4. In contrast, our Experiment 3 involved non-verbal stimuli for the declarative load task, combined with verbal stimuli for the procedural load task, and this combination appeared to have reduced cross-task load effects, at least on RTs. Therefore, we decided to test whether we also obtain a cross-task load effect when the two declarative load tasks load WM with verbal and non-verbal stimuli, respectively, as in Experiment 3.

Method

Twelve new participants (mean age 22.9 years, 9 women) took part in Experiment 6. We replaced the consonants of the letter-recognition task of Experiment 5 by the symbols we used in Experiment 3 (see Fig. 3). Otherwise the tasks, procedure, and design were the same as in Experiment 5.

Results

RT data. The ANOVA for RTs is shown in Table 4, mean RTs for each condition are listed in Appendix F. The top right panel of Fig. 4 shows the proportional load effects by task transition and task. We observed significant task-switch costs of 175 ms. Furthermore, we again obtained significant within-task load effects (353 ms) and cross-task load effects (64 ms). Task transition moderated both load effects (see Fig. 4, upper right panel). Within-task load effects increased from 281 ms for task-repetition trials to 426 ms for task-switch trials, contrary to the reduction of within-task load effects in switch trials observed in Experiments 1–5. Cross-task load effects were −64 ms for task-repetition trials and 192 ms for task-switch trials. The interaction between within-task and cross-task load was influenced by task transition. Whereas the cross-task load effect increased by 144 ms with high within-task load for repetition trials, it decreased by 154 ms for high within-task load in switch trials (see Appendix F).

Accuracy data. The ANOVA for accuracy is shown in Table 5, and the mean accuracy rates for each condition are listed in Appendix F. Mean overall accuracy was 87.6%. All main effects for task transition, within-task load and cross-task load were comparable to the RT data. Switching between tasks produced a cost of 7.0%. Increasing within-task load resulted in a drop in accuracy by 14.6%, whereas increases in the cross-task load led to a decrease in accuracy of 4.4%. Task transition interacted with cross-task load, mirroring the RT data (6% for switch trials compared to 3% cross-task load effect for repetition trials).

Discussion

As Experiment 5, the present experiment yielded a significant cross-task load effect when both load manipulations affected declarative WM. In contrast to Experiment 3, which used comparable sets of materials and yielded a cross-task load effect on accuracy only, here we obtained cross-task load effects for both RT and accuracy. Together with Experiment 5, Experiment 6 further supports the notion that cross-task load effects are larger when both tasks tax declarative WM than when the two tasks involve WM loads from different representational domains.

Experiment 7

In Experiment 7, we considered the relevance of the variability of the memory sets again, asking specifically whether a cross-task load effect is also obtained with constant sets for both declarative load tasks. In comparison with the previous Experiments 5 and 6, this experiment will show whether memory-set variability is necessary for obtaining a cross-task load effect when both tasks load WM with declarative representations.

Method

Twelve new participants (mean age 21.2 years; 8 women) took part in Experiment 7. We used the constant letter sets as in Experiment 4 and this time also created constant digit sets for the recall task. There were no item repetitions within any memory set. In all other respects, the tasks, procedure, and design were the same as in Experiments 5 and 6.

Results

RT data. The ANOVA for the RTs can be found in Table 4, and the mean RTs for each condition are listed in Appendix G. The lower left panel of Fig. 4 shows the proportional load effects in Experiment 7. Participants were 100 ms slower in case of a task switch compared to a task repetition. Having to hold more items in WM impaired performance, as reflected in the within-task load effect of 155 ms. We also observed a significant cross-task load effect of 33 ms. Furthermore, within-task load effects were moderated by task transition.[2] In case of a task switch, the within-task load effects decreased to 145 ms compared to 166 ms in case of a task repetition (see Appendix F).

Accuracy data. The ANOVA of the accuracies is shown in Table 5, and the mean accuracies for all conditions are listed in Appendix G. Overall accuracy was 92.6%. Task transition yielded a main effect mirroring RT data. Increasing within-task load lowered accuracy by 9.5%. We failed to observe a cross-task load effect in the accuracy data. Task transition interacted with within-task load; in case of a task repetition within-task load effects were 3.2% and increased to 15.7% in case of a task switch, opposite to the RT data.

