2015

Are declarative and procedural working memory functionally analogous? Testing working memory using the task span

Kaitlin Reiman
Lehigh University

Are Declarative and Procedural Working Memory Functionally Analogous?

Testing Working Memory Using the Task Span

by

Kaitlin M. Reiman

A Dissertation

Presented to the Graduate and Research Committee

Of Lehigh University

In Candidacy for the Degree of

Doctor of Philosophy

in

Psychology

Lehigh University

May 18, 2015

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Kaitlin M. Reiman
Are Declarative and Procedural Working Memory Functionally Analogous? Testing Working Memory Using the Task Span

____

Defense Date

____
Catherine Arrington
Dissertation Director

____

Approved Date

Committee Members:

____
Jason Chein (External)

____
Almut Hupbach

____
Padraig O'Seaghdha

Table of Contents

List of Figures

List of Tables

Abstract

Working memory is one system that supports the flexible nature of cognition. It is a highly debated topic, with several competing models of organization. One recent model (Oberauer, 2009) is progressive, drawing on past work with well-supported adaptations. The model describes two subsystems, declarative (representing items that can be manipulated such as lists of words and numbers) and procedural (representing procedures used for manipulation, like processes for mental math) working memory. Both include three components with successively higher levels of activation. The base layer is an activated portion of long term memory, followed by a capacity-limited portion from which candidates for attentional selection are drawn (the bridge and region of direct access), and a focus of attention that highlights a particularly relevant item. One assumption of this model is that declarative and procedural working memory are functionally analogous, which means that they may operate according to equivalent mechanisms. However, much of the work investigating this model has explored declarative information. This dissertation used the task and memory span paradigms (Logan 2004) to evaluate this selected assumption of functional analogy in Oberauer's (2009) model.

Experiment 1 established the task span as a test of procedural working memory, a novel methodological contribution. Experiment 2 tested the fate of recently activated information in the intermediary working memory stores. Experiment 3 tested the characteristics of information capacity in the intermediary procedural store. All three experiments shared methodology, the task span, in different variations. This dissertation tested the strong assumption that the two subsystems are functionally analogous. Results from Experiment 1 indicate that the task span is a suitable methodology for testing procedural working memory. Experiment 2 was inconclusive because of failure to replicate Oberauer's past work possibly due to floor effects in the memory span. Experiment 3 yielded results that are consistent with Oberauer's model. Results from all experiments are discussed in the context of Oberauer's theoretical model and a computational version of that model. In all, the current data provide support for the assumption of a functional analogy, and leave open questions of independence and of mechanisms of declarative and procedural subsystem communication.

Introduction

Working memory is a complex system that influences many aspects of cognition. From planning and reasoning to language comprehension, working memory supports a diverse array of behaviors. In fact, working memory has been shown to predict about half of the variance in general intelligence and reasoning abilities (Oberauer & Lewandowsky, 2011), and is related to many critical functions for successful navigation of our daily lives like problem solving (Wiley & Jarosz, 2012) and language comprehension (King & Just, 1991). Working memory may underlie the process by which we sustain all thought (Baddeley, 2010). As such it is important to understand the organization of this system so that we can accurately predict and explain behavior in diverse domains. Historically there has been much conflict about many aspects of this construct. For example, the organization, capacity, and extent to which working memory can be trained for transfer of skills to other cognitive tasks are all routinely debated in the literature (Richardson, Engle, Hasher, Logie, Stolzfus, & Zacks, 1996).

There is no universally agreed upon definition of the working memory construct, but in the current dissertation I will be referring to working memory as a dynamic cognitive system that allows for the temporary storage of some information while simultaneously processing other information. That is, the working memory system goes beyond a simple short term store allowing only for rehearsal of simple lists, but additionally allows for more complex cognitive operations along with that basic maintenance. As I think about working memory, it is separable from but linked to long term memory. The working memory system is also capacity limited, as only small subsets of information can be held active at any one time. Indeed, the very first mention of working memory in the literature comes from Miller (1956), introducing the well-known “7 ± 2” measure of working memory capacity. This value is still relevant when considering basic short term maintenance, but capacity with regard to manipulation and processing of information is decidedly more complex.

I will begin by reviewing several of the most relevant models of how the working memory system is organized. These models include that of Baddeley and Hitch (1974), a multi-store model with domain-specific information representation, and those of Cowan (1988) and Oberauer (2009), which are more dynamic and less reliant on domain-limited structures. This dissertation speaks specifically to the general organization of the working memory system, and I focus on Oberauer’s (2009) model to test one of its main assumptions. This model was selected because it is the most recent working memory model, and as such it has received little empirical attention beyond the experiments used to support its development. Following introduction of the theoretical models, I describe in detail a computational model based on Oberauer’s model, and how this computational model can be made to fit the relevant experimental paradigm used in this project—the task span (Logan, 2004). I then describe three studies that investigate Oberauer’s model, with the goal of understanding the organization of the dynamic working memory system that is so important for cognitive behavior as well as one paradigm that may be useful for studying it. To preview, results of the current experiments suggest that Logan’s (2004) task span is a suitable choice for testing Oberauer’s (2009) model of working memory. Additionally, at least one main assumption of Oberauer's model (regarding the mechanisms of capacity for two types of information) holds after testing using that paradigm. The results are then discussed in the context of Oberauer's and other's working memory theories.

Review of Selected Working Memory Models

Baddeley and Hitch. The original model of working memory is the historically most cited model, and it is still dominant in classrooms and textbooks forty years after its conception (Baddeley & Hitch, 1974). In fact, when the construct of working memory is brought up, Baddeley and Hitch's is likely to be the only model of working memory with which many people—even experts in psychology—are familiar. This reputation is well deserved, as the model was ground breaking when originally developed, and still is able to account for much empirical work.

When the model was originally published, it was formally introduced as including three-parts: the central executive and two slave systems, the phonological loop and the visuo-spatial sketchpad. The central executive was described as having several supervisory characteristics. For example, it was expected to have some storage capacity, the ability to cross-talk with long term memory, and most importantly, it was described as being able to allocate attention and resources to the two supplementary slave systems. According to recent modifications to the model, the central executive may be further fractionated into multiple elements representing different executive processes (Baddeley, Chincotta, & Adlam, 2001). For example, the central executive may have different resources dedicated to updating the contents of working memory, inhibition of irrelevant or conflicting information, and set shifting between multiple tasks or cognitive processes, but all were presumed to fall under the central executive umbrella.

The “slave” systems are named as such because they have no capacity to make decisions or control the flow of information through the mind. Instead, they are passive storage systems, and information fades rapidly if it is not kept active through rehearsal. The phonological loop processes auditory information including language and other forms of acoustic messages, while the visuo-spatial sketchpad processes visual, spatial, and potentially kinesthetic information (e.g. Smyth & Pendleton, 1989). The phonological loop and visuo-spatial sketchpad are both passive stores subject to trace decay and both operate through general rehearsal mechanisms (Logie, 1995), but are seen as being completely distinct from one another.

While the central executive was originally described as having storage capacity and the ability to interact with long term memory, in an update Baddeley (2000) modified those characteristics by adding a new fourth component: the episodic buffer. This addition was in response to many criticisms of the original model, in particular that the central executive was not sufficient for accounting for the influences of long term knowledge on working memory. The episodic buffer is expected to integrate information from the phonological loop and visuo-spatial sketchpad with information from long term memory. This component is called the “episodic” buffer because it is assumed to hold chunks of information, and the representations are expected not to be domain specific like the slave systems. This means that the episodic buffer has the critical role of binding information from the slave systems with the end goal of a unified cognitive experience.

The episodic buffer is mainly a storage device—not designed for manipulation, although it is able to integrate information from different systems into coherent messages. Instead, in this updated version of the model, the central executive takes on only the role of manipulating and updating information without storing it. Baddeley has also stated that the episodic buffer is what gives us our experience of consciousness, as it allows for keeping track of the experiences we are currently having while reflecting on what has happened in the past, all in real time (Baddeley, 2007).

Despite this update (Baddeley, 2000) to the original model of working memory (Baddeley & Hitch, 1974), there are still several gaps in the story. While the interaction with long term knowledge has been addressed, Baddeley's model is not consistent with a large body of neurological evidence (Postle, 2006). There is limited neurological support for the idea of separate storage buffers for visual/spatial and auditory information (Conway, Moore, & Kane, 2009; cf. Smith & Jonides, 1999; Smith, Jonides, Koeppe, Awh, Schumacher, & Minoshima, 1995). Indeed, if there were locations in the brain dedicated to the rehearsal and maintenance of domain specific information, there would be many more than two. There are multiple dissociations between different *kinds* of verbal and spatial information (e.g. Postle & D'Esposito, 2003; Mecklinger, Gruenwald, Besson, Magnie, & Von Cramon, 2002) and different *kinds* of auditory information (e.g. Shivde & Thompson-Schill, 2004), which might suggest that the visuo-spatial sketchpad and the phonological loop should be further subdivided into an ever increasing number of slave systems. Additionally there is evidence for olfactory (Dade, Zatorre, Evans, & Jones-Gottman, 2001) and tactile (Harris, Miniussi, Harris, & Diamond, 2002) working memory, but no slave systems posited to account for those domains. There have been no attempts to reconcile these dissociations with Baddeley's original model (Postle, 2006), and most importantly, to do so would stretch the boundaries of parsimony. That is, fractioning the working memory system into an ever increasing number of components to account for each new piece of evidence could continue in an almost limitless fashion. Even if Baddeley and Hitch's model can be adapted to include new slave systems, it is not necessarily prudent to do so. Eventually, the most simplistic and potentially realistic solution becomes to begin the theoretical development process anew, instead of trying to account for the mounting pile of disconfirming evidence within the outdated framework. As such, among working memory researchers this model has largely fallen out of favor and new explanations of this cognitive system have arisen. In particular, researchers have begun to think about working memory as being an emergent system, which relies on existing sensory and action systems to represent information of different modalities with coordination through attention instead of dedicated storage buffers (Postle, 2006). Two models that could be compatible with this new representation of working memory are described below.

The Embedded-Processes Model. A substantial departure from Baddeley and Hitch's conceptualization of working memory was developed by Cowan (1988, 1995, 1999), termed The Embedded-Processes Model. This model is such a departure because Cowan focuses primarily on the attentional aspect of working memory, and specifically considers working memory to be a component of long term memory, rather than the supplemental temporary passive storage systems of Baddeley's model (Baddeley, 2010). Cowan does not assert domain specific storage, arguing that while information from within the same domain with the same code will certainly cause interference, this does not mean that information with separate codes must be processed separately (Cowan, 1995). In articulatory suppression, speaking irrelevant words while trying to maintain other verbal information disrupts memory performance, but the same irrelevant speech does not disrupt visual information. Baddeley argues that the two types of verbal information must be competing for the same resource while the visual information does not. Cowan instead suggests that while the verbal information creates domain-specific interference, the visual information is not necessarily stored completely separately.

Cowan's model includes an activated portion of long term memory along with a focus of attention manipulated by separate executive processes. Information is input into perceptual systems via brief sensory stores, i.e. those that are responsible for sensory persistence, like echoic and eidetic memory. The focus of attention is a subset of the activated portion of long term memory and is comprised of about four pieces of information. According to Cowan's model, there are only two levels of activation in working memory for any given piece of information: within the focus of attention and outside of the focus. This focus is a mechanism for maintaining several memory items at once in an activated state that are the center of awareness. The items in the focus of attention are maintained in a privileged state compared to other pieces of information in activated long term memory, and are somewhat protected from interference

(Gilchrist & Cowan, 2011). Cowan's model is labeled as being embedded, because he conceptualizes the focus of attention being a contained subset of the activated portion of long term memory that, in turn, is embedded in all of the information present in long term memory. The focus of attention is controlled by both voluntary and involuntary processes that will influence what information is drawn into awareness and granted higher levels of activation (Cowan, 1999). For example, items may be brought into the focus by goal-driven processes controlled by a central executive, or stimulus-driven processes like attentional orienting (Cowan, 1999). This provides great flexibility concerning what factors can influence the contents of working memory.

Baddeley and Cowan disagree concerning the way in which the long-term and working memory systems interact. According to Baddeley, the episodic buffer is a separate cognitive mechanism where items from long term memory and the slave systems can be stored and manipulated by the central executive, but Cowan argues that the working memory system can hold the "addresses" of long term memory information in order to access relevant memoranda without any sort of transfer (Cowan, 1999). For example, when remembering long lists of words compared to long lists of non-words, memory is much better for the words, with the only difference between the two lists being that there is long term memory representation of the actual word list to activate (Cowan, 1995).This mechanism allows for a flexible change between items in the focus and activated long term memory. Essentially, Cowan believes that related items in long term memory are so easily accessed that they function in a very similar way to items in the focus of attention (much like the long term working memory model proposed by Ericsson and Kintsch, 1995).

Oberauer's Three Embedded Components Working Memory Model. Another model of working memory has recently been developed to more fully encompass several empirical findings in the literature (Oberauer, 2009). Oberauer developed his model in accordance with six principles of the working memory system he identified as being critical. He proposed that a functional working memory system must be able to: 1. create and maintain mental representations, 2. selectively manipulate those representations, 3. flexibly adapt to new and changing goals, 4. update the previously created representation rapidly, 5. draw from stored information in long term memory, and 6. adjust the organization and level of activation of that information for more permanent long term storage (Oberauer, 2009). These points are not widely argued, but the way in which Oberauer (2009) has designed his model based on these points is somewhat of a departure from some of the classic thinking about this construct mostly because of the distinction between declarative and procedural information, and the embedded components, described below.

This model separates the processing of two distinct forms of information, but does not divide processing according to specific perceptual domains as Baddeley's does. The two systems are differentiated based on the characteristics of the information they handle. One subsystem manages representations of declarative information, which are elements that could potentially be manipulated, also called the "objects of thought". This includes things like lists of digits or words (e.g. a phone number or grocery list), or any mental information that makes up “content” available for manipulation. This subsystem would handle most of the sort of information that Baddeley theorized the visuo-spatial sketchpad, phonological loop, and episodic buffers encompassing. The complementary subsystem is called procedural working memory, and manages information that could be used to manipulate declarative information (i.e. the legal moves a chess pieces can make) [1]. These representations are cognitive or motor procedures bound to their relevant conditions and prospective outcomes, and separate from simple verbal descriptions of or instructions for a task. In an experimental context, the mentally represented task rules used to respond to presented stimuli would be covered in the domain of procedural information. Overall, the declarative subsystem has the responsibility of making mental representations available to be processed by the procedural subsystem.

The organization of this model is similar to that of Cowan (1988) in that it is viewed as a set of attentional processes interacting with other cognitive systems (Oberauer, 2009), but clarifies the role of long term memory as a mechanism for learning structural representations that can be later retrieved into working memory, znd

[1] Historically in long term memory there has been a tradition of differentiating between declarative and procedural memory (Squire, 2004). Declarative information is that which can be explicitly or consciously accessed, while procedural information is implicit and cannot be consciously accessed. These different types of long term memory are represented in different brain areas and subject to substantially different processing. The distinction between these two types of information had not been integrated into a working memory model until Oberauer’s model was introduced.

adds an additional level of focus that pinpoints one item or response. See figure 1 for a visual depiction of the model.

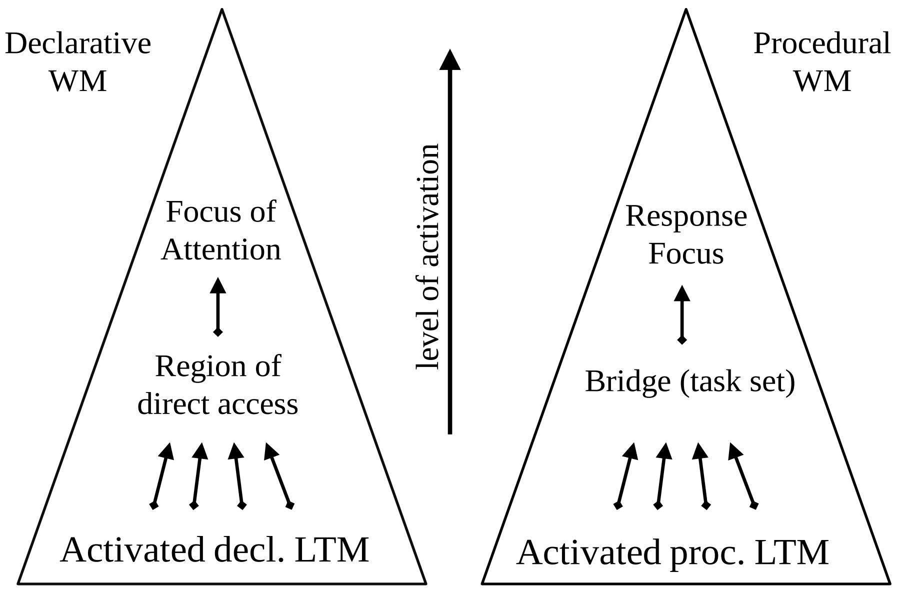

Figure 1. A visualization of the hierarchical activation in declarative and procedural working memory (Oberauer, 2009). Figure adapted from Gade, Druey, and Oberauer (2011).

Each of the two subsystems, declarative and procedural, is comprised of three components, and each component is characterized by what is believed to be a certain level of activation, resulting in increasingly focused levels (Oberauer, 2009). The base component includes an *activated portion of long term memory*—subsets of available thought objects and procedures, which are active at any given time. The intermediary component includes a small amount of information, which is immediately accessible and ready to be integrated into a mental representation. This component is responsible for the development of new structural representations in working memory; this is item one of Oberauer's critical requirements of a working memory system (Oberauer, 2009). The *intermediary store* creates dynamic bindings in order to develop these new representations, and these bindings can be flexibly updated according to task demands. This flexibility in manipulation—or item three on Oberauer's list—occurs with the contribution from other executive processes that monitor goals and adjust the parameters of performance according to those goals. The third and final component in the three embedded components model is the *focus of attention*, which highlights the piece of information about which we are currently thinking (Oberauer, 2009). This component is responsible for selection and manipulation of specific items or processes, this is item two of Oberauer's critical working memory requirement. The three components of the model are characterized by successively increasing levels of activation—not completely separate stores or "containers" (Oberauer, 2013). As with Cowan's model, these three levels are essentially "embedded"; that is, the candidates for the focus of attention are drawn from the intermediary store which are in turn drawn from the activated portion of long term memory (Oberauer, 2009).

Like many models of long term memory, Oberauer (2009) assumes that long term memory is associative by nature, with related representations activating one another automatically and rapidly from any relevant cue. There is also a highly flexible relationship between the activated portion of long term memory and the intermediary store, with representations rapidly shifting between the two levels. This feature exemplifies item five of Oberauer's working memory requirements—information can be drawn into the region of direct access and the bridge from long term memory. Critically, the region of direct access and the bridge can also be “partially decoupled” (Oberauer, 2009) from activated long term memory, such that information in the intermediary store can be rapidly updated without modifying long term memory representations. For example, if a master chess player holds in mind the current state of a chess board in order to manipulate that representation and find the best move, all possible chess board orientations or memories of past boards with that same organization will not also be manipulated and altered (Oberauer, 2009). This represents Oberauer’s item four of a working memory model. Finally, the transfer of information in the intermediary store and activated long term memory is bidirectional; as information is manipulated in the intermediary store, it can be transformed into chunks in long term memory for more permanent encoding—item six from Oberauer’s list.

In declarative working memory, the activated portion of long term memory leads to the intermediary store called the “region of direct access”, while the primary focus is called “the focus of attention” (Oberauer, 2009). In procedural working memory, the activated portion of long term memory leads to the “bridge”, named for the command bridge of a ship, or the bridge between a stimulus and its response, and the focus of attention is called the “response focus”, which temporarily holds a recently selected procedural response. The focus of attention in Oberauer’s model is similar to the focus of attention in Cowan’s model, but differs in the nature of capacity. While Cowan argues that this focus can contain about four unique chunks of information, Oberauer argues that the intermediary store would contain about that many items or response options while the focus of attention would hold only one item or response at the maximum level of awareness. However, capacity limitations are not inherently based on a number of items in Oberauer's model, although his description is sometimes interpreted in that way. Instead, Oberauer suggests that capacity limitations in the intermediary stores arise from retrieval competition from related bindings and interference from other representations in the intermediary store at that moment (Oberauer, 2009).

Bindings can be described in one of two ways, depending on which subsystem is in question. For procedural working memory, the bindings are associations between conditions and actions, which according to Oberauer (2009) in an experimental context is the binding between stimulus and response. In declarative working memory, the bindings are associations between items and their contexts. In an experiment this could be the association between serial position in a list or spatial location and a particular item in a visual display. These bindings provide a direct means of retrieving information that should be currently active; contexts (serial positions or stimuli for example) provide a means of accessing connected representations (items stored in memory or appropriate responses for example) (Risse & Oberauer, 2010). So the capacity of the intermediary component is dependent on what information is being maintained. When items need to be maintained with more precision, or items have more elements, they will exhaust more of the capacity of the intermediary store (Oberauer & Eichenberger, 2013). The nature of this capacity speaks to the critical role of the intermediary store in information processing; bindings are the way in which information is represented in this store, and the ease of updating these bindings allows for one of the hallmarks of human cognitive processing: flexibility. Competition and interference between these bindings contributes to limitations, whether that competition arises from similar elements or from an abundance of information to be maintained.

To more fully understand this three components model, consider a brief thought experiment. Imagine you will be presented with four digits: 6, 3, 8, and 11. You are then expected to multiply each of these digits by 4. You are to perform this task mentally, without the aid of any notes. As far as maintenance is concerned, you can rehearse these four digits in turn, or encode a chunk to remember them all at once. But in order to actually engage in the processing component of this task, the multiplication, you must focus on each item one by one. Simultaneously performing this transformation on all four digits in the mental space would be impossible. In this analogy, the four digits would be active at the level of the region of direct access, pulled from a subset of numbers in activated long term memory, while to perform the operation each would be raised to the focus of attention for processing in conjunction with the response focus. It is clear from this example that Cowan's model of the focus of attention, which may be able to hold four items at once, is limited when considering how simultaneous processing might occur. Additionally, it is helpful to clarify the distinction between declarative representations of instructions, and actual procedures. Declarative representations, or verbal descriptions of procedural information are useful for remembering how to perform some tasks, but these instructions are not actual procedural representations themselves—they are helpful in guiding the binding of relevant procedures in the bridge, but Oberauer (2009) assumes that the immediate control of behavior is guided by actual procedural representations.

A body of work in support of Oberauer's model has its foundation in these simple arithmetic tasks during which participants maintain and manipulate information held in mind. Straightforward evidence for the three components of Oberauer's model comes from work using this sort of paradigm (Oberauer, 2002). In this paradigm, participants were presented with two lists of digits to memorize, and following memorization, one list was marked as irrelevant for the immediately following activity. Participants were then instructed to perform basic mathematical calculations on the memorized digits, with the appropriate item cued based on the spatial location of the arithmetic instruction. After completing several of these mathematical manipulations, participants were asked to recall both the active list on which the operations had just been performed, and the passive list that was originally deemed irrelevant.

Results were in line with the three component model (Oberauer, 2002). That is, the set size of the active list impacted performance (i.e. slowed RTs) while the set size of the passive list has no influence on performance. When information was deemed irrelevant for the short term, but would be necessary later, the items were maintained in activated long term memory (passive list). When items were present in the active list and had to be accessed regularly, they were maintained in the region of direct access. Critically, bindings between the items and their locations (the information necessary for selecting on which object the arithmetic operation must be performed) compete for selection, and more bindings, or larger set sizes, results in slower response times (RTs) for the arithmetic task. However, there was no such effect on RTs to the arithmetic updating as a function of the set size of the irrelevant list, presumed to be stored in activated long term memory. These two levels of activation are functionally distinct in this way—the level of activation of the two types of lists, passive or active, changes the degree to which the set size of the list influences performance. Additionally evidence for the single-item focus of attention comes from the finding that repetitions of the selected item yielded an RT benefit (Oberauer, 2002; 2003). Items that were just in the focus of attention stayed there and did not require additional processing time.

The constraints of Oberauer's model have also been explored by considering the role of attention to the contents of working memory. Recent work examining Oberauer's model has adopted the use of a *retro-cuing paradigm* to address such questions of the role of attention (Rerko, Souza, & Oberauer, 2014; Rerko & Oberauer, 2013; Souza, Rerko, & Oberauer, 2014). In this paradigm, participants are presented with a memory array, and then on most trials this array is followed by a cue or multiple sequential cues to specific items in that array. After cuing, participants complete a recall task. For example, in one study participants were presented with an array of colorful dots in a circle, followed by 0, 1, or 2 sequential cues, and subsequently a probe item that either matched or mismatched the actual item in the array to which the participant had been most recently cued (Rerko & Oberauer, 2013). Participants had to indicate whether the probe item matched the item initially presented in that cued location or not. Results from this paradigm indicated a retro-cue benefit, that is, items to which attention was directed after presentation yielded higher accuracy in responding to the later probe, even when the information was held in working memory and no longer presented externally. So, attention to the contents of working memory can aid recall performance. But this benefit is not entirely reliant on focal attention, as attention can be directed to other things following the central cue without disrupting performance (Rerko, et al., 2014). These experiments provide support for the flexibility of movement of information between different levels of activation, depending on relevant information provided post hoc following encoding. When information is currently relevant (cued most recently), there is increased activation for that item and it is more likely to be selected as the focus of attention. When information is not currently relevant (cued in the past or not cued from the original memory array) the activation levels are lower than in the recently-cued case, and the item is less likely to be selected. Depending on the circumstances presented in the experiment, the activation for particular items can be rapidly adapted.

Critically for this dissertation, Oberauer (2009) theorizes that the declarative and procedural working memory subsystems are functionally analogous, but independent from one another. This means that each subsystem of working memory is expected to work in similar, if not identical ways, although the type of information contained in the subsystem differs. These two subsystems are also expected to operate separately without sharing capacity (Gade, Druey, & Oberauer, 2012). There is some support for this assumption, described below, but the model is quite new, and the bulk of the evidence examining it focuses on declarative working memory. By extension, for most working memory phenomena there is no direct comparison between predominantly declarative and predominantly procedural tasks. This dissertation aims to develop evidence for the mechanisms of procedural working memory, specifically to address the assumption of a functional analogy between the two subsystems.

The analogy between procedural and declarative working memory has preliminary support from Oberauer's work. The procedural mechanisms involved in switching between task sets are assumed to be analogous to declarative mechanisms involved in switching between memory sets (Oberauer, et al. 2013). While this does not mean that the exact same phenomena must be observed for the two types of information, the mechanisms in support of declarative and procedural working memory are expected to be comparable. When one effect is observed in declarative working memory, a similar effect should be observed in procedural working memory, while the reverse should hold true as well. Comparisons between task switching (procedural) and list-switching (declarative) experiments have suggested that there are many analogous behaviors exhibited when participants switch between declarative memory items organized in short lists and procedural task sets (Souza, Oberauer, Gade, & Druey, 2012). Participants learned up to three task sets and up to three lists of three items on each trial, and were then prompted to switch between lists and switch between tasks (or repeat) based on presented cues (e.g. Garavan, 1998). There was a cost of switching between lists, just as there is a cost of switching between tasks in task switching (Rogers & Monsell, 1995). This switch cost in list switching also persisted even with additional preparation time, which is fundamental to task switching (Rogers & Monsell, 1995). There was also evidence of mixing costs in declarative working memory, as RTs on repetition trials in blocks with only repetitions were faster than repetition trials in mixed blocks with both repetitions and switches, again, a common finding in task switching literature (Rubin & Meiran, 2005). These findings provide preliminary evidence for the assumption of analogous processing in declarative and procedural working memory These findings also highlight the reliance of Oberauer and colleagues (2012) on using task switching to study the procedural element of the model. While task switching is a paradigm especially useful for studying cognitive control and may require procedural working memory, it was not designed to study working memory, and does not require much input from the actual memory portion of working memory, beyond the maintenance of stimulus-response bindings. In most types of task switching, cues are presented to participants in advance of every stimulus. This reliance on task switching suggests the necessity of developing a test of the procedural subsystem more suited for that role. An appropriate option will be discussed later in the introduction.

Direct manipulations of object switching and task switching as declarative and procedural tasks, respectively, have indicated that these two subsystems are analogous but separate in their capacity limits (Risse & Oberauer, 2010). These experiments were a combination of object switching and task switching. Participants had to retrieve from memory both task sets and lists of objects, and the objects served as cues for the task switching. The characteristics of mappings between the tasks or between the objects and their cues were orthogonally manipulated. Sometimes the task sets were stable across trials while the objects were not, sometimes the objects were stable across trials while the task sets were not, and sometimes both the objects and the task mappings varied across trials. The researchers were interested in the characteristics of selecting objects and selecting tasks. The number of objects stored in working memory negatively affected object-switch costs, but did not affect task- switch costs. According to the authors, objects and tasks are selected separately, and they interpret this dissociation as evidence for separate and independently functioning declarative and procedural working memory components (Risse & Oberauer, 2010). This conclusion also explains such findings as the benefit of repeating objects when tasks change, and repeating tasks when objects change. This dissociation would not likely be observed unless objects and tasks are selected and processed separately.

Computational model. Oberauer and colleagues (2013) have recently developed a connectionist computational model based on the findings of past research, with the intention of predicting behavior that would be in accordance with the three component model Oberauer (2009) introduced. A depiction of the computational model is presented in figure 2, and each element will be described in detail below.

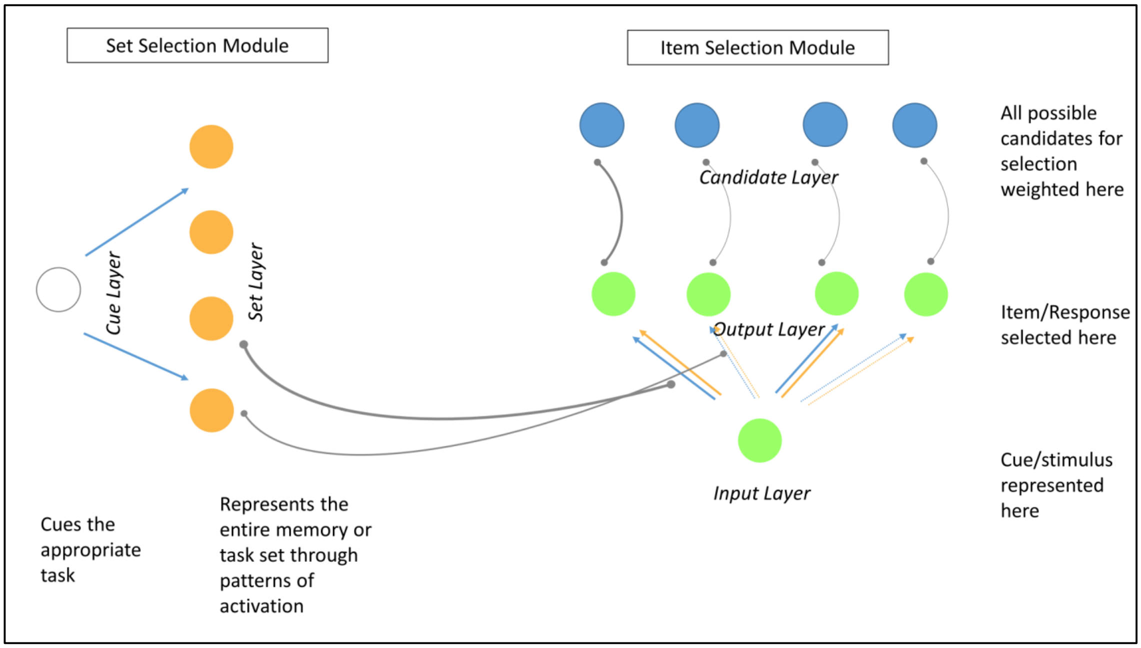
Set Selection Module
Item Selection Module
All possible candidates for selection weighted here
Candidate Layer
Cue Layer
Set Layer
Output Layer
Item/Response selected here
Input Layer
Cue/stimulus represented here
Cues the appropriate task
Represents the entire memory or task set through patterns of activation

Figure 2. A depiction of the computational model of Oberauer’s three embedded components model of working memory (adapted from Oberauer, et al., 2013).

In the computational model the authors describe two modules within their connectionist framework: an item-selection module and a set-selection module. The item module selects individual representations, like objects or responses, and the set module selects collections of items, like memory or task sets. The item module is comprised of three units: a candidate layer, an output layer, and an input layer. These three layers do not map directly onto the three components of Oberauer’s model (activated long term memory, direct access region/bridge, and focus of attention/response focus) with a one-to-one relationship, but those components can be represented within these three layers and the set-selection module. In figure 2, green items represent the focus of attention or response focus (top layer in figure 1). This includes the current cue or stimulus and the associated response or item—the pieces of information most relevant for actually selecting and making a response. Orange items represent the intermediary store, the region of direct access or the bridge (middle layer in figure 1). This includes fast changing bindings between items and responses, and currently activated task or memory sets. Blue items represent activated long term memory (bottom layer in figure 1). This includes the representations of all items in the candidate layer, connections between cues and memory/task sets, and slow changing bindings between items and responses.

The set selection module itself allows for rapid switching among different sets, like between multiple task sets or multiple memory sets. This rapid switching between representations in the set layer of the set-selection module is elicited by a cue layer in that module, which indicates what memory or task set is currently relevant. The item and set selection modules can be adapted for both declarative and procedural information; however, as most of the empirical work on Oberauer's model has been with declarative information, the current computational work tends to focus on that domain.

In outlining the computational model, Oberauer suggests that items represented in working memory are bound to their contexts, and each of these bound units exhausts some of the capacity of the working memory system (Oberauer, et al., 2013). The binding occurs in the item-selection module, specifically between the input and output layers. These bindings are represented by arrows connecting the two layers in figure 2. The input and output layers are connected by associations that can change, strengthen or weaken, either at a very slow rate over the long term based on adaptive behavioral adjustments or rapidly through flexible changes as online task demands change. The candidate layer, an element of activated long term memory, modifies the items to which it is connected by adjusting the likelihood that each particular item will be selected for the current iteration of a task based on internal or external task goals (e.g. individual choice or experimenter instructions). Increases in activation in the candidate layer will increase an item's likelihood of retrieval. The candidate layer represents item strength for each output item, setting the current memory array apart from other information in long term memory.

In declarative working memory, an item is any unit of information such as a number or an object. These possible numbers or objects would be represented in the output layer. The input layer handles context, which is anything that can be used as a cue for retrieving a particular item, like position on a list or position in space. So the input-output bindings in this case would be specific potential items in the output layer bound to specific serial or spatial positions in the input layer. In declarative working memory, as cues for recall occur, like an instruction to begin recalling the item presented in the first position of a sequence, the candidate layer weights the different memorized objects, and the first item is selected because it has the most activation from both the input and the candidate weightings.

The set-selection module works in a similar manner, with set cues or contexts attached to specific memory sets. When a memory set is selected based on the cue, it activates specific item units based on modifications to the input-output bindings. This is represented in figure 2 with the grey line connecting the two modules. The set selection module allows for the retrieval of chunks of memory items from long term memory. The memory sets retrieved could also be task sets from long term memory, which would allow for flexible adaptation among task sets in a dynamic multitasking environment.