Discussion

In Experiment 7, when keeping both declarative memory sets constant, we observed the same main effects for task transition, within-task load and cross-task load as in Experiments 5 and 6. Furthermore, whereas cross-task load effects on RTs were negative in Experiment 4 (combining declarative and procedural loads with constant sets), for Experiment 7 we observed positive cross-task load effects.

Experiment 8

In our final experiment, we tested whether a cross-task load effect is also observed when procedural load is manipulated in both tasks. Experiment 8 combined two choice-RT tasks to implement two manipulations of load on procedural WM. The first task was identical to the procedural load task used in Experiments 1–4; the second task used grid positions as stimuli, matching the material of the declarative task in Experiments 1–6. In the recognition

---

[2] The interaction of within-task load and task transition was significant only in log RT data but did not survive when mean RT were further trimmed, i.e. all RTs above three *SD* from the cells' mean of the participant were removed.

task in Experiments 1–6, participants had to hold bindings between items and grid positions in WM for making recognition judgments. In our new procedural task, participants were required to map grid positions to vocal responses.

Method

Sixteen new participants (mean age 22.1 years, 14 women) took part in Experiment 8. One procedural task was the same as in Experiments 1–4 (i.e., digit task): Participants had to press one of the function keys in response to digits. The second procedural task involved responding to one out of two potentially filled cells in a 3 × 3 grid (for low load), or to one out of four potentially filled cells (for high load) by uttering a syllable ("bu", "do", "ta" or "ki"). On each trial, one cell of the grid was filled, and participants were asked to respond as quickly as possible with the syllable assigned to that cell. In low-load trials, the cells to the left and to the right of the central cell could be filled; on high load trials, the four corner cells could be filled. We will refer to this task as the grid task. Responses were recorded by a voice key and coded online by the experimenter. We kept the stimulus–response mappings consistent for both tasks, but counterbalanced them across participants. Task sequence within a run was random, with the restriction that the digit and the grid tasks were each performed in 50% of the trials. The procedure was the same as in the preceding experiments with one slight difference. To ensure that participants formed procedural representations of stimulus–response mappings in both tasks, rather than relying on declarative representations of the instructed mappings, participants practiced both tasks individually with both load conditions before starting the first session (e.g., single task runs). Practice comprised 25 responses for each digit/filled grid cell in each load condition (2 vs. 4) distributed over 5 (load 2) or 10 runs (load 4). Before each run participants were reminded of the SR mapping in both tasks. As in the remaining experiments, they performed four sessions working on all four possible load conditions combining the two tasks. Practice trials were performed prior to the first session. Data were analyzed as before.

Results

RT data. The ANOVA results for the RT data are shown in Table 4, and the mean RTs for each cell of the complete design are listed Appendix H. Proportional load effects are presented in the lower right panel of Fig. 4. Participants were 94 ms slower in the grid task than in the digit task. Furthermore, they were 114 ms slower when they had to switch the task compared to when they had to repeat the task. Participants were 199 ms slower with high compared to low within-task load. Most important, though, we observed a significant cross-task load effect (36 ms). Both within-task load and cross-task load interacted with task transition. Whereas within-task load decreased by 60 ms in task-switch trials (169 ms) relative to task-repetition trials (229 ms), cross-task load increased by 37 ms in task-switch trials (55 ms) compared to task-repetition trials (18 ms).

Accuracy data. The ANOVA results of the accuracy rates are presented in Table 5, and the mean accuracies for all conditions can be found in Appendix H. Mean overall accuracy was 96.4%. We found no significant main effect for cross-task load. However, descriptively, all effects went in the same direction as in the RT data. Task transition yielded a main effect, as did within-task load. Participants showed switch costs of 0.8%. Performance decreased by 2.8% in runs with high within-task load. The within-task load effect was larger (3.9%) in case of a task repetition than in task-switch trials (1.6%). Within-task load interacted with cross-task load in that the effect of cross-task load was −0.5% for low within-task load, and 3.7% for high within-task load.

Discussion

For Experiment 8, in which both tasks loaded WM with procedural representations, a cross-task load effect on RTs was observed. Therefore, across Experiments 5–8, loading WM with either declarative or procedural representations in two different tasks produced consistent cross-task interference.