In procedural working memory, the item selection module is more aptly referred to as the response module. Critically, this reference is strictly a change in the content (type of information) and the name, but not the organization. The identical computational model architecture for declarative and procedural information highlights the functional analogy between the two subsystems. In the response module, the output layer is adjusted for selecting responses from represented options rather than items. The input layer is adjusted for representing items or stimuli on which those procedures will be carried out. The set-selection module retrieves task sets from long term memory according to presented task cues (or task cues in memory). In a similar fashion to declarative working memory, the candidate layer will weight certain responses as most applicable as different stimuli are perceived and responses are selected.

The matrices of bindings between the input and output layers that connect items to contexts and responses to stimuli (stimulus-response mappings) can be rapidly updated depending on the circumstances of the environment in which the individual is working. Old bindings are removed and replaced by new ones using Hebbian learning (Oberauer, 2013). As contexts or cues become associated with new items or new responses, the initial network of bindings is rapidly modified to account for those changes. However, no matter how rapid this updating is, it is still slower than no updating. Therefore, on trials for which the same stimulus-response pair repeats (no updating of input-output bindings required), RTs are faster than when a new response must be matched to a new stimulus (updating of input-output bindings required). This distinction can account for such effects as the list switch and task switch costs (Gilbert & Shallice, 2002).

The computational model formalizes the assumptions of independent but analogous processing in declarative and procedural working memory, which I test in this series of experiments. As mentioned above, this model operates analogously for both procedural and declarative tasks (Oberauer, et al. 2013). Oberauer and colleagues (2013) explicitly state that the same computational model architecture can be used to account for findings and make predictions about both declarative and procedural working memory. This assumption is critical for the predictions I make in the experiments—if the same model architecture is used for both of the subsystems, then they are expected to operate according to the same mechanisms, but handle different types of information independently. However, the majority of the work on this computational model has focused on declarative content only, largely ignoring the procedural side. This dissertation tests model while focusing on this procedural element, addressing a gap in the working memory literature.

Open question about the model. While Oberauer's model of working memory is consistent with a large body of literature including task switching and modern working memory experiments (Oberauer, 2013; Oberauer, et al., 2013), there are several remaining questions open for inquiry. As mentioned above, Oberauer and his research group have done relatively little to examine the evidence in support of the procedural working memory dimension of the model, mostly focusing on the easier to investigate declarative component and drawing on findings from the task switching literature to support arguments about the procedural component. The full understanding of the procedural component is critical in order to solidify Oberauer's model as a good account of observed working memory phenomena. Additionally, a particularly interesting question regarding the procedural component is whether the relationship proposed between declarative and procedural working memory as functionally analogous is accurate. This is a strong claim, and while there are several pieces of evidence in support of it, there are many predictions made in line with the model that have not yet been tested. The current dissertation tests this assumption: *to what extent are the declarative and procedural working memory subsystems functionally analogous?* There are several ways in which this question can be tested. For example, the comparison between procedures and stored information in activated long term memory bears investigation. How analogous is storage and retrieval of task sets to storage and retrieval of broader content in long term memory? Additionally, how analogous are the processes by which information is selected for further analysis? Are procedures selected from a range of options by the same cognitive mechanisms as items are selected from a collection in a list? In this dissertation, I specifically focus on the region of direct access and the bridge—the two intermediary stores. I address this question by examining predictions made with the assumption that declarative and procedural working memory operate in comparable ways. Is information handled in the two intermediary stores through equivalent operations? Through investigation of this assumption, one of the most critical elements of Oberauer's model will be examined. If procedural and declarative working memory are functionally analogous, then another main assumption of Oberauer's model can be tested as well: the independence of the two components. If procedural and declarative working memory are not functionally analogous, then the functional analogy element of the model will certainly require revision, but the evidence could still point towards support for the functional independence assumption.

The Task Span and Its Application to Working Memory Research

One challenge in comparing declarative and procedural working memory comes from the difficulty in designing tasks that draw from these two subsystems independently and comparably. For one, it is essentially impossible to imagine a declarative experimental task that does not involve some sort of procedural response, or a procedural task that does not include manipulation of declarative information. Any time you are engaging in a procedural behavior, you must have some declarative information on which to act. Oberauer suggests this by indicating that declarative information stored in long term memory is highly related to associated procedures, and often the activation of a fact or an object will cue the activation of related procedures automatically (Oberauer, 2009). This idea of automatic activation implies that declarative and procedural information must mutually activate one another, but the maintenance and selection processes may still occur separately, keeping the two subsystems independent.

Oberauer and colleagues (2013) have drawn on a list switching paradigm to directly test the equivalence of switching among task sets and switching among memory sets. In their experiments, at the beginning of an experimental block participants learned two lists of three numbers—with each number bound to a location. One list was presented in blue text while the other was presented in red text. These lists contained a common item in a common location; this item was termed the congruent item. Participants then were instructed on each trial to perform a simple arithmetic operation on one of the items, cued only by the location of that item. They were also instructed to switch between the two lists, cued by the color of the arithmetic instruction. This led to participants ultimately sometimes performing list repetitions and sometimes performing list switches, while the items were sometimes congruent between lists and sometimes not. Participants were slower to switch between lists, much like the robust switch cost observed in task switching (e.g. Rogers & Monsell, 1995). Participants were also faster and less likely to make errors on congruent trials (e.g. Meiran & Kessler, 2008). Additionally, when lists repeated, participants benefited from making the same response. However, when lists switched, there was a cost associated with repeating the same response. These effects were interpreted as evidence for the similar processing of form and function of declarative and procedural working memory, as they are very similar to commonly observed effects in task switching, much like the residual switch costs, and mixing costs described above (Souza, et al., 2012).

There are limitations to the types of tasks chosen by Oberauer and colleagues thus far. In particular, the list switching and task switching paradigms are similar, but the task environments are not identical. The components of Oberauer's model are more easily applied to the declarative tasks than the procedural. Another option for testing declarative and procedural working memory that is useful for testing the organization in structure and function of these two subsystems is a combination of the memory span and task span (Logan, 2004). These paradigms provide a way to examine declarative and procedural elements of working memory in a directly comparable environment. In these related paradigms, participants encode simple sequences of information and then either recall those sequences (memory span) or perform associated tasks in the memorized order (task span). For example, participants may memorize a sequence such as "height, shape, height, shape, height, shape", and then view subsequently presented stimuli, performing tall/short categorizations on the first, third, and fifth items, and rectangle/oval categorizations on the second, fourth, and sixth items (see figure 3) in the task span, or viewing those same stimuli but simply recalling the task names without performing the categorizations in the memory span.

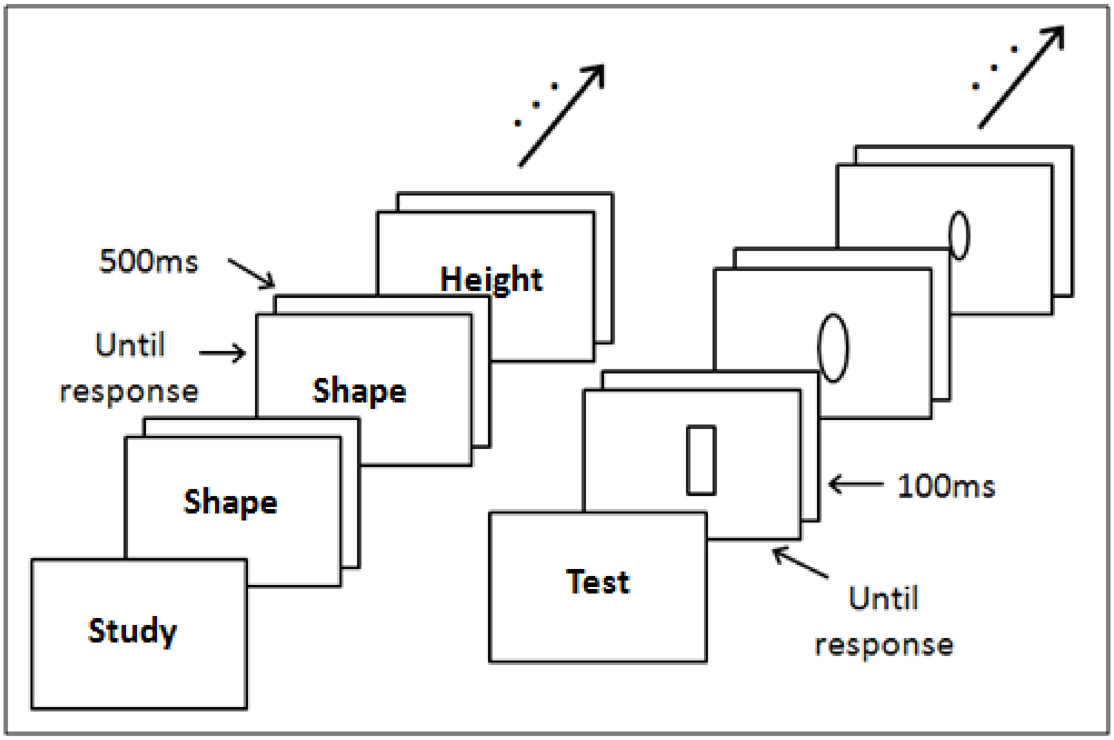

Figure 3. A depiction of the standard task span paradigm. Participants memorize the sequence presented during the study phase, and then perform associated tasks in the same order in response subsequently presented stimuli in the test phase (Logan, 2004).

The memory span is a version of the task span without task switching. So ultimately, the memory span is less complex because there are not multiple task switching procedures to carry out, but the underlying framework is the same for both of the paradigms. Both the memory and task spans recruit the cognitive processes necessary for storing and updating sequences of information, but only in the task span must participants also use that stored information to guide task performance.

Explanations of task span performance suggest that this paradigm is clearly situated at the intersection of set switching and working memory—participants must maintain a list of tasks in mind while simultaneously processing each stimulus that appears by executing tasks, rapidly shifting between task sets when the memorized sequence calls for it (Logan, 2004). This paradigm recruits additional control over simple cued task switching, evidenced by increased switch costs when memorized sequences are used to guide behavior rather than single task cues (Logan, 2007). Comparisons between the memory span and the task span reveal that memory for sequences is not different between the two span types, but performance with regard to RTs is delayed in the task span, (Logan, 2004; Logan, 2006) and participants were also more biased towards simple sequences in a voluntary version of the task span (participants generate their own sequences) than in the voluntary memory span (Reiman & Arrington, in prep.) Overall the data also suggest that the task span requires an interaction between storage in long term memory and the more active processing required in working memory (Logan, 2004). In connecting the task span to Oberauer's model, sequences may be stored at low levels of activation in activated long term memory and retrieved to the bridge in chunks for immediate task performance. This finding is evidenced by scalloped RT patterns; in past task span work, Logan (2004) found that RTs for the first item in a logical chunk (**A**BC**A**BC**A**BC) were slowed compared to later items in that chunk. When participants are actually performing tasks, their response is selected from candidates in the bridge and implemented through the response focus. The chunking behaviors mentioned above, which implicate activated long term memory in the task span, are consistent with both Cowan's (1988) and Oberauer's (2009) theorizing of embedded working memory organization.

Observation of participants’ behavior in the memory and task span paradigms suggests that these paradigms may be drawing from procedural working memory in different ways. In the memory span, participants must store in mind only task names, and they may not be activating knowledge of task sets or responses when they bring these cues to mind to recall (Logan, 2004). These task names are exactly the type of information that is handled by declarative working memory. In contrast, in the task span, participants must not only store and recall the task names, they must use this stored information to guide activation of different task sets. These task sets are prototypical examples of procedural information as conceptualized in Oberauer’s (2009) model, and are highly active in the task span. While the memory span cannot be a purely declarative task (i.e. some response must be given to indicate the memory item), and while the task span cannot be a purely procedural task (i.e. subjects store and recall declarative information in terms of task names), based on logical analyses the task span certainly draws on procedural working memory in a way that the memory span does not. Said another way, the memory span draws on declarative working memory in order to maintain the task names (declarative information), and the task span requires the same maintenance of the declarative task names. However, the memory span requires very little procedural working memory, as the key responses are so simple and do not vary across the test phase. While there is declarative component to the task span, it requires much more procedural input, as there are defined task sets for each task (see figure 4- “no load” section) and these must be updated from moment-to-moment across the test phase for the participant to respond correctly. The task span therefore draws on procedural working memory in a way that the memory span does not.

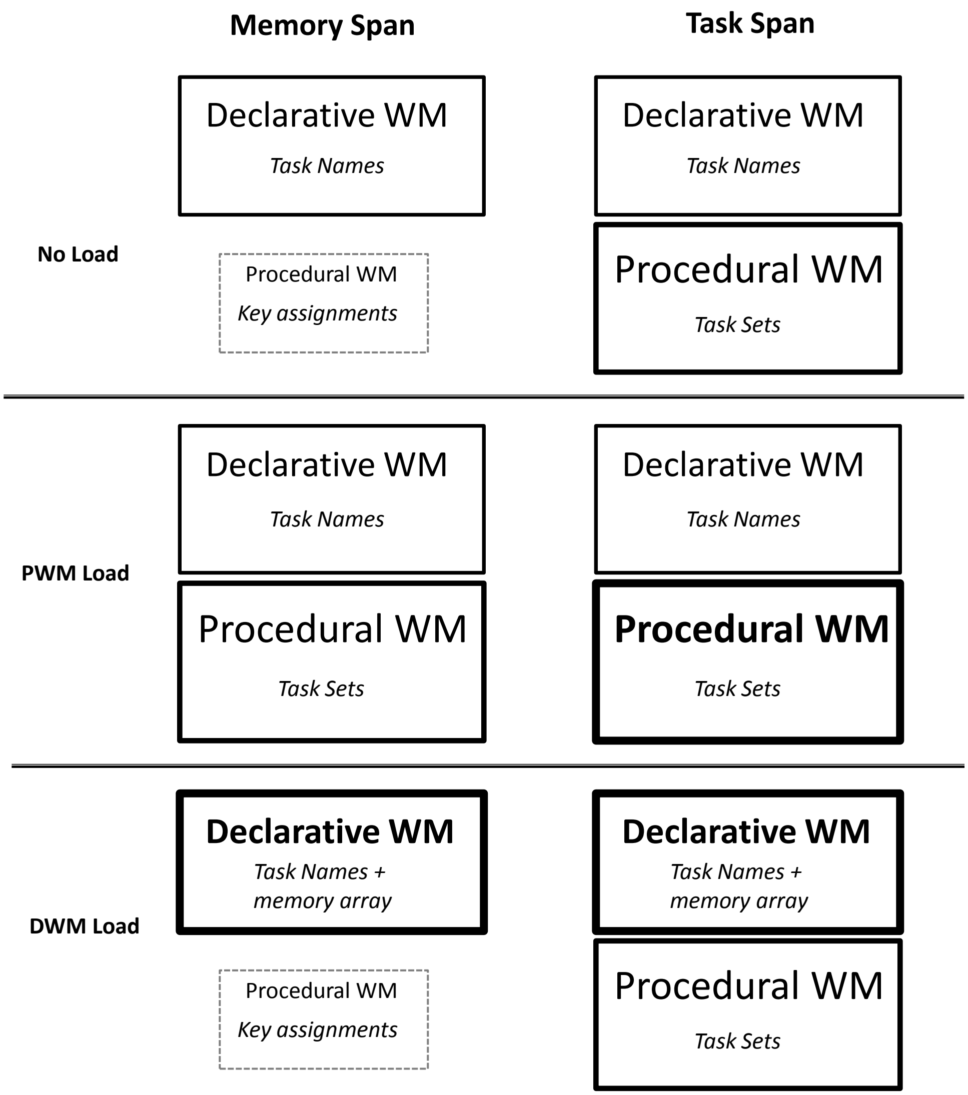

Figure 4. A depiction of the predicted demands on the procedural and declarative working memory systems stemming from added procedural and declarative loads in the task and memory spans. When a procedural working memory load is added to the memory span, performance will be similar to the task span without a load. Thicker boxes and bolder type indicate more demand on that system. Procedural and declarative loads were added to the memory span only in Experiment 1.

This analysis of the paradigms suggests that the memory span and task span may be appropriate for use in studying Oberauer's model of working memory, in particular whether the declarative and procedural subsystems are functionally analogous. Knowing that the task span places high demand on procedural working memory is especially valuable considering the prevalence of list memory tasks in past work testing Oberauer's model, and the imbalance of empirical data collected favoring declarative tasks.

Current Experiments

The overall purpose of this dissertation is to determine to what extent declarative and procedural working memory are functionally analogous, specifically focusing on the region of direct access and the bridge. The three experiments described below were designed to establish the task span as a valid platform to answer this question (Experiment 1), and to test two areas of Oberauer's working memory system where the assumptions of functionally analogous processing have not previously been tested (Experiments 2 and 3). Specifically, I investigate areas where declarative working memory has been examined, but procedural working memory has not. The first experiment was designed to solidify the memory span and task span as declarative and declarative + procedural working memory tasks, respectively. This experiment involved loading the memory span with either a declarative or procedural working memory task and then making comparisons of performance with each of these two different types of load and the task span. If the memory span with a procedural load results in performance similar to the task span, we can infer that the task span has procedural elements beyond the memory span. Experiment 1 is critical to establish the task span as a measure of procedural working memory—the main methodological contribution of this dissertation. Experiments 2 and 3 delved further into examining Oberauer’s theory, focusing on functional equivalence of the bridge and region of direct access. Experiment 2 tests the state of information that has been previously held in the intermediary store, investigating whether reinstated information is maintained with a privileged or inhibited state (i.e. increased or decreased activation compared to baseline) and whether the status of that reinstated information differs depending on whether declarative or procedural working memory is being considered. Experiment 3 tests how the capacity of the procedural intermediary store influences performance, and whether a change in the level of activation of the bulk of task-relevant information impacts performance in the same way that increased set sizes have been shown to interrupt performance in declarative tasks. These three experiments together are valuable in elucidating one of the critical assumptions of the three-embedded components model of working memory, namely that the procedural and declarative subsystems are functionally analogous.

Experiment 1: Can the task span be used as a test of procedural working memory?

Experiment 1 sought to test whether the task span (Logan, 2004) is an appropriate means to test Oberauer’s (2009) model of working memory. Developing such a connection would provide a richer way to test the assumptions and predictions of Oberauer’s model without relying on the task switching paradigm. In order to examine this question, Experiment 1 involved comparing performance on the task span to a similar memory span (only recall of task names) paradigm, with and without a declarative and procedural load. If participants’ performance on the memory span while under a procedural load, which is an identical experimental environment to the task span without the task switching requirement, looked like performance in the task span, we could infer that the task span has significant procedural elements. This argument will be laid out in more detail below. While logical analysis makes it quite obvious that the task span requires procedural elements, this finding has not been demonstrated in previous literature.

Considering the computational model of Oberauer’s framework is helpful to explain the basis of the current experiment. The item-selection module, and specifically the connection between the input layer and the output layer, is important for understanding the logic of the current experiment. In the memory span, declarative information is primarily being maintained. The application of the memory span to Oberauer’s computational model might look like what is represented in figure 5.

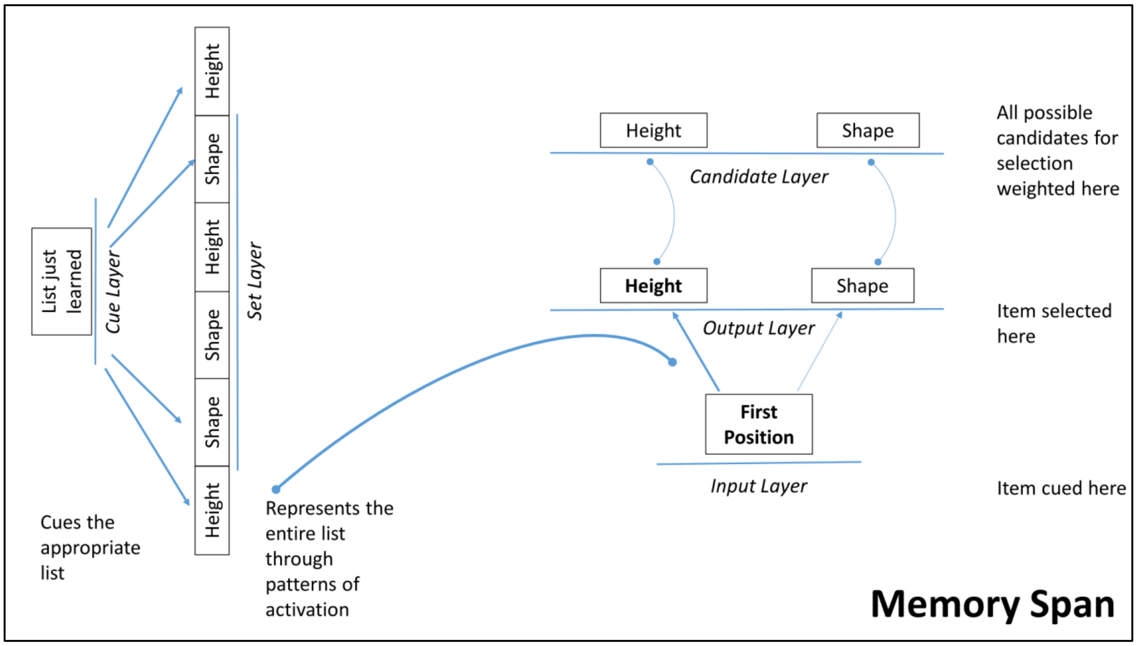

Figure 5. A depiction of how the memory span would be implemented in the computational model of Oberauer's work.

In this case, the entire span, or some subset of the span, is represented in the set selection module (left), and depending on the serial position, different binding strengths will result in a different task name being selected in the item selection module (right). The application of the task span to the computational model might require the same task selection process depicted above, *as well as* the response selection process depicted in figure 6. At this point it is unclear whether these processes occur in parallel or sequentially.

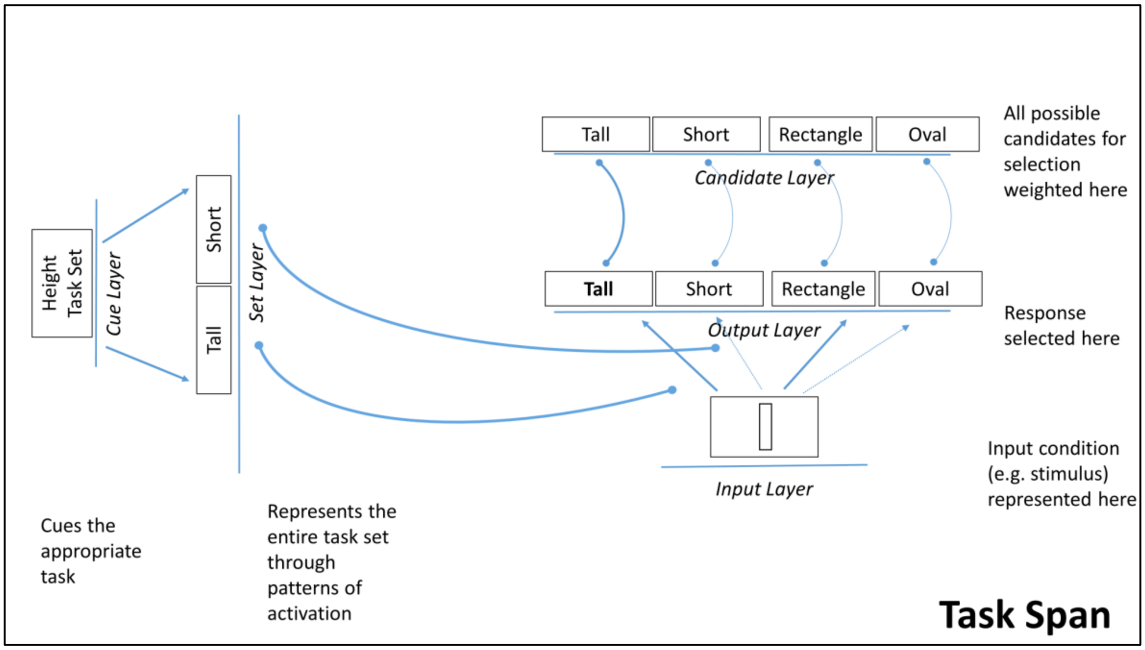
Tall
Short
Rectangle
Oval
All possible candidates for selection weighted here
Candidate Layer
Height Task Set
Cue Layer
Short
Tall
Set Layer
Tall
Short
Rectangle
Oval
Response selected here
Output Layer
Input Layer
Input condition (e.g. stimulus) represented here
Cues the appropriate task
Represents the entire task set through patterns of activation
Task Span

Figure 6. A depiction of how the task span would be implemented in the computational model of Oberauer's work.

In the case of the task span, the cue (the task name) will already have been selected by the declarative subsystem, and the procedural subsystem will be responsible for generating a response based on the presented stimulus. In the set selection module (left), the cue from the output layer of the declarative module identifies the appropriate task set for the given trial. Based on this identification, the elements of the task set are represented in the set layer. In the response selection module (right), the output and candidate layers represent all possible responses that can be made in the experimental context. Once a stimulus comes online, the bindings between certain responses and the conditions associated with that stimulus are strengthened. Based on weighting from the input conditions (e.g. stimulus) and weighting from the set layer, the appropriate response is selected. This process is quite different from the memory span in the current case, in which there are only two possible bindings between the input and output layers among which to select. Considering these two task analyses, it is easy to see that the task span requires procedural elements that the memory span does not, in that the working memory system must process an iteration of the procedural architecture in the task span. An additional change between the memory span and task span in the current experimental context is the increasingly complex binding matrix between stimulus and response options.

Hypothetically, adding a declarative load (e.g. a sequence of colors to be maintained and accessed over the course of the memory span performance) to the memory span would potentially result in two iterations of the declarative module architecture (or two separate processes occurring concurrently). Adding a procedural load (e.g. additional sets of response mappings to use dependent on stimulus characteristics) to the memory span would potentially result in iterations of both the declarative and procedural processes, fundamentally similar to the task span itself. It is on this distinction that the logic of Experiment 1 rests.

In order to fully examine the nature of procedural working memory and to what extent the mechanisms of procedural and declarative working memory are equivalent, the appropriate methodology must first be determined. The task span could be a paradigm that encapsulates the procedural working memory component: in this paradigm participants must retrieve task sets (procedural information) based on stored sequences of task names (declarative information). The related memory span is a good point of comparison for the task span, as it requires the same declarative but not the same procedural information.

Specifically, consider Figure 4. The standard memory span includes strong contributions from declarative working memory in the maintenance of task names. There is almost no procedural contribution. The task span, in contrast, has those same task names, but also has substantial contributions from procedural working memory in the task sets that must be represented and executed. Modifying the memory span in two ways can provide evidence that the task span requires procedural input: adding a declarative working memory load and a procedural working memory load. If adding a distinctly procedural working memory load to the memory span results in performance like the task span, then we have evidence that the difference between these two types of span paradigms is the addition of procedural information. This finding would suggest that the task span, could be a good test of procedural working memory. Including a declarative working memory load condition is critical in that it will allow for the conclusion that changes in performance with the procedural load do not come only from adding complexity to the memory span.

In their past work, Oberauer and colleagues (e.g. Oberauer, 2009; 2010; Oberauer, et al., 2013) have drawn from the findings of the task switching literature in order to come to conclusions about procedural working memory. They have then specifically designed declarative tasks separately that are meant to be similar in structure to task switching paradigms in order to draw conclusions about the declarative subsystem (e.g. Oberauer, et al., 2013). Experiment 1 tested the possibility that the task span can serve as a measure of procedural working memory. If the manipulations are successful in producing a parallel between the task span and the memory span with a procedural load, the primary difference between the memory span and task span conditions will be the presence of additional procedural task execution in the task span. Otherwise, the experimental environments surrounding the declarative and procedural tasks are identical.

Experiment 1 tested the hypothesis that the task span is a task appropriate for testing procedural working memory in Oberauer's model.   To do so, Experiment 1a modified the memory span by adding a declarative working memory load (a sequence of colors that must be matched on every trial throughout the memory span). Experiment 1b modified the memory span by adding a procedural working memory load (extra task sets for recalling the memorized sequences, with each task set used conditionally based on the color of the current stimulus). Both of these loads change participants' performance in the memory span (i.e. slowing). However, only with the procedural load should participants' performance change in a way to align with performance in the un-modified task span—that is, increasing RT with increasing sequence complexity. If adding a procedural load to the memory span results in similar performance to the task span, then we can conclude that the task span is procedurally demanding task above and beyond the memory span. The task span could then be used in later experiments as a measure that incorporates procedural working memory. Critically in Experiment 1, the declarative and procedural loads were equated for complexity, as well as the timing of the interference with memory span performance. That is, they both involved retrieving information learned at the beginning of the experiment on every trial, and updating performance based on the presented stimuli during task recall. Differences in performance between these two conditions were the result of the different nature of the load (declarative or procedural), rather than differences in complexity or other task requirements.

Methods.

Design. Experiments 1a and 1b used within-subjects designs with span paradigm type (three levels: memory span-load, memory span-no load, and task span) as the main independent variable. Sequence complexity and task transition were also manipulated. Experiment 1a involved a declarative working memory load, while Experiment 1b involved a procedural working memory load.

Experiment 1a- Declarative Load.

Participants. Participants included 19 Lehigh University students who participated for partial course credit or payment of $10/hr. All participants had normal or corrected to normal vision.

Materials and procedure. Participants completed three span paradigms in one 90-minute session. The order of the three paradigms, task span, memory span-no load, and memory span with declarative load, were counterbalanced among participants.

Task span. In the task span, participants began by first learning four stimulus-response mappings organized into two tasks: shape and height. They practiced each of the two categorization tasks (tall/short for height, and rectangle/oval for shape) 24 times. Key assignments were counterbalanced among participants, with *s, d, k,* & *l* as possibilities. Assignments were always constrained such that each hand was dedicated to one task only (e.g. *s* & *d* for height and *k* & *l* for shape). Following practice of the basic tasks, participants were instructed on the entire task span procedure, described below, and completed four practice blocks with experimenter support as needed. They then moved to the main phase of the experiment, where they completed 60 study-test blocks. During the study phase, participants were presented with a sequence of six task name cues. The cues were either “height” or “shape”. Each sequence included three instances of each task cue. Under these parameters, there are 20 possible unique sequence orders (presented in table 1).

Sequence	Complementary Sequence	Switch Frequency
hhhsss	ssshhh	1
hssshh	shhhss	2
hhsssh	sshhhs	2
hhshss	sshshh	3
hsshhs	shhssh	3
hshhss	shsshh	3
hhsshs	sshhsh	3
hsshsh	shhshs	4
hshssh	shshhs	4
hshshs	shshsh	5

Table 1. List of all possible sequences used in Experiments 1 and 3.

Each sequence was presented three times. The cues were presented for 500 ms with 500 ms inter-cue intervals. After viewing the entire sequence, there was a 500 ms break screen with the word “TEST” presented, after which participants advanced to the test phase. During the test phase participants made height and shape categorizations on tall and short rectangles and ovals. Participants were instructed to perform all categorizations as quickly and accurately as possible. They were instructed to make their categorizations in accordance with the order of tasks that they had memorized immediately prior. That is, if they memorized “height, height, height, shape, shape, shape” during the study phase, then during the test phase they would categorize the first, second, and third objects to appear as either tall or short, and categorize the fourth, fifth, and sixth objects as either rectangle or oval. Tall objects were 2cm x 5cm, and short objects were 2cm x 3cm. Stimuli were presented in a random order on a light grey background in one of six possible colors: yellow, green, blue, red, pink, and orange. These colors were chosen in order to maintain visual constancy with the declarative load paradigm described below. After participants made one categorization there was a 100 ms inter-stimulus interval before the next object appeared. After all six stimuli were categorized, a rest screen appeared until the participant pressed the space bar to advance to the next block.

Memory span-no load. In the memory span-no load, participants’ visual experience was identical to the task span but their task during the test phase changed. During the study phase they were presented with a sequence, and during the test phase they were presented with tall and short rectangles and ovals in varying colors. However, in the memory span-no load condition participants did not need to perform categorizations on the presented objects, therefore they did not learn stimulus-response mappings for the different stimulus dimensions. Instead, they learned two response keys, one that represented “height”, and one that represented “shape” (key options were either 4 and 7 or 5 and 8 on the number pad). During the test phase, participants simply recalled the memorized sequence from the study phase using those two keys. The characteristics of the object on the screen on any given trial had no bearing on the participant’s performance; instead, they were instructed to use the objects simply as a prompt to recall the next task name in the sequence using their designated height or shape key. While it may seem unusual to present stimuli to the participants that do not influence their performance, this was done in order to maintain visually constant displays (Logan, 2004). All sequence, stimulus, and timing characteristics were identical to the task span. Participants completed four practice blocks with experimenter supervision before beginning 60 blocks of memory span execution.

Memory span with declarative load. The memory span with the declarative load largely matched the basic memory span, with one major addition during the test phase and one additional practice element. These procedures are depicted in figure 7.

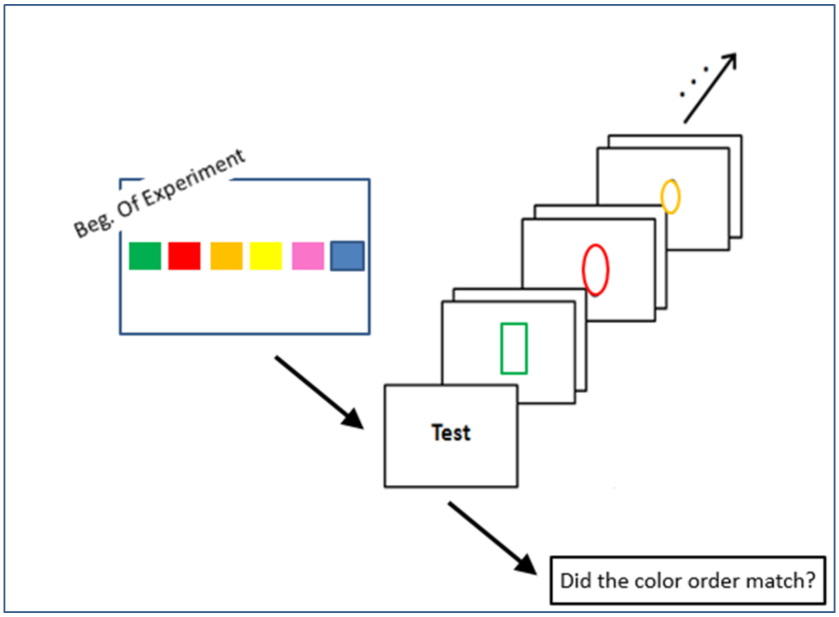

Figure 7. A depiction of procedures used in Experiment 1a. With the declarative working memory load, participants learned a color sequence at the beginning of the memory span administration, and then monitored the colors of stimuli presented in the test phase to determine whether the color order matched their memorized sequence or not.

First, participants learned a color order sequence that they were instructed to memorize and maintain over the course of that entire span session. This learning phase occurred once at the very beginning of the memory span-load implementation. There were four possible color order sequences comprised of the colors red, blue, green, orange, pink, and yellow, but each individual participant learned only one color order. Participants were first given self-paced study time, during which six square color patches were presented on the computer screen with one of the numbers 1-6 under each patch, which reaffirmed the serial order of that color in their color sequence. Once participants indicated they were ready to be tested, one color patch at a time appeared on the screen for 500 ms until all six options were presented. Participants then had to indicate whether the just-presented color order matched the one they had originally memorized by pressing either the *c* or *m* key (counterbalanced). There were 5 tests, and if participants did not get 100% accuracy they were given another opportunity to study the color sequence followed by 5 additional tests. The importance of perfectly memorizing this color order sequence was stressed to participants repeatedly.