Cross-experiment analyses

To summarize, we observed consistent cross-task load effects on RTs, and in two out of four cases also on accuracies, when two tasks in a dual-task paradigm loaded WM with the same type of representation (Experiments 5–8). In contrast, we obtained no effect of cross-task load (Experiment 1), a reversed cross-task load effect (Experiment 4), or a cross-task load effect in only one of the two dependent variables (Experiments 2 and 3), when each task in a dual-task paradigm loaded declarative and procedural WM differentially. We carried out an analysis across all experiments to formally test the hypothesis that declarative and procedural representations interfere less with each other than they interfere within each representational domain. To this end, we directly compared the cross-task load effects of Experiments 1–4 (load manipulation in different domains) to those of Experiments 5–8 (load manipulation in the same domain). We conducted the same 2 × 2 × 2 × 2 ANOVA as before with the data of all experiments and included a between-subjects variable (type of manipulation) indicating whether both declarative and procedural WM were loaded (Experiments 1–4) or whether the load manipulation in both tasks concerned either declarative or procedural WM (Experiments 5–8). Here we focus specifically on the cross-task load effect and its interaction with type of manipulation.

We found the predicted interaction of type of manipulation (mixed declarative and procedural load vs. only declarative or only procedural load) and cross-task load in the RTs, $F(1,98) = 17.63$, $p < .001$, $\eta_p^2 = .153$. Mean cross-task load effects were 11 ms in Experiments 1–4, in which declarative load was manipulated in one task and procedural load in the other; this effect did not differ significantly from zero, $t(47) = 1.28$, $p = .207$. The mean cross-task load effect was 62 ms for Experiments 5–8, in which either declarative or procedural load was manipulated in both tasks. This load effect differed significantly from zero, $t(51) = 6.32$, $p < .001$. This result shows that when loading declarative and procedural WM concurrently, the amount of cross-task interference is significantly smaller than when both tasks load either declarative or procedural WM (see Fig. 5).

For the accuracy data the same pattern was observed, at least descriptively. Cross-task load effects were small (0.9%) but significantly different from zero ($t(47) = 3.64$, $p = .001$) for the experiments loading both declarative and procedural WM, and increased to 1.7%, $t(51) = 4.45$, $p = .001$, when the load manipulations affected only declarative or procedural WM in both tasks. However, the interaction between cross-task load and type of manipulation failed to reach significance, $F(1,98) = 1.22$, $p = .272$, $\eta_p^2 = .012$.

General discussion

In the present study we investigated whether the distinction between declarative and procedural representations, well established in research on long-term memory, can be extended to working memory, as suggested in a recent theoretical framework by Oberauer (2009). As in studies dissociating verbal and visuo-spatial WM, we used a dual-task approach testing for a double dissociation to achieve this aim. In the first four experiments (Experiments 1–4) we manipulated the declarative load (number of item-position bindings to be remembered) in a recognition task, and procedural load (number of stimulus–response mappings) in a choice task. We found no, small, or even reversed cross-task load effects (i.e., effects of load of the currently irrelevant task on the currently relevant task) in these experiments. In contrast, more consistent cross-task load effects emerged in Experiments 5–8, in which both tasks loaded either declarative or procedural WM. Thus, taken together, the data of the current study can be taken as support for the assumption that declarative and procedural WM can be distinguished. This double dissociation is consistent with the framework proposed by Oberauer (2009).

Even though cross-task interference between declarative and procedural WM was negligible in size, it was still present, especially in task-switch trials (see Fig. 6). There was a consistent interaction between task transition and cross-task load in six of our eight experiments. In the remaining two experiments (Experiments 2 and 7), the interaction was descriptively present and went in the same direction. Thus, in all experiments, the cross-task load effects were larger (or less negative, in the case of Experiment 4) in task-switch trials than in task-repetition trials (see Fig. 6).