After the color order sequence was memorized, participants were instructed on the basic memory span, with one modification. In the test phase, in addition to recalling the memorized task name sequence, participants also needed to monitor the color of the objects appearing while they recalled the height-shape sequence using the 5 and 8 keys. Following their sequence recall, they pressed either the *c* or *m* key to indicate their decision about whether the color sequence matched their memorized order. Participants received feedback about their color order choice only, and were instructed to try to keep their color order accuracy as close to 100% as possible. On 75% of trials, the colors of the objects were presented in the same order as their memorized color sequence. On the remaining 25% of trials the color order did not match. These mismatched blocks were excluded from analysis, as well as matching blocks on which the incorrect decision was made. These stipulations ensure that on all analyzed trials, participants were actively maintaining the declarative load for the entire memory span performance. If mismatch blocks were included, participants could stop monitoring the color order as soon as there was one error, even if that occurred on the second or third serial position. For the memory span with load, each possible sequence was presented 4 times for a total of 80 study-test blocks, to account for the mismatched blocks' exclusion. A maximum of 60 blocks were therefore analyzed in the load condition, just as in the task span and standard memory span, unless there were blocks with inaccurate color order recall.

In this memory load, participants first learned relevant declarative information (the color sequence) at the beginning of the span paradigm. They had to maintain that information over the course of the entire span administration. On any given block, participants had to refer to this memorized information, and on every trial they were required to retrieve the appropriate color based on serial position cues from activated long term memory and update the contents of the region of direct access. At the same time, participants also needed to retrieve the memory span sequence and update that response based on serial position cues.

Experiment 1b- Procedural Load

Participants. Participants included 18 Lehigh University undergraduates who participated for partial course credit or payment of $10/hr., and did not participate in Experiment 1a. All participants had normal or corrected-to-normal vision.

Materials and Procedure. As in Experiment 1a, participants completed three span paradigms in one 90-minute session. The order of the three paradigms, task span, memory span without load, and memory span with procedural load, were counterbalanced among participants.

Task span. The task span in Experiment 1b was the same as Experiment 1a with only minor modifications. Namely, the stimuli presented were either red or blue only. Possible key assignments were *s, f, j,* & *l.* All key assignments were counterbalanced among participants.

Memory span-no load. The memory span-no load condition in Experiment 1b also had only minor changes from that of Experiment 1a. Stimulus colors were limited to only red and blue, and the potential key assignments were either 4 and 7 or 5 and 8.

Memory span with procedural load. To add a procedural load to the memory span, the number of task sets participants needed to learn and implement was increased compared to the standard memory span. In the no-load memory span, participants learned only one task set—a key for height and a key for shape. In the loaded memory span, participants learned two task sets—one set they used when the stimulus on screen was red, and one set they used when the stimulus on screen was blue. These procedures are depicted in figure 8.

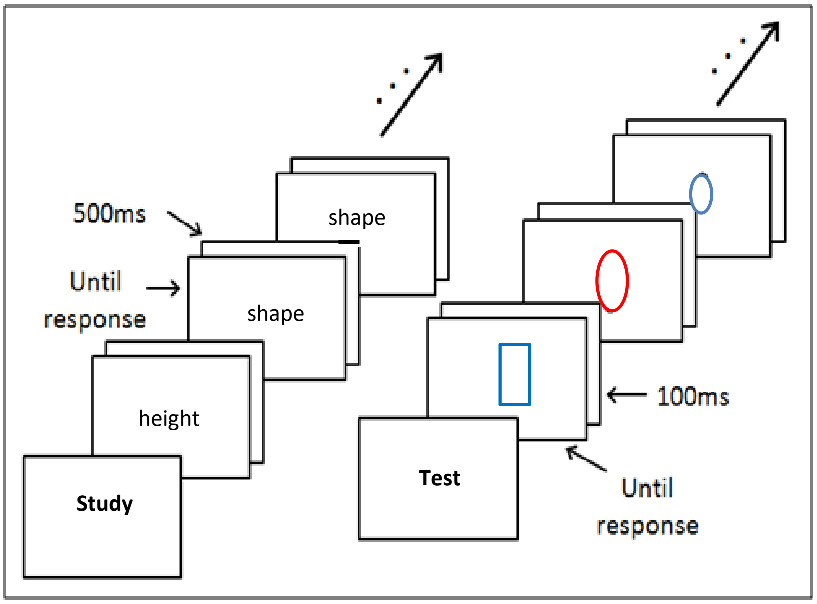

Figure 8. A depiction of the procedures used in Experiment 1b. With the procedural working memory load, participants needed to perform the memory span using two task sets, one each for red and blue objects (four keys total).

Their task of recalling the memorized sequence made up of height and shape cues remained the same—only the way in which they indicated their selections changed. Stimuli were presented in a random order, so participants could not anticipate which set of keys they would be using on any given trial. Instead, they needed to evaluate the color of the stimulus and make their response conditionally based on what was presented. The two task sets were represented by the same key options as in the memory span, 4 and 7 and 5 and 8, but participants needed all four keys in this case. Participants learned one task set for red objects, and one task set for blue objects. Therefore their major task of recalling was the same, but they had to actually pay attention to the color of the object on screen to know with which task set they were meant to respond on that trial (i.e. the procedure for responding varied from trial to trial).

In this memory load, participants first learned relevant procedural information (the task sets) at the beginning of the span paradigm. They had to maintain that information over the course of the entire memory span session. On every trial participants were required to retrieve these contents from activated long term memory into the bridge for selection and implementation.

Results.

Experiment 1a- Declarative Load. Data were first analyzed on a participant level to evaluate basic accuracy and memory performance. Participants were excluded from further analysis if their task accuracy in the task span (i.e. their object categorization performance) was lower than 80% or if their task memory was lower than 75%. Three participants were excluded from analysis based on low accuracy (*n* = 2) or failure to complete all three span paradigms (*n*=1). Therefore, data from 16 participants were included in final analyses. For all variables, comparisons were made between the task span, the memory span- declarative load, and the memory span-no load. RT data were trimmed to exclude trials for which RT was greater than 3500 ms or lower than 150 ms or errors were made. Accuracy data for the task span can be found in Appendix A.

Target RT. RT can be investigated as a function of either local or global measures. This distinction reflects the ways in which trials are related to one another. The local measure of transition involves how two neighboring trials are related, either as a switch or a repetition of the task from trial n-1 to trial n. The global measure of complexity involves the relationship among all trials in a sequence. Participants' RTs to targets were first analyzed on a local level, investigating the difference between switching and repeating tasks. Then a more global measure of sequence complexity, switch frequency, was considered. This measure was the overall number of switches in a sequence.

The effects of span type (task span, memory span-load, and memory span-no load) and local transition (repetition vs. switch) on target RT were examined with a 3 x 2 repeated measures ANOVA. These data are presented in figure 9. There were significant main effects of span type, $F(2, 30) = 205.17$, $p < .001$, $\eta^2_{\text{partial}} = .932$, transition, $F(1, 15) = 41.32$, $p < .001$, $\eta^2_{\text{partial}} = .734$, and a significant interaction between these two factors, $F(2, 30) = 46.47$, $p < .001$, $\eta^2_{\text{partial}} = .756$. RTs were fastest in the memory span-no load condition and slowest in the task span condition, but pairwise comparisons using

Fisher's LSD[2] reveal that all three span conditions were significantly different from each other (*p*s < .001). Generally repetitions were faster than switches, but as the significant interaction demonstrates, switch costs were much larger for the task span condition (*M* = 267 ms—difference score) than they were for either the memory span-no load (*M* = 17 ms) or memory span-load conditions (*M* = 27 ms). Follow up simple effects *t*-tests comparing repetition trials to switch trials (switch costs) revealed that switch costs were not significant for the memory span-no load condition, *t*(15) = -1.69, *p* = .111, or the memory span-load condition, *t*(15) = -1.22, *p* = .242, but switch costs were significant for the task span, *t*(15) = -8.23, *p* < .001.

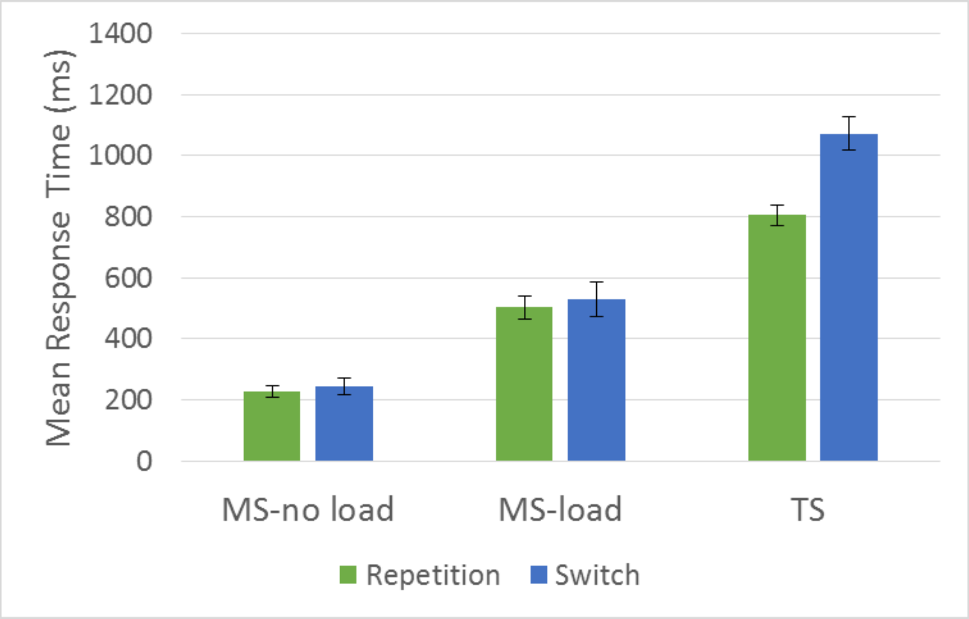

Figure 9. Mean RT to targets in Experiments 1a as a function of transition for the memory span (MS)-no load, memory span-load, and task span (TS) conditions. [3]

[2] No corrections were made for multiple comparisons here or elsewhere because there were small numbers of comparisons selected for analysis *a priori.*

[3] Here and throughout error bars represent standard error.

A critical question for Experiment 1 is how the declarative and procedural working memory loads impact the memory span performance. The hypotheses for this experiment were that the declarative load should not result in the memory span performance matching task span performance, but the procedural load should result in that pattern of behavior. As such, follow-up tests were conducted for all relevant interactions (described below) to determine if there were significant differences between the memory span-load conditions and the task span. The prediction is that there should be differences for the declarative load (Exp. 1a), but no such differences should exist for the procedural load (Exp. 1b). This would support the idea that the task span requires procedural input in a way that the memory span does not. In order to test the first half of this prediction (the declarative load portion) follow-up ANOVAs directly comparing the memory span-load condition to the memory span-no load and task span conditions were conducted. The 2 (memory span-load and memory span-no load) x 2 (repetitions and switches) repeated measures ANOVA indicated that the interaction between load and transition on target RT was not significant, $F < 1$. However the 2 (memory span-load and task span) x 2 (repetitions and switches) repeated measures ANOVA indicated a significant interaction between these two factors on target RT $F(1, 15) = 55.12$, $p < .001$, $\eta^2_{\text{partial}} = .786$. This pattern of data is clearly indicated in figure 9 when comparing the pattern of switch costs among the three conditions; memory span-load is not different from the memory span –no load, whereas the task span is quite different from the memory span-no load.

The effects of span type (task span, memory span-load, and memory span-no load) and switch frequency (five levels- 1, 2, 3, 4, & 5) on RT were examined with a 3 x 5 repeated-measures ANOVA. These data are presented in figure 10. The effect of span type was significant, *F*(2, 30) = 153.50, *p* < .001, η^2_{partial} = .911, as was the effect of switch frequency, *F*(4, 60) = 31.17, *p* < .001, η^2_{partial} = .675, and the interaction between span type and switch frequency, *F*(8, 120) = 16.64, *p* < .001, η^2_{partial} = .526. Generally RTs increased as the switch frequency increased, but this effect was qualified by the significant interaction such that RTs were most influenced by switch frequency in the task span. The same follow-up comparisons as described above were conducted for switch frequency as well. These repeated measures ANOVAs allowed for a comparison between the memory span- load and the two other conditions independently. These follow up tests indicate that the interaction between load and switch frequency was significant as evidenced by the 2 (memory span-load and memory span-no load) x 5 (switch frequency) repeated measures ANOVA, *F*(4, 60) = 5.09, *p* = .039, η^2_{partial} = .253. The interaction between the memory span-load and task span conditions with switch frequency was also significant as evidenced by the 2 (memory span-load and task span) x 5 repeated measures ANOVA, *F*(4, 60) = 18.54, *p* < .001, η^2_{partial} = .553. Based on these results, target RT is influenced by complexity in the declarative load condition in a different way than when there is no load. However, critically, the effect size of the interaction for the task span/ memory span-load condition was much larger than that of the memory span-no load/memory span-load interaction. Therefore, the effect of complexity on RT is more substantial in the task span than in the memory span-load condition. Although the declarative memory load does alter the pattern of participants' performance, it does not alter it in a way that is consistent with the task span. This finding is consistent with the prediction that the memory span with declarative load should not match performance in the task span.

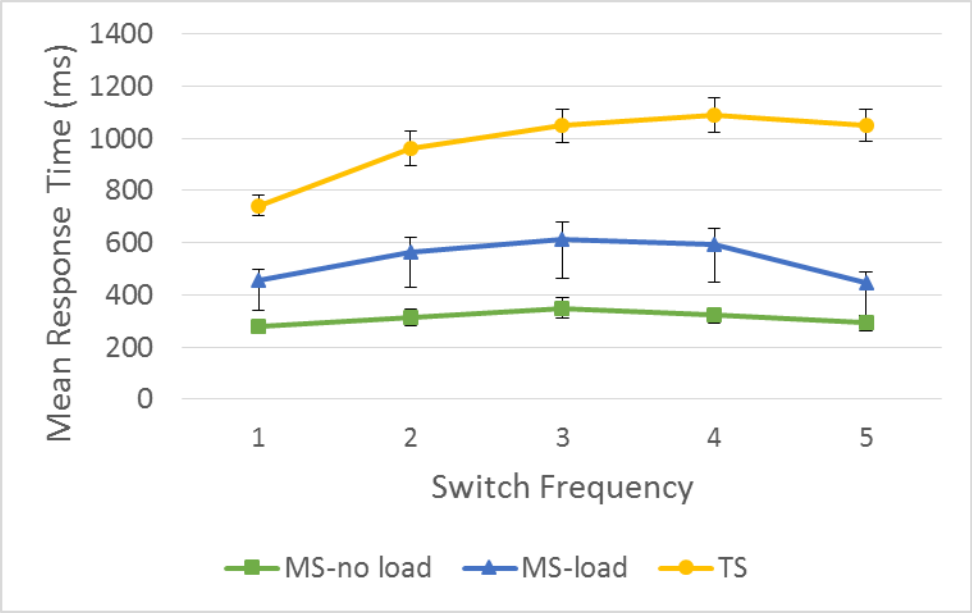

Figure 10. Mean RT to targets in Experiment 1a as a function of switch frequency for the memory span (MS)-no load, MS-load, and task span (TS) conditions.

Overall these target RT tests, particularly the effects of transition, suggest the interpretation that adding a declarative working memory load to the memory span does not yield an equivalent condition to the task span. While the pattern of performance indicates that the declarative load alters memory span performance, it does not do so in a way to match performance in the task span.

List memory. List memory (participants' perfect memory for entire sequences) was analyzed only as a function of switch frequency as local transitions are not relevant for this measure. The effects of span type (task, memory-load, and memory-no load) and switch frequency (five levels- 1, 2, 3, 4, & 5) on list memory were examined with a 3 x 5 repeated measures ANOVA. These data are presented in figure 11. The effect of span type was not significant, *F*(2, 30) = 2.29, *p* = .151, η^2_{partial} = .132. The effect of switch frequency was significant, *F*(4, 60) = 27.67 *p* < .001, η^2_{partial} = .648, as participants' list memory was generally best for sequences with very few or the maximum amount of switching. The interaction between these two factors was not significant, *F*(8, 120) = 1.78, *p* = .201.

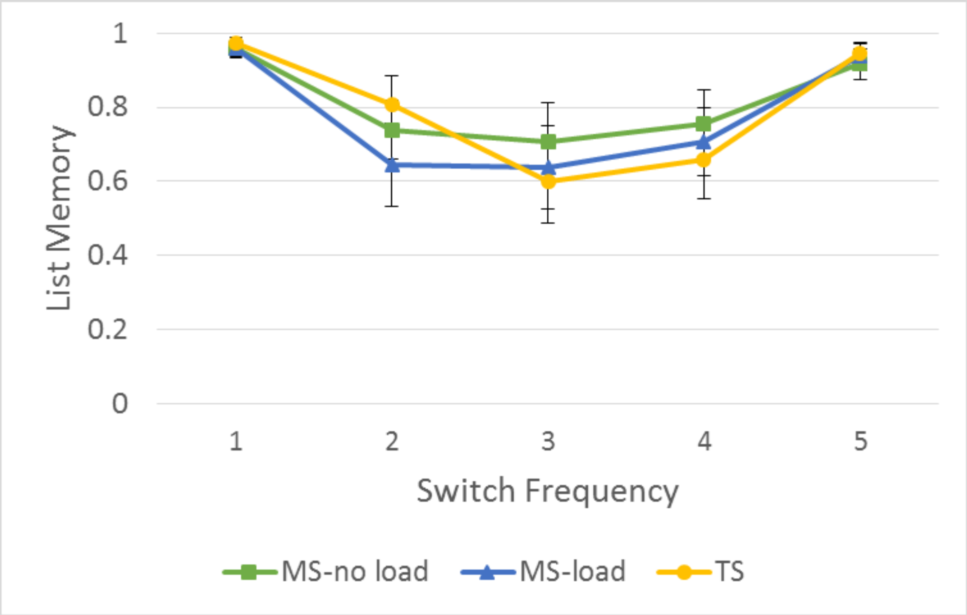

Figure 11. Mean list memory for sequences in Experiment 1a as a function switch frequency for the memory span (MS)-no load, (MS)-load, and task span (TS) conditions.

List memory data were less supportive of the main hypothesis than target RT. When adding a declarative working memory task to the memory span, there were not significant main effects of span type or interactions between span type and complexity.

Participants' memory for sequences declined with complexity, but not in a different way depending on the type of span they were performing.

Experiment 1b- Procedural Load. As in Experiment 1a, data were first analyzed on a participant level to evaluate basic accuracy and memory performance. Participants were excluded from further analyses if their task accuracy in the task span (object categorization performance) was lower than 80% or if their task memory was lower than 75%. Two participants were excluded from analysis based on low accuracy in the task span. Therefore, data from 16 participants were included in final analyses. RT data were trimmed to exclude trials for which RT was greater than 3500 ms or lower than 150 ms or errors were made.

Target RT. As in Experiment 1a, participants' RTs to targets were first analyzed on a local level, investigating the difference between switching and repeating tasks. The global measure of overall sequence complexity was then considered.

The effects of span type (task span, memory span-load, and memory span-no load) and transition (repetition vs. switch) on target RT were examined with a 3 x 2 repeated measures ANOVA. These data are presented in figure 12. There were significant main effects of span type, $F(2, 30) = 293.79$, $p < .001$, $\eta^2_{\text{partial}} = .951$, as participants were fastest in the memory span-no load condition. Pairwise comparisons (Fisher's LSD) reveal that the memory span no-load condition was significantly faster than both the memory span-load and task span conditions, *p*s <.001, but the load and task span conditions were not significantly different $p = .641$. There was also a significant main effect of transition, $F(1, 15) = 71.41$, $p < .001$, $\eta^2_{\text{partial}} = .826$, as repetition trials were faster than switch trials. There was also a significant interaction between these two factors, *F*(2, 30) = 28.31, *p* < .001, η^2_{partial} = .654. The cost of switching was evident for the task span (*M* = 337 ms)   and memory span-load conditions (*M* = 292 ms), but was much less substantial for memory span- no load (*M* = 34 ms). Follow up simple effects *t*-tests comparing repetition trials to switch trials (switch costs) further investigated this interpretation, and revealed that switch costs were significant for the memory span-no load condition, *t*(15) = -2.47, *p* = .026, the memory span-load condition, *t*(15) = -6.74, *p* < .001, and for the task span, *t*(15) = -8.02, *p* < .001. Despite all comparisons being significant, as evidenced by figure 12, obviously the switch costs were much lower numerically in the memory span-no load condition than the other two.

As in Experiment 1a, follow up repeated measures ANOVAs were conducted to make comparisons between the memory span-load condition and the other two conditions. Examining the interaction between the memory span-load and memory span-no load conditions with a 2 (memory span-load and memory span-no load) x 2 (repetitions and switches) repeated measures ANVOA yielded a significant effect, *F*(1, 15) =37.26, *p* < .001, η^2_{partial} = .713. The 2 (memory span-load and task span) x 2 (repetitions and switches) repeated measures ANVOA for memory span-load and the task span was not significant, *F* < 1.

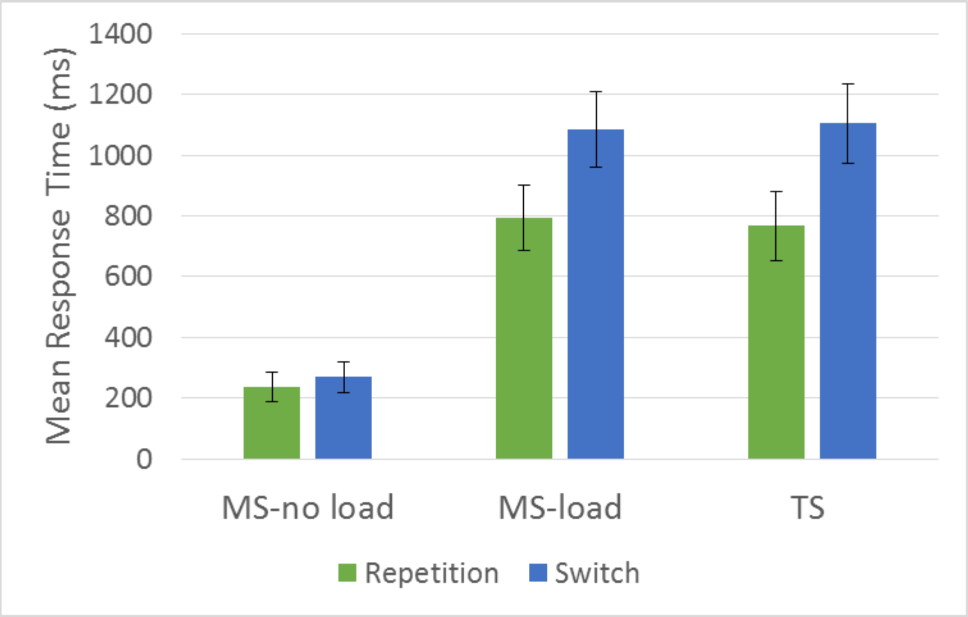

Figure 12. Mean RT to targets in Experiment 1b as a function of transition for the memory span (MS)-no load, (MS)-load, and task span (TS) conditions.

The effects of span type (task, memory-load, and memory-no load) and switch frequency (five levels- 1, 2, 3, 4, & 5) on target RT were examined with a 3 x 5 repeated measures ANOVA. These data are presented in figure 13. The effect of span type was significant, *F*(2, 30) = 288.74 *p* < .001, $\eta^2_{partial}$ = .951, as was the effect of switch frequency, *F*(4, 60) = 60.38, *p* < .001, $\eta^2_{partial}$ = .801 and the interaction between span type and switch frequency, *F*(8, 120) = 17.01, *p* < .001, $\eta^2_{partial}$ = .531. Sequences with more switches generally resulted in increased RTs, but the 5-switch condition did show a substantial decrease in RT compared to what would be expected based on a linear trend. The effect of switch frequency was much stronger for the memory span-load and the task span conditions than it was for the memory span-no load condition. Follow up tests comparing subsets of the three conditions were again conducted, as in Experiment 1a. A 2 (memory span-load and memory span-no load) x 5 (switch frequency) repeated measures ANVOA indicated that the interaction between the memory span-load and memory span-no load conditions was significant, *F*(4, 60) = 16.01, *p* < .001, η^2_{partial} = .516. A 2 (memory span-load and task span) x 5 (switch frequency) repeated measures ANVOA yielded a non-significant effect on target RT, *F*(4, 60) = 3.29, *p* = .09. Adding a procedural load to the memory span made participants' RT performance look like RTs in the task span, as shown in figure 13.

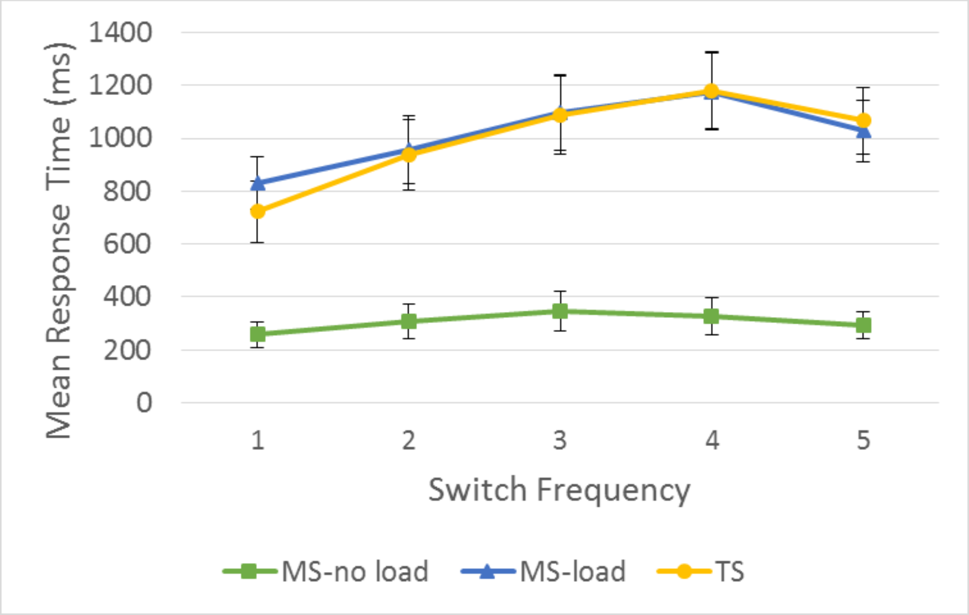

Figure 13. Mean RT to targets in Experiment 1b as a function of switch frequency for the memory span (MS)-no load, MS-load, and task span (TS) conditions.

List Memory. As in Experiment 1a, list memory was analyzed only as a function of switch frequency as local transitions are not relevant for this measure considering that it is based on an entire-sequence metric. The effects of span type (task, memory-load, and memory-no load) and switch frequency (five levels- 1, 2, 3, 4, & 5) on list memory were examined with a 3 x 5 repeated measures ANOVA. These data are presented in figure 14.The effect of span type was significant, *F*(2, 30) = 5.68 *p* = .031, $\eta^2_{partial}$ = .275. Participants' list memory was best for the memory span-no load condition, and worst for the memory span with the procedural load. The effect of switch frequency was also significant, *F*(4, 60) = 15.14 *p* = .001, $\eta^2_{partial}$ = .502, as participants' list memory was generally best for sequences with very few or the maximum amount of switching. The interaction between these two factors was not significant, *F*(8, 120) = 2.54 *p* = .132.

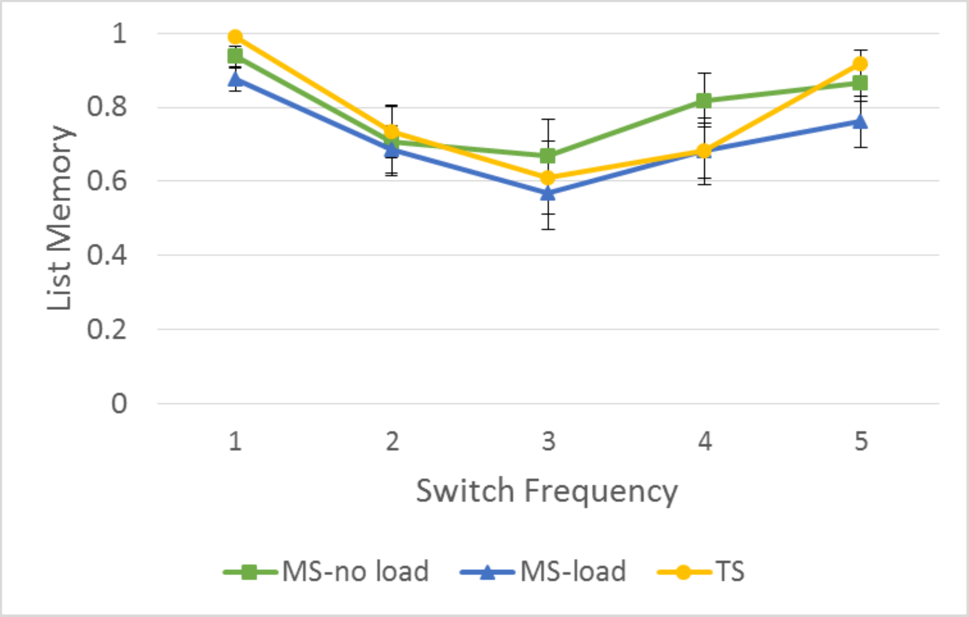

Figure 14. Mean list memory for sequences in Experiment 1b as a function switch frequency for the memory span (MS)-no load, (MS)-load, and task span (TS) conditions.

As seen in Experiment 1a, list memory data were less supportive of the main hypothesis than target RT as to the effect of adding a procedural working memory load to the memory span. Participants' memory for sequences declined when a memory load was added, but not to the point of being matched with task span performance as was seen in the target RT data.

Between-experiment comparisons. Finally, the effects of the declarative and procedural load conditions were directly compared between Experiment 1a and 1b to confirm that the two different types of load had clearly different effects on behavior in the memory span.

Target RT. First considering RT, a 2 x 2 mixed factors ANOVA was conducted with load type (declarative or procedural-between subjects) and transition (repetition or switch- within subjects) as factors. See figure 9 (1a) and figure 12 (1b). There was a significant effect of load type, $F(1, 30) = 39.94$, $p < .001$, $\eta^2_{\text{partial}} = .571$. RT was longer for the procedural load condition (M = 931 ms) than the declarative load condition (M = 515 ms). There was also a significant effect of transition, $F(1, 30) = 43.18$, $p < .001$, $\eta^2_{\text{partial}} =$ .590, with longer RTs for switch trials (M = 806 ms) than repetition trials (M = 642 ms). There was also a significant interaction between these factors, $F(1, 40) = 30.03$, $p < .001$, $\eta^2_{\text{partial}} = .500$, as the effect of transition was much larger for the procedural load (switch cost M = 292 ms) than the declarative load (switch cost M = 27 ms).

Considering the effects of switch frequency, a 2 x 5 mixed factors ANOVA with load type and switch frequency (1, 2, 3, 4, or 5 switches, within subjects) as factors indicated a significant effect of load type, $F(1, 30) = 58.89$, $p < .001$, $\eta^2_{\text{partial}} = .663$. These data are presented in figure 10 (1a) and figure 13 (1b). RTs were longer with the procedural load than the declarative load. There was also a significant effect of switch frequency, $F(4, 120) = 32.07$, $p < .001$, $\eta^2_{\text{partial}} = .517$. Participants RTs slowed as the number of switches increased, but sped up for the 5-switch condition. There was also an interaction between these two factors, *F*(4, 120) = 8.44, *p* = .007, η^2_{partial} = .219. The effect of switch frequency was larger for the procedural load than for the declarative load.

List Memory. Just as list memory was less supportive of the main hypothesis for Experiments 1a and 1b individually, it was also less supportive with regard to these comparisons. A 2 x 5 mixed factors ANOVA with load type (procedural and declarative, between subjects) and switch frequency (1, 2, 3, 4, & 5, within subjects) as factors yielded a non-significant effect of load type *F* < 1. These data are presented in figure 11 (1a) and figure 14 (1b). There was a significant effect of switch frequency, *F*(4, 120) = 20.46, *p* < .001, η^2_{partial} = .406, as list memory declined for increased switches except for the highest switch values. There was no significant interaction, *F*(4, 120) = 1.94, *p* = .174.

Discussion.

The results of Experiment 1 suggest that the task span can be used as a test of procedural working memory. In particular, RT data reveal significant differences between declarative and procedural loads on the memory span performance. Complexity (switch frequency) changed RT performance in the memory span-declarative load condition, but not to the same extent as the memory span-procedural load condition. In the latter case, RT performance became nearly identical to RTs in the task span. Similarly, adding a declarative load to the memory span did not yield a switch cost whereas adding a procedural load to the memory span did.

Task analysis based on computational model. In order to understand the implications of the declarative and procedural loads in Experiment 1, the current procedures and data should be examined in the context of a task analysis guided by Oberauer's computational model as outlined in the introduction to Experiment 1. These considerations support the idea that the task span can be used as a paradigm to test the mechanisms of procedural working memory. For example, when completing the memory span, the working memory system could start by running through an iteration of the declarative working memory architecture as described in figure 5. In doing so, the set layer represents the entire declarative sequence, and based on serial position cues (input layer), the correct task is selected (output layer) into the focus of attention on each trial. When the memory span is modified with a declarative working memory load, the system may run through this declarative architecture twice; once to select the correct item in the presented span, and again to match the current stimulus color with the stored color order sequence from the beginning of the experiment. In this case, there is an increase in RT, because of the additional demand on the declarative subsystem to maintain and update the color order. However, while this additional load puts demand on working memory, it puts extra load only the declarative portion, not procedural. No extra procedural responses are required during the actual execution of the memory span, only monitoring of declarative information. The response requirements from the declarative load occur after the completion of the memory span.

When considering the addition of the procedural load to the memory span, the procedure begins the same way. The declarative information (tasks) is selected through an iteration of the declarative architecture, and either the height or shape task is selected. Then the process is completed by an iteration of the procedural architecture. That is, once the correct task is selected, it serves as the cue for retrieving the correct task set in the set selection module. Once the appropriate task set is recruited into the set layer, the correct response can be made. This response is made conditionally based on the stimulus that appears on screen through the weighting of input-output bindings from the set layer. So in the case of the procedural load, the nature of the additional information is quite different from the information typically needed in the memory span, or that is needed in the declarative load manipulation. Instead of accessing the declarative color order sequence on each trial, the participant must dynamically access the correct task set, dependent on what information is presented on screen. This dynamic activation of different sets depending on incoming cues is the same process that occurs in the task span—the declarative architecture selects the appropriate task and then the procedural architecture selects the response. Based on this task analysis, the reader can see how the procedural load results in performance like the task span.