The reason for this interaction could be that in task-repetition trials the currently irrelevant memory set or task set has already been irrelevant during the preceding trial. It is possible that people partially remove the irrelevant set from WM before or during execution of the current task. When the same task has to be carried out in the next trial, the previously irrelevant set is still irrelevant, and cross-task interference is reduced because the irrelevant set has already been removed from WM. In contrast, in task-switch trials, the currently irrelevant set has just been relevant, and therefore is represented strongly in WM when the trial starts. People might still be able to remove the irrelevant set from WM to some extent on switch trials, but certainly not as much as on repeat trials. Moreover, the previously irrelevant set is now relevant. To the extent that it has been removed on the preceding trial, it now has to be brought back into WM by retrieving it from LTM. Our finding of substantial task-switch costs in every experiment is consistent with this idea: The task-switch cost could reflect the time for retrieving the previously irrelevant memory set or task set from LTM (Mayr & Kliegl, 2000; Oberauer et al., 2013).

If this interpretation is correct, the task-repetition trials underestimate the size of the cross-task load effect, whereas the task-switch trials provide a more accurate assay of the cost of holding two memory sets in WM simultaneously. We therefore conclude that, whereas cross-task load effects are smaller between declarative and procedural WM than within each domain, they are not entirely absent. As noted in the introduction, a complete absence of cross-task load effects would not be expected even if declarative and procedural WM were entirely distinct sub-systems because our declarative-load tasks inevitably required procedural representations, and our procedural-load tasks likewise involved some declarative representations in WM.

We also investigated in which way cross-task interference is affected by mapping variability. The qualitative pattern of results did not change, but constant mappings led to smaller within-task and cross-task load effects. This decrease in load effects might be due to the fact that constant mappings enable the formation of a robust representation of the memory set or the task set in LTM. This LTM representation can be drawn upon to stabilize and improve the WM representation of that set, so that the item-position bindings in a memory set and the stimulus–response bindings in a task set are highly distinctive, thereby minimizing interference between them.

In sum, we found evidence for a double dissociation between declarative and procedural WM. This finding can be interpreted in two ways. One interpretation follows the approach of subdividing WM into different sub-systems, as proposed for instance by Baddeley (1986) for the distinction between verbal and visual-spatial WM. This is the interpretation originally proposed by Oberauer (2009, 2010).

An alternative explanation for the double-dissociation pattern arises from theories of WM that assign interference between representations an important role in determining performance. For instance, Oberauer et al. (2012) applied an interference-based computational model of WM to the double dissociation of verbal and visual-spatial WM. This double dissociation arises naturally from the model because verbal contents and visuospatial contents are represented in separate (though perhaps partially overlapping) feature spaces. Representations in separate feature spaces do not interfere with each other. The finding of very little interference between declarative and procedural representations in WM could be explained in the same way: If

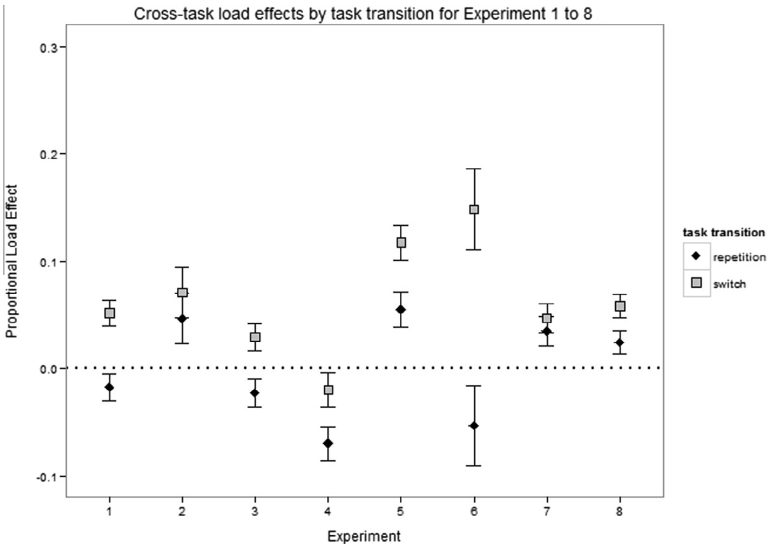

Fig. 6. Proportional cross-task load effects for task repetition and task switch trials in Experiments 1–8. Error bars denote 95% within-subjects confidence intervals.

declarative and procedural representations involve different feature dimensions, that is, if they reside in largely separate feature spaces, they do not interfere with each other. Yet interference between them is not entirely absent, suggesting that declarative and procedural representations do have slightly overlapping feature spaces. Some degree of overlap between declarative and procedural representations should be expected given the consistent evidence for shared feature representations for stimuli and responses (Hommel, 2004).