The differences in task performance in the two load conditions was apparent in the analysis of complexity. The results of Experiment 1 reveal that RT does not change dramatically with increasing complexity in the declarative load condition, whereas complexity is very influential in the procedural load condition. This difference can be explained by the nature of bindings in the region of direct access and the bridge, respectively. Oberauer and colleagues (2013) suggest that the region of direct access (and presumably the bridge by extension based on the functional analogy claim), is comprised of the representations that are currently active in the input and output layers. For the intermediary stores, these representations include all possible input-output bindings, not just those that have been actually selected. In the cases of the memory span and memory span with declarative load in the current context, there is a very small amount of competition between bindings in these two layers. There are only two pieces of information from which a selection can be made in the memory span, height and shape. These items are selected variably based on serial position. In the declarative load condition, there are six pieces of information, the color names, but there is no variation in how they are mapped to serial position over the course of the experiment. In both of these cases- the memory span and the declarative load, the set layer is more complex than the task span, with six items to be remembered in sequence. As for the focus of attention, there is very low competition between the input and output layers in both cases. This is because these bindings do not change dynamically depending on the stimulus characteristics. With a less complex binding matrix between the input and output layers, switching between items is not costly. Since single switches are not costly, there is no additive effect of switch costs over more complex sequences.

The nature of the procedural load is very different, and consequently performance is quite different as well. In the task span and the memory span with procedural load, participants must engage task sets. There is a determined set of responses used for each task, or each colored stimulus, and the bindings that represent the different aspects of the task set (e.g. binding tall to the *a* key and rectangle to the *k* key) overlap and compete with each other. Complexity matters in both of these cases (the task span and the memory span with procedural load) because of the nature of the bindings in the procedural architecture. There are four possible responses in either case, and critically, they are organized into task sets. Task sets give rise to switch costs (Dreisbach, Goschke, & Haider, 2007), and therefore subsequent slowing effects of complexity. That is, when each switch results in a cost, having a sequence with more switches will aggregate to larger increases in RTs, which is generally the pattern we see in Experiment 1.

Previous task span work can also inform the discussion of complexity. Logan (2006) found that when added together, RTs for the memory span plus RTs for completing a simple single task sequence (i.e. height, height, height, height, with no memory requirements or switching) were less than RTs for the task span. Logan interpreted this result as evidence that the task span requires some additional process above and beyond just recalling the sequence and performing the task. One likely candidate for this additional processing is the time necessary to retrieve task sets from activated long term memory into the bridge. In the single task sequence, the same task set is always active, and only the response focus needs to be updated. In the task span, the task set can change on every trial. Additional time is also required in the task span in that the cue layer must be updated based on the output of the declarative architecture. In the memory span, there are no task sets, only the memorized sequence. In the single task condition, there is only one task set, and it only needs to be implemented on any given trial, not retrieved. In the task span, there are multiple task sets, and they need to be retrieved conditionally based on the memorized sequence. There is also potentially lingering activation from past trials to be overcome, which results in the increase in RT for the task span compared to the sum of the memory span and single task performance. Engaging tasks sets is clearly a very different cognitive process from simply recalling a task name, which results in the differing effects of complexity.

To summarize, this analysis of the task span and memory span with and without declarative or procedural loads and how they could potentially be implemented in the context of Oberauer's computational model is completely in line with the results of the current experiment. The memory span without load results in quick RTs. Adding a declarative load increases RT (another iteration of the declarative architecture), but is not largely influenced by complexity. Adding a procedural load increases RT (additional iteration of procedural architecture) and is influenced by complexity because of the competition between task sets. The task span (additional iteration of procedural architecture) also results in increased RT and a strong effect of complexity because of competition between task sets.

Sequential or parallel selection? One question that arises from Experiments 1a and 1b that is still unanswered in the literature is whether declarative and procedural information are processed sequentially, or whether there is some degree of overlap in the selection timeline in situations like the task span[4]. The question of whether information processing occurs in a sequential or parallel fashion is one that has been asked in a large variety of domains and with many possible approaches (e.g. see Logan,

[4] There is a large literature on serial memory processing independent of the discussion of the task span. See Hurlstone, Hitch, & Baddeley (2014) for a recent review.

2002). Understanding whether this processing occurs sequentially or in parallel is important because it is informative about the way processing occurs in the task span, and what limitations might exist for other cognitive processes. For example, if information must be processed sequentially in the task span (declarative task name selected followed by procedural output being made), there is smaller processing capacity overall than if it can be handled in parallel—that is, with parallel selection, there could be a division of labor between the declarative and procedural subsystems that allows for more efficient processing. These limitations can explain boundary conditions for cognitive processing more generally.

In the case of Experiment 1 and the memory span and task span, information processing may be happening in two ways. For one, it is possible that for the task span, the procedural information (response selection) is not processed at all until the appropriate declarative task name is determined which would be evidence of sequential processing. Alternatively, information processing could be occurring in a cascading fashion (e.g. McClelland, 1979), with information from the memory span impacting processing in the task span as it is established. In this second case, evidence for potential task categorizations could begin to accumulate before the actual task itself has been selected meaning that some processing occurs in parallel. For example, the stimulus, perhaps a tall rectangle, may get no processing at all until “height” has been determined as the appropriate task (first option), or evidence may begin accumulating for the potentially appropriate “tall” and “rectangle” responses before “height” has even been selected (second option). The procedural subsystem would then actually execute a stimulus response once the declarative iteration was completed and the task had fully been selected (De Jong, 2000). The current data do not provide a definitive answer to this question.

Considering the sequential or parallel processing of declarative and procedural information also brings up a question of how these two subsystems interact. Oberauer and his colleagues are surprisingly quiet on this topic of communication between the declarative and procedural subsystems. There is mention of how declarative and procedural information are mixed throughout long term memory, but no published discussion of how separation is achieved in the intermediary store while communication is still possible in order to perform tasks on items for example (Oberauer et al., 2013). There must be some interaction, given that we can carry out procedures on objects, but how this interaction occurs is a question open for future research and theorizing. This question will be revisited in the general discussion.

Nature of memory for sequences. Participants' perfect recall of sequences, or their list memory, did not provide as clear a story in support of the main hypothesis as did the RT data. For Experiment 1a with the declarative load, the effect of span type on memory was non-significant, as was the interaction between span type and complexity. Regardless of whether they were performing the memory span, the task span, or the memory span with the color monitoring task, participants' memory was about the same. For Experiment 1b with the procedural load, the effects of span type and complexity were significant, but there was no interaction between these two factors. Most critically, the between-experiments comparison of list memory for the memory span with the declarative and procedural loads was not significant. This finding suggests that the significant effects in Experiment 1b came from changes in the comparison conditions of the task span and standard memory span, not because of the nature of the procedural load itself. That is, the memory span and the task span showed different patterns for Experiments 1a and 1b. These effects made list memory appear to be different for the procedural and declarative load conditions despite not actually being different at all.

These list memory data are consistent with the findings of Logan (2004). In that work participants' memory for complete sequences was the same for the memory span and the task span (in this case, recall and perform conditions). The memory data match was maintained even in sequences with more switches. Logan talked about these data as being consistent with the idea that there is no tradeoff between processing and storage. The current experiment supports this claim; even when sequences are more complex, memory performance does not significantly decline when tasks must be carried out rather than just recalled.

Examining these data begs the question, why might the two memory loads have such differential effects on participants' RTs but not their list memory? It may be because of the nature of the memory measure. List memory requires successful task memory for all six items in a sequence. Thinking back to the previous discussion of the computational model, a fundamental step in either the task span or the memory span with either of the two loads is an iteration of the declarative architecture to select the appropriate task to be either recalled or performed. As long as this iteration occurs correctly for all six tasks, the participants' list memory will be perfect—they will retrieve the correct sequence, which is critical for successful execution. Since memory is primarily dependent on this declarative retrieval in the set selection module and not the reconciliation of competition in the binding matrix of the item or response module, the load, regardless of type, does not interfere with the accuracy of this process. The bulk of the work that supports list memory occurs separately from the part of the process influenced by the load (the actual item or response selection), leading to the observed results. The load does, however, interfere with the *time* taken to execute the full span recall and implementation, with list memory relatively untouched after load.

Why does complexity influence list memory at all? Logan (2004) found that in the task span, sequences are retrieved from long term memory in chunks. This finding comes from observation of a scalloped data pattern in RTs for structured sequences. As mentioned before, RTs for the first item in a logical chunk (**A**BC**A**BC**A**BC) were slowed compared to later items in that chunk. Perhaps complexity influences list memory because of the process implemented when retrieving these chunks. Representations in the set layer are brought into the intermediary store through activations from the cue layer. These patterns of activation are distributed and interconnected over the entire set, allowing for more than one memory set to be maintained at any given time (Oberauer, 2013; Oberauer, et al., 2013). This retrieval process could be more accurate when recalling sequences with more structure because they are recalled into the region of direct access more easily. The effect of complexity does not vary by load, because of the aforementioned proposal that the declarative sequence is retrieved separately from task implementation.

In Experiment 1, the declarative working memory subsystem contributes to successful implementation of both the memory and task span while the procedural subsystem largely contributes only to successful implementation of the task span. For the memory span-no load and the memory span with declarative load, there is limited contribution of procedural working memory. However, for the task span and the memory span with procedural load, there is a large contribution from the procedural subsystem. Considering the dependent measures, list memory does not differ between the span conditions because it relies only on declarative working memory, which contributes to all conditions. However, target RT does differ between the span conditions because it draws on procedural working memory, which is not recruited in the same way for all conditions. List memory is not influenced by load, but is by complexity, whereas target RT is influenced in an interactive way with load type and complexity. If we think about list memory as a measure of declarative processes and RT as a measure of procedural processes, and they are influenced so separately, this provides some evidence for independence of the two working memory subsystems. If declarative and procedural information were processed together by the same resource, we would expect them to be influenced by complexity in the same way for both dependent variables. That is not the case in the current experiment—we see very different patterns among the two types of load, as well as between the two main dependent variables. These findings could support an interpretation of independent mechanisms for declarative and procedural working memory. This conclusion also serves to extend the previously described discussion of list memory in Logan's (2004) work. Not only is there no tradeoff between processing and storage in the task span, these two components of task span performance might be separable.

Interim conclusions. Experiment 1 shows that the task span has procedural working memory requirements above and beyond the memory span. Although there is really no such thing as a purely declarative or purely procedural task, the task span has certainly been shown to have procedural elements. Establishing the task span as a procedural working memory task appropriate for testing Oberauer's model is essential for the rest of the dissertation, as the task span is utilized as such in the following experiments. More broadly, Experiment 1 is also a critical methodological contribution of this dissertation to the greater working memory literature, as this paradigm has not been previously connected to Oberauer's model. Based on these data from Experiment 1, there is evidence for independence of the two subsystems, based on the different effects of load on RT and list memory.

Experiment 2: Information Reinstatement

Experiment 2 seeks to address the question of how activation of different content in the bridge and region of direct access influences performance. Past research using a declarative retro-cuing task has suggested that recently used information in the region of direct access maintains a "privileged" state of activation (Rerko & Oberauer, 2013) resulting in better performance compared to other types of content. In that research, the authors contrasted accuracy for items that were either recently focused, recently defocused, or recently unfocused. Specifically, they presented participants with a memory array of colored dots in a circle, and then removed this array from the computer screen for a retention interval, during which they sometimes cued one or more of the locations from the circle. Participants then had to determine if a probe either matched or mismatched the most recently cued location's studied target, and the cued item was sometimes valid and sometimes invalid as to the actual probe presented. There were varying numbers of sequential cues presented—one, two, or three cues, and participants did not know on any trial how many cues there would be. This procedure is referred to as the retro-cuing paradigm, as the presented cue is supposed to be activating information viewed in the past and therefore stored in working memory. The critical trials in this experiment were the three-cue trials, during which items could be cued in a CBA or ABA pattern (among others). Focused items were those to which attention is directed after encoding and the item remains cued until the probe appears (i.e. a BAA cuing sequence). Unfocused items are those that have not been previously cued and therefore attention is not directed to them after encoding until the probe test (i.e. a *CBA* cuing sequence). Defocused items are those that were previously cued and then attention was directed away from them before returning (i.e. an *ABA* cuing sequence). Rerko and Oberauer (2013) found that when items were defocused and then returned to the focus of attention, accuracy was improved and RT was decreased compared to items that were not previously focused, even when that focus did not require an actual task response as the participants only responded to a probe that followed the final cue. They assert that this defocused information enjoys a privileged state and is strengthened.

However, this finding from a declarative working memory task is inconsistent with a long standing effect observed in task switching: backward inhibition or lag 2 inhibition (Mayr & Keele, 2000). Whereas the retro-cuing paradigm largely depends on contributions from declarative working memory, the task switching paradigm is largely dependent on procedural working memory. In the case described below, the implementation of the task switching paradigm is quite similar to that of retro-cuing. In task switching, one potential theoretical explanation of how individuals switch between multiple tasks is that previously relevant task sets and response mappings are inhibited in order for a new task to be successfully implemented (Mayr & Keele, 2000, Schneider & Verbruggen, 2008; cf. Lien, Ruthruff, & Kuhns, 2006). This mechanism is termed “backward inhibition” (Mayr & Keele, 2000), and evidence from a three-task multitasking environment supports this hypothesis. When switching between three possible tasks, participants were slower to perform a recently abandoned task (one that was performed on trial *n – 2)* than they were to perform a new task that had not been performed on trial *n – 2* (Mayr & Keele, 2000). For example, performance was longer for the second instance of task *A* in a sequence such as *ABA* than for a sequence *CBA,* taken as evidence that the participant needed to overcome inhibition of task *A* when it had recently been switched away from. This inhibition mechanism arises from the demands of dual goals during task switching, one of flexibility that allows for the actual switching between established tasks, and another of stability that prevents one from moving away from a task that has not been completed or towards a task that is inappropriate at a particular moment (Goschke, 2000; Hübner, Dreisbach, Haider, & Kluwe, 2003). This inhibitory mechanism is critical for successful task switching because it facilitates delineation of tasks when performing more than one at a time, but is simultaneously costly when considering the additional time needed for reestablishing previously relevant task sets.

Clear parallels exist between this instance of an environment with three task sets in task switching (dependent more on procedural working memory) and the retro-cueing paradigm (dependent more on declarative working memory) used in Rerko and Oberauer (2013) described above. In both cases, the currently active contents of working memory that are relevant for the current operation have either been previously activated and were then switched away from (defocused, or ABA), or not been previously active (unfocused or CBA). However, in the retro-cuing paradigm, previously activated content remained privileged with lingering activation, but in the task switching paradigm, previously activated content is inhibited. If declarative and procedural working memory work analogously, then similar outcomes for reinstated information would be expected. The inconsistency between results in these two paradigms therefore makes it seem unlikely that a direct comparison of tests of procedural and declarative working memory to test the qualities of reinstated information will result in evidence of functionally analogous mechanisms.

Directly comparing these two types of test in this experiment, focusing on the characteristics of information storage in the bridge and the region of direct access, will allow for a test of the assumption of functionally analogous subsystems when considering the state of previously relevant information in those areas. To test this question, the task span and memory span paradigms (Logan, 2004) were used, focusing on particular sequences with characteristics that result in focused, defocused, and unfocused task sets, comparable to Rerko and Oberauer (2013). Participants learned task sets for their left and right hands for each of two tasks. Focused trials are repetitions with the same hand used to perform both trials. Defocused trials are those that were previously performed on trial *n*-2 with the same task set (i.e. same hand in this manipulation) and unfocused trials are those that were performed on trial *n*-2 with the alternative task set (opposite hand). See figure 12 for examples of each of these situations. Utilizing the memory span and the task span for this experiment provides a consistent task environment between two paradigms that make distinct demands on declarative and procedural working memory.

Methods.

Design. Experiment 2 was a 2 x 3 within subjects design, with span types (memory or task) and focus classification of trials (focused, defocused, or unfocused) as factors.

Participants. 20 Lehigh University students participated in Experiment 2 in exchange for partial course credit or $10/hr. compensation. None had participated in either version of Experiment 1. All participants had normal or corrected to normal vision.

Materials and procedure. Participants completed two experimental sessions of approximately one-hour each. All sessions were completed on separate days no more than one week apart. In one session, participants completed the task span, and in the other they completed the memory span. The order of the two span paradigms was counterbalanced.

The basic paradigm structure remained the same as the memory and task spans in Experiment 1: participants completed study-test blocks during which they memorized a sequence and either recalled that sequence (memory span) or performed associated categorizations (task span).

Adaptations to the basic task and memory span paradigms were made in order to facilitate characterization of targets as focused, defocused, or unfocused. First, 10 sequences were selected from the maximum possible 20 that were used in Experiment 1. All 10 sequences were made up of three instances of the height task and three instances of the shape task. The selected sequences were chosen based on certain characteristics like equating serial position and complexity for the different target types. (The 10 selected sequences and the task characteristics are presented in table 2).

												Serial Position		
	Sequence		1	2	3	4	5	6	1	2	3	4	5	6
switch after 2 tasks	1	hhshss	h	h	s	h	s	s		fo		un	de	fo
	2	hshhss	h	s	h	h	s	s			un	fo		fo
	3	hhsshs	h	h	s	s	h	s		fo		fo		de
	4	hsshsh	h	s	s	h	s	h					de	de
	5	hshssh	h	s	h	s	s	h			un	un	fo	
	1	sshshh	s	s	h	s	h	h		fo		un	de	fo
	2	shsshh	s	h	s	s	h	h			un	fo		fo
	3	sshhsh	s	s	h	h	s	h		fo		fo		de
	4	shhshs	s	h	h	s	h	s					de	de
	5	shshhs	s	h	s	h	h	s			un	un	fo	

												Serial Position		
	Sequence		1	2	3	4	5	6	1	2	3	4	5	6
switch after 4 tasks	1	hhshss	h	h	s	h	s	s		fo		de	un	fo
	2	hshhss	h	s	h	h	s	s			de	fo		fo
	3	hhsshs	h	h	s	s	h	s		fo		fo		un
	4	hsshsh	h	s	s	h	s	h			fo		un	un
	5	hshssh	h	s	h	s	s	h			de	de		
	1	sshshh	s	s	h	s	h	h		fo		de	un	fo
	2	shsshh	s	h	s	s	h	h			de	fo		fo
	3	sshhsh	s	s	h	h	s	h		fo		fo		un
	4	shhshs	s	h	h	s	h	s			fo		un	un
	5	shshhs	s	h	s	h	h	s			de	de		

Table 2. Sequences used in Experiment 2 with focused (fo), defocused (de), and unfocused (un) task sets indicated.