The larger cross-task load effects in Experiments 5–8 can be understood as arising from larger overlap of the feature dimensions involved in the representations required in two declarative-load tasks, or in two procedural-load tasks. If this assumption is correct, then it is not because of the materials used in the experiments: The majority of experiments involved materials with very little apparent overlap in feature dimensions – for instance, digits and non-nameable symbols (Experiment 6) share no obvious feature dimensions, and likewise, the retrieval cues (colors and spatial locations) in Experiments 5–7 have no obvious feature dimensions in common. Moreover, the cross-task load effects were not systematically higher in those experiments that used materials with highly overlapping feature spaces (i.e., letters and digits, Experiments 5 and 7) than in those that did not (digit and symbols, Experiment 6; digits and spatial locations, Experiment 8). For an interference explanation of our results to work, we have to assume that all declarative representations share some feature dimensions that they do not share with procedural representations, and conversely, all procedural representations share some feature dimensions that they do not share with declarative representations. At present we can only speculate about what these dimensions are. One thing that procedural representations have in common and sets them apart from declarative representations is that their features code intended actions. For instance, a procedural representation of a digit or a word specifies how to speak it, whereas a declarative representation of a digit or a word specifies its sound and its meaning – a related distinction has been made between phonological output and input representations (Jacquemot & Scott, 2006). Likewise, a procedural representation of a spatial location specifies the instruction to act toward that location, for instance to press the left key, whereas a declarative representation of a spatial location specifies the location of an object or an event. The difference between declarative and procedural representations lies not in their content but in their role in cognition. If this role is coded in terms of features that form part of the representation, then it is plausible that declarative and procedural representations occupy different feature spaces defining their role in cognitive processes.

Finally, an interference theory of WM can also explain why within-task load effects were larger than cross-task load effects in all experiments (see Fig. 5). Increasing within-task load increases the number of items (declarative representations) or response options (procedural representations) in the set from which one item or response needs to be selected. This increases the chance of confusion between selection candidates. For instance, increasing the declarative load from two to four letters means that there are more chances to confuse the letter in the tested position with another letter from the current memory set. In contrast, increasing cross-task load means to increase the number of representations in a different set than the one from which one element must be selected. When the current task is letter recognition, the number of digits concurrently held in WM does not increase the chance of confusion because people do not tend to confuse letters with digits.

We conclude that the double dissociation of declarative and procedural loads on WM supports the conceptual distinction between declarative and procedural WM. The two kinds of WM can be conceptualized either as separate sub-systems, or as partially non-overlapping feature spaces within a unitary WM system. In any case, declarative and procedural representations in WM interfere with each other only very weakly. This relatively benign interference enables the WM system to hold a set of declarative representations available as the objects of processing, and at the same time hold a task set available to guide the cognitive or overt actions on those objects.

Acknowledgments

The research reported in this article was supported by a Grant (Nr. 100014_130113) from the Swiss National Science Foundation to Miriam Gade, Michel D. Druey, and Klaus Oberauer.

The authors thank Carla de Simoni, Nina Ingold, Franziska Kühn, Laura Schaad, and Olivia Schär for their help in data collection. The authors thank André Vandierendonck and two anonymous reviewers for comments on a previous version of this paper.

Appendix A

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 1.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Letter recognition	Switch	887.1	67.0	932.4	104.0	1093.3	96.9	1129.4	81.9
	Repetition	753.8	53.3	773.9	72.8	1112.4	96.2	1025.6	103.2
CRT task (digits)	Switch	694.0	131.2	746.8	145.9	970.5	230.5	1020.7	198.3
	Repetition	505.3	60.6	510.1	67.6	847.7	252.4	854.9	259.8
		Accuracy							
Letter recognition	Switch	96.8	3.4	96.7	3.4	98.5	1.9	97.8	2.3
	Repetition	98.5	2.0	98.9	1.2	97.5	2.5	98.1	1.8
CRT task (digits)	Switch	97.6	2.1	97.5	4.2	96.2	3.9	97.2	4.2
	Repetition	99.3	1.3	98.5	2.0	97.6	1.9	97.7	3.1

Note. CRT = choice reaction time.

Appendix B

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 2.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Letter recognition	Switch	1006.1	276.6	1083.3	257.9	1192.0	357.3	1272.3	305.8
	Repetition	828.5	178.4	848.4	194.5	1070.3	247.6	1111.2	223.4
CRT task (digits)	Switch	939.0	288.7	1027.4	314.7	1199.1	205.3	1255.2	224.0
	Repetition	748. 3	154.1	833.8	251.8	1085.0	251.6	1122.7	244.3
		Accuracy							
Letter recognition	Switch	92.3	5.0	87.6	6.8	92.2	4.7	92.0	5.2
	Repetition	96.8	3.5	96.1	4.2	95.2	4.6	92.8	4.8
CRT task (digits)	Switch	97.8	2.8	98.9	1.4	98.1	2.2	96.4	4.7
	Repetition	99.3	1.8	98.7	3.1	97.9	2.3	98.5	1.6

Note. CRT = choice reaction time.

Appendix C

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 3.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Symbol recognition	Switch	1094.2	314.4	1147.0	260.3	1354.3	396.8	1336.5	389.1
	Repetition	853.5	227.9	851.4	171.7	1167.7	280.4	1117.5	279.5
CRT task (digits)	Switch	1122.8	219.8	1211.8	292.0	1365.0	232.5	1388.2	240.5
	Repetition	857.9	203.0	870.5	157.0	1213.1	213.6	1158.7	242.1
		Accuracy							
Symbol recognition	Switch	95.3	3.1	91.3	7.2	92.6	6.0	89.3	9.1
	Repetition	95.9	3.6	95.2	4.0	94.4	5.9	91.1	7.0
CRT task (digits)	Switch	98.7	2.2	99.2	1.4	98.5	2.1	98.2	3.0
	Repetition	99.6	1.1	99.6	0.1	98.6	1.2	97.4	2.4

Note. CRT = choice reaction time.

Appendix D

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 4.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Letter recognition	Switch	842.8	141.4	889.5	166.2	949.4	164.8	949.5	138.6
	Repetition	801.8	123.7	820.8	122.1	935.1	178.1	913.6	143.8
CRT task (digits)	Switch	665.5	142.6	692.8	130.1	893.2	179.2	754.1	188.2
	Repetition	551.7	101.5	556.5	112.8	778.3	175.4	565.7	110.8
		Accuracy							
Letter recognition	Switch	81.8	7.5	83.0	4.6	98.0	2.6	98.1	2.0
	Repetition	95.9	4.5	93.4	4.0	96.4	3.8	95.8	2.1
CRT task (digits)	Switch	97.0	3.0	96.9	2.4	98.9	2.2	97.0	2.7
	Repetition	98.6	2.8	98.3	2.9	98.9	2.6	96.2	3.3

Note. CRT = choice reaction time.

Appendix E

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 5.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Letter recognition	Switch	1127.8	246.6	1192.4	235.6	1380.6	216.9	1527.8	321.9
	Repetition	1090.0	268.1	1097.8	240.7	1264.5	281.5	1348.7	301.4
Digit recall	Switch	930.9	140.1	1112.7	196.0	1459.5	169.8	1712.4	310.4
	Repetition	704.7	111.5	736.1	135.9	1300.4	169.8	1433.8	251.7
		Accuracy							
Letter recognition	Switch	97.0	3.0	96.5	4.3	64.9	2.9	64.3	4.6
	Repetition	97.9	2.9	97.8	3.7	90.2	3.4	87.7	4.7
Digit recall	Switch	95.4	5.8	89.5	13.3	84.7	18.1	77.8	17.5
	Repetition	98.6	3.6	93.9	8.5	92.1	14.6	87.3	11.1

Appendix F

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 6.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Symbol recognition	Switch	903.4	140.5	1118.7	222.6	1249.3	222.6	1326.4	292.7
	Repetition	958.7	187.8	920.4	135.2	1117.2	235.5	1131.1	185.2
Digit recall	Switch	846.7	165.1	1168.9	262.4	1506.0	320.2	1658.9	499.1
	Repetition	992.0	270.8	757.7	179.7	1250.3	198.3	1252.3	192.4
		Accuracy							
Symbol recognition	Switch	96.0	2.9	93.0	6.6	64.2	4.7	61.4	4.7
	Repetition	96.1	3.6	96.6	5.5	87.6	6.0	80.0	11.7
Digit recall	Switch	96.2	4.4	90.1	6.4	91.9	11.8	80.3	18.2
	Repetition	95.4	6.0	96.2	3.8	91.3	9.6	85.4	18.5

Appendix G

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 7.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
Letter recognition	Switch	788.3	114.6	810.7	131.0	882.2	123.5	915.6	150.7
	Repetition	747.4	101.1	753.6	95.0	815.6	108.1	843.3	154.0

Appendix G (*continued*)

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
Digit recall	Switch	696.1	155.1	745.7	158.4	887.4	142.3	934.6	138.8
	Repetition	529.9	93.5	568.1	119.6	783.6	140.5	819.8	175.8
		Accuracy							
Letter recognition	Switch	94.8	2.6	96.7	2.1	66.5	2.5	66.4	2.5
	Repetition	96.0	3.8	96.9	1.9	92.3	2.9	92.2	2.1
Digit recall	Switch	98.7	1.6	96.9	2.5	95.1	5.0	96.0	4.2
	Repetition	99.3	1.6	99.3	1.7	96.7	4.4	97.6	2.6

Appendix H

Means and standard deviations (*SD*) for the reaction time (in ms) and accuracy (in percent) data for Experiment 8.

		Within-task load Low Cross-task load				Within-task load High Cross-task load			
		Low		High		Low		High	
Task	Task transition	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
		Reaction time							
CRT task (digits)	Switch	816.4	193.5	871.1	234.2	976.7	212.0	1006.6	245.9
	Repetition	668.9	101.4	702.2	116.4	937.8	209.5	915.4	149.9
CRT task (grid)	Switch	705.1	168.6	755.3	195.0	878.6	257.7	962.6	292.2
	Repetition	581.1	127.7	622.1	141.5	809.1	246.1	828.2	204.9
		Accuracy							
CRT task (digits)	Switch	96.0	4.3	97.9	2.9	96.7	2.2	95.3	3.4
	Repetition	98.9	2.6	98.6	2.1	97.2	3.7	95.1	3.1
CRT task (grid)	Switch	96.3	2.8	97.1	2.7	96.6	3.0	92.2	9.5
	Repetition	98.8	2.7	98.4	1.8	97.1	2.5	90.0	14.4

Note. CRT = choice reaction time.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89*, 369–406.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Earlbaum Associate.

Baddeley, A. D. (1986). *Working memory*. Clarendon Press.

Baddeley, A. D., & Hitch, G. J. (1974). Working memory. *The Psychology of Learning and Motivation, 8*, 47–89.

Bartlett, M. S. (1947). The use of transformations. *Biometrics, 3*, 39–52. http://dx.doi.org/10.2307/3001536.

Cocchini, G., Logie, R. H., Sala, S. D., MacPherson, S. E., & Baddeley, A. D. (2002). Concurrent performance of two memory tasks: Evidence for domain-specific working memory systems. *Memory & Cognition, 30*, 1086–1095. http://dx.doi.org/10.3758/BF03194326.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *The Behavioral and Brain Sciences, 24*, 87–114. discussion 114–185.

Ecker, U. K. H., Oberauer, K., & Lewandowsky, S. (2014). Working memory updating involves item-specific removal. *Journal of Memory and Language, 74*, 1–15. http://dx.doi.org/10.1016/j.jml.2014.03.006.

Hick, W. E. (1952). On the rate gain of information. *The Quarterly Journal of Experimental Psychology, 4*, 11–26.

Hommel, B. (2004). Event files: Feature binding in and across perception and action. *Trends in Cognitive Sciences, 8*, 494–500. http://dx.doi.org/10.1016/j.tics.2004.08.007.

Jacquemot, C., & Scott, S. K. (2006). What is the relationship between phonological short-term memory and speech processing? *Trends in Cognitive Sciences, 10*, 480–486. http://dx.doi.org/10.1016/j.tics.2006.09.002.

Kiesel, A., Steinhauser, M., Wendt, M., Falkenstein, M., Jost, K., Philipp, A. M., et al. (2010). Control and interference in task switching – A review. *Psychological Bulletin, 136*, 849–874. http://dx.doi.org/10.1037/a0019842.

Lewandowsky, S., Geiger, S. M., Morrell, D. B., & Oberauer, K. (2010). Turning simple span into complex span: Time for decay or interference from distractors? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 36*(4), 958–978. http://dx.doi.org/10.1037/a0019764.

Lewandowsky, S., Geiger, S. M., & Oberauer, K. (2008). Interference-based forgetting in verbal short-term memory. *Journal of Memory and Language, 59*, 200–222. http://dx.doi.org/10.1016/j.jml.2008.04.004.

Mayr, U., & Kliegl, R. (2000). Task-set switching and long-term memory retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*, 1124–1140. http://dx.doi.org/10.1037/0278-7393.26.5.1124.

McLeod, P. (1977). A dual task response modality effect: Support for multiprocessor models of attention. *Quarterly Journal of Experimental Psychology, 29*, 651–667. http://dx.doi.org/10.1080/14640747708400639.

Oberauer, K. (2001). Removing irrelevant information from working memory: A cognitive aging study with the modified Sternberg task. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27*, 948–957. http://dx.doi.org/10.1037/0278-7393.27.4.948.

Oberauer, K. (2002). Access to information in working memory: Exploring the focus of attention. *Journal of Experimental Psychology: Learning Memory and Cognition, 28*, 411–421.

Oberauer, K. (2005). Binding and inhibition in working memory: Individual and age differences in short-term recognition. *Journal of Experimental Psychology: General, 134*, 368.

Oberauer, K. (2009). Design for a working memory. *Psychology of Learning and Motivation, 51*, 45–100.

Oberauer, K. (2010). Declarative and procedural working memory: Common principles, common capacity limits? *Psychologica Belgica, 50*, 277–308.

Oberauer, K., & Kliegl, R. (2001). Beyond resources: Formal models of complexity effects and age differences in working memory. *European Journal of Cognitive Psychology, 13*, 187–215.

Oberauer, K., Lewandowsky, S., Farrell, S., Jarrold, C., & Greaves, M. (2012). Modeling working memory: An interference model of complex span. *Psychonomic Bulletin & Review*. http://dx.doi.org/10.3758/s13423-012-0272-4.

Oberauer, K., Souza, A., Druey, M. D., & Gade, M. (2013). Analogous mechanisms of selection and updating in declarative and procedural working memory: Experiments and a computational model. *Cognitive Psychology, 66*, 157–211. http://dx.doi.org/10.1016/j.cogpsych.2012.11.001.

Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116*, 220–244.

Ricker, T., Cowan, N., & Morey, C. (2010). Visual working memory is disrupted by covert verbal retrieval. *Psychonomic Bulletin & Review, 17*, 516–521. http://dx.doi.org/10.3758/PBR.17.4.516.

Risse, S., & Oberauer, K. (2010). Selection of objects and tasks in working memory. *The Quarterly Journal of Experimental Psychology, 63*, 784–804.

Souza, A. da. S., Oberauer, K., Gade, M., & Druey, M. D. (2012). Processing of representations in declarative and procedural working memory (2006). *Quarterly Journal of Experimental Psychology, 65*, 1006–1033. http://dx.doi.org/10.1080/17470218.2011.640403.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory, 82*, 171–177. http://dx.doi.org/10.1016/j.nlm.2004.06.005.

Sternberg, S. (1969). Memory-scanning: Mental processes revealed by reaction-time experiments. *American Scientist, 57*, 421–457. http://dx.doi.org/10.2307/27828738.

Stevens, M. Ë., Lammertyn, J., Verbruggen, F., & Vandierendonck, A. (2006). Tscope: A C library for programming cognitive experiments on the MS Windows platform. *Behavior Research Methods, 38*, 280–286. http://dx.doi.org/10.3758/BF03192779.