Note: Focused trials are repetitions. Defocused trials are those for which the task was completed on trial *n*-2, with the same task set (same hand). Unfocused trials are those for which the task was completed on trial *n*-2, with the alternative task set. Defocused and unfocused trials are always switches. In all cases, declarative information remains the same despite cases for which procedural information switches (defocused and unfocused). Only trials 2-6 were classified as being focused, defocused, or unfocused.

Sequences were selected with consideration of equating serial position, complexity, and number of switches. Grey shading in the figure indicates complementary sequences with the same focused/unfocused/defocused characteristics. Switching after two tasks and after four tasks was created by manipulating hand order, which was either rrrrll or rrllll, or for separate participants, llllrr or llrrrr.

Cues were presented on either the right or left side of the screen in a blocked and predictable order. The side of the screen on which the cue was presented indicated to the participant which hand he or she should use to respond in the test phase. The hand switch occurred either after the second (early switch) or fourth (late switch) item in the test phase. Each of these types of hand switches occurred for half of each span administration. The switch timing and initial hand was counterbalanced among participants. Participants either had hand orders [rrllll and rrrrll] or [llrrrr and llllrr].

In both the memory span and task span, cues were presented for 750 ms each. This timing scheme was selected based on previous cued task span work (Logan, 2004) with additional time at encoding to allow for eye saccades between locations. As in Experiment 1, there was a 100 ms inter-stimulus interval following target responses in the test phase.

As before, for the task span participants first practiced the task set assignments, learning which keys corresponded to which categorizations. However, in Experiment 2, there were two task sets for each of the height and shape tasks. Participants learned height and shape task sets for their left hand using the *a, s, d,* & *f* keys, as well as height and shape task sets for their right hand using 4, 5, 7, & 8 on the number pad. Horizontally adjacent keys for the left hand *a* & *s* and *f* & *d*, and vertically adjacent keys for the right hand, 4 & 7 and 5 & 8, were paired for the two responses for a given task. Different physical relationships between response keys on the left and right hands were selected to limit cross-talk between task sets for the two hands. Participants first practiced all key assignments with 28 trials for each hand/task combination. They were then guided on four practice blocks before completing 80 blocks of data collection. In the memory span, participants had only four practice blocks before completing the 80 data collection blocks (no task practice). There were also only two keys per hand used in the memory span, either *a* & *s* and 4 & 7, or *d* & *f* and 5 & 8.

Sequences were selected to maximize critical trial types. Each trial was characterized as being either focused, defocused, unfocused, or uncharacterized. Only tasks in positions 2-6 were identified in these ways. Focused trials were repetition trials on which both the task and the hand used repeated. Defocused trials were those on which the task and the hand used were the same on trial n-2, but were not the same on trial n-1. Unfocused trials were those on which the task was the same on trial n-2, but the hand used was not the same. These characterizations were made to match the characterizations made in Rerko and Oberauer's (2013) work. In all cases, declarative information, the task name, remained the same between the trials, but the nature of the procedural information, or the task set to be used could differ.

Sequences were selected for inclusion in this experiment with the intention of equating serial position for the defocused and unfocused trials—that is, a selected sequence might have a trial that would be classified as defocused if it was an early hand switch, and that same trial would be classified as unfocused with a late hand switch. The overall number of switches and complexity in the sequences were also considered and equated among the different trial types.

Results.

Data were first analyzed on a participant level to evaluate basic accuracy and memory performance. No participants were excluded from analysis based on the exclusion criteria used in Experiment 1. Trials for which RTs were greater than 3500 ms or shorter than 150 ms or on which errors were made were removed from analyses. Task memory and accuracy data for the task span can be found in Appendix B.

Target RT. The critical question in Experiment 2 is the RT relationship between focused, defocused, and unfocused trials for the memory span and task span. The effects of span type (memory or task) and focus classification (focused, defocused, or unfocused) on target RT were examined with a 2 x 3 repeated measures ANOVA. These data are presented in figure 15. The effect of span type was significant, $F(1, 19) = 1005.72$, $p < .001$, $\eta^2_{\text{partial}} = .981$. Participants took considerably longer to perform tasks in the task span than they did to recall tasks in the memory span. The effect of focus classification was also significant, $F(2,38) = 99.14$, $p < .001$, $\eta^2_{\text{partial}} = .839$, that was qualified by a significant interaction, $F(2, 38) = 55.17$, $p < .001$, $\eta^2_{\text{partial}} = .744$.

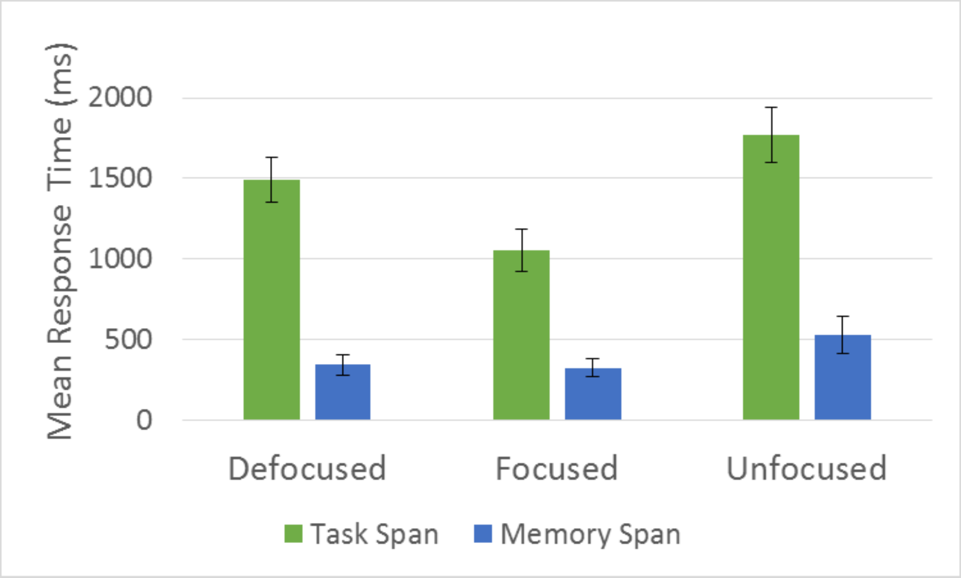

Figure 15. Mean RT for items in Experiment 2 for memory and task spans as a function of focus classification[5].

In order to determine whether the memory span replicates Oberauer's (2002) declarative working memory results, the defocused trials were compared to both the unfocused and focused trials. Oberauer's declarative results would predict that there should be a benefit for defocused trials compared to unfocused trials, just as there is a benefit for focused trials compared to unfocused trials. For the memory span, Fisher's LSD pairwise comparisons reveal that the difference between defocused and unfocused trials was significant, *p* < .001, whereas the difference between defocused and focused trials was not significant, *p* = .219. These results reveal a benefit for defocused trials compared to unfocused trials that is as substantial as the benefit seen for focused trials.

[5] As indicated in table 2, only focused trials appeared in the second serial position in any sequence. These trials were removed from the calculation of average RT for that focus classification, and the numerical difference was only a 3ms shift. Therefore, these second position trials were left in subsequent calculations.

The same comparisons were made for the task span to determine whether procedural working memory produces the same pattern as declarative, which would support Oberauer's model of a functional analogy. For the task span the difference between defocused and unfocused trials was significant *p* <.001. The difference between defocused and focused trials was also significant, *p* <.001. These results suggest that consistent with Oberauer's model, defocused procedural information shows a benefit compared to unfocused information. This benefit was not as large as that of focused information compared to unfocused information in the task span.

At first blush these data appear very consistent with Oberauer's predictions: it appears that there is a benefit for defocused trials compared to unfocused trials for procedural information. However, further investigation indicates that this is not actually the case. The requirement for classification as an unfocused trial is that the trial is not a repetition and that participants are doing the same task as trial *n*-2 with a different task set (different hand). Therefore, these unfocused trials are comprised of trials on which the participants switched task sets immediately prior, or switched task sets following trial *n*-2 (i.e. one intervening trial after switching hands). For example, for the sequence SH|SHHS with the hand switch indicated by the vertical line, the task in the third serial position would be an unfocused trial for which the hand switch occurred immediately prior to that task's performance. For the same sequence SH|SHHS, the task in the fourth serial position would be an unfocused trial for which the hand switch occurred after trial *n*-2. When we consider these two types of unfocused trials (unfocused-immediate vs. unfocused-deferred) independently, they appear to be quite different.

These separated unfocused trials were entered into a 2 x 4 repeated measures ANOVA with span type (memory and task) and focus classification (focused, defocused, unfocused-immediate, and unfocused-deferred). These data are presented in figure 16. The effect of span type was significant, $F(1, 19) = 805.45$, $p < .001$, $\eta^2_{\text{partial}} = .977$. Participants took longer to respond to targets in the task span than in the memory span. The effect of focus classification was also significant, $F(3, 57) = 122.20$, $p < .001$, $\eta^2_{\text{partial}} = .865$. Participants were fastest for focused trials, and slowest for unfocused-immediate trials. There was a significant interaction of span type and focus classification, $F(3, 57) = 37.37$, $p < .001$, $\eta^2_{\text{partial}} = .663$. Unfocused-immediate trials were especially costly and defocused trials did not match focused trials in the task span. A follow-up 2 x 2 repeated measures ANOVA indicated that there was no interaction between unfocused-deferred and defocused trials for the memory and task span, $F < 1$. Statistically, these two conditions were identical—therefore there was no benefit for defocused trials compared to unfocused trials without a hand switch immediately prior. In fact, the same cost of switching hands occurs even when added to repetition (focused) trials. A 2 x 2 x 2 repeated measures ANOVA with span type (memory and task), task transition (repetition or switch) and hand transition (repetition or switch) was conducted to examine the effects of switching hands. The effect of hand transition was significant, $F(1,19) = 109.\ 01$, $p < .001$, $\eta^2_{\text{partial}} = .852$. Switching hands was very costly, and this cost explains the effects seen in the first analysis with unfocused trials undivided.

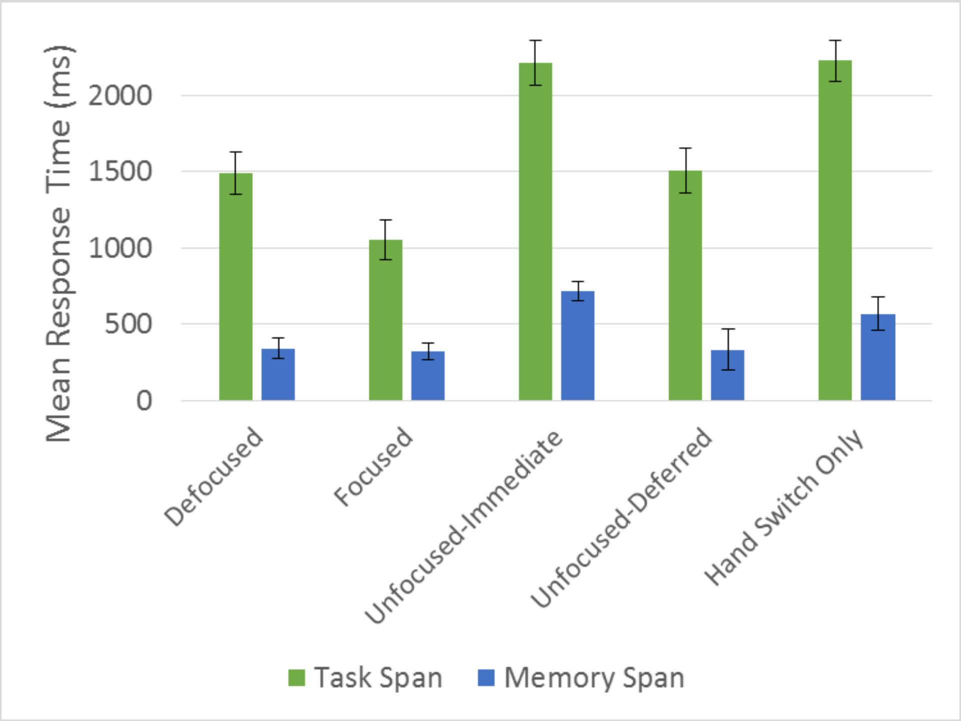

Figure 16. Mean RT for items in Experiment 2 for memory and task spans as a function of focus classification with unfocused trials separated into deferred and immediate varieties based on proximity to hand switch.

Discussion.

Experiment 2 tested the fate of previously focused information in procedural working memory compared to previously focused information in declarative working memory. A comparable previous examination of declarative working memory (Rerko & Oberauer, 2013) suggests that this type of information remains in a privileged state, with higher activation than an item that was not previously utilized. In order to test the hypothesis that previously focused information in procedural working memory does *not* remain in a privileged state (which would be predicted based on backwards inhibition effects), trials for the task span and memory span were characterized as either being focused (repetition), defocused (same task and task set on trial n-2) or unfocused (same task, different task set on trial n-2). The prediction based on backwards inhibition effects was that defocused trials would show a longer RT to targets compared to unfocused trials in the task span[6]. The prediction based on Rerko and Oberauer's (2013) work was that defocused trials would show shorter RT to targets compared to unfocused trials in the memory span. At first blush, the data from both the memory span and task span appeared to be consistent with Oberauer's model and the previous work studying declarative working memory. However, when the unfocused trials were broken down as trials with the hand switch immediately prior to the unfocused task and trials with the hand switch one trial earlier, there were no differences between the unfocused-deferred condition and the defocused condition, for either the memory or task span. This way of breaking down the data was supported when considering a "hand-switch only" condition, which also showed a steep cost of switching hands even when the actual task stayed the same. Therefore the data suggest that there is no privilege for defocused procedural information in the current experiment.

A critical requirement needed to be met in order to draw theoretical conclusions from the task span in Experiment 2: the memory span data had to be consistent with Rerko and Oberauer's (2013) retro-cuing data. That is, for the memory span in

[6] While data from Experiment 2 did not demonstrate typical backwards inhibition effects, we cannot say for certain that there was a failure to replicate Mayr & Keele (2001). In the original backwards inhibition experiments, the same exact task and task set was being inhibited (i.e. AB**A**) when compared to not having done any version of that task previously (i.e. point of comparison was CB**A** sequence). In our defocused condition, participants are performing the same task they had previously switched away from (i.e. HS**H**), but the unfocused condition is not comparable to the CBA situation used in Mayr and Keele (2001). The unfocused comparison condition represents performing the same task previously switched away from, but with a new set of task rules. Therefore it is more akin to a H'SH situation, which is not the same as the CBA situation used in Mayr and Keele's (2001) work. Therefore we cannot say there is a failure to replicate backwards inhibition, because the lack of inhibition could just be a result of the different task comparison.

Experiment 2, defocused information needed to show a privileged level of activation compared to unfocused information. There needed to be evidence that when information was transferred from the focus of attention to the region of direct access, and then returned to the focus of attention, there was a benefit for that previously active state. While this pattern appeared to hold at first, once the data were reanalyzed considering the timing of the task set change, there was no effect beyond just the cost of switching hands. Therefore, this critical assumption was not met based on the Experiment 2 data and it is impossible to make a claim about why the defocused privilege did not appear in the task span. Potentially, defocused and unfocused information are not handled the same way in both subsystems. Alternatively, and more likely considering these data, the span paradigms as used in Experiment 2 were not properly designed to capture the different focus classification types without adding a confound of hand switch.

There are several differences between the current work and that of Rerko and Oberauer (2013) that could have contributed to this lack of replication. According to Rerko and Oberauer, the typical retro-cue benefit comes from the cued item becoming the focus of attention, which is highly active and easily accessible. RTs to this item later are faster because that increase in activation lingers. In their work, participants had different declarative items activated in sequence, but a response was made only to the last cued item. In Experiment 2, participants had to make a response to every item in the sequence. Perhaps this act of responding to each item changed the way in which the input-output bindings were activated. Potentially, the retro-cue benefit disappears when the information is actually used, rather than just activated. Oberauer and colleagues (2013) suggest that when carrying out list switches, some residual activation of the switched away from list remains in the region of direct access. This accounts for congruency benefits observed for common items among two lists. This finding appears at odds with the proposal that making a response in the memory span would change the nature of input-output bindings, which could account for non-replication of the retro-cue benefit. Perhaps the congruency benefit would be even stronger if there was only activation of a particular item, rather than the use of it. This is an empirical question not discussed in Oberauer's model which could support the possibility that residual activation in the binding matrix is stronger when information is not used, and there is some resetting process that occurs if information is used to make a response or selection. If the exploration of this possibility produces results consistent with the idea that whether information is used or not used changes the nature of residual bindings, Oberauer's model would need to be updated to account for this finding.

Considering both the retro-cuing paradigm and the memory span in the context of Oberauer and colleagues' computational model may clarify the nature of differences between them that could have led to the inconsistent results seen in the current experiment. In retro-cuing, the memory set is first learned and represented in the set layer. Then, a cue (spatial location) is given, which is represented in the input layer. This cue increases the activation of one element of the binding matrix, which prepares a response. This response is not made. An additional cue is given, which results in the preparation of another response, but the original response has some residual activation

(Rerko & Oberauer, 2013). When the final cue is given, if it is the same as the original cue, the residual activation from the previous cuing will result in improved performance because the appropriate input-output binding is already primed for responding.

In the memory span, the procedure occurs in largely the same way. A memory set is learned and represented in the set layer. A cue (serial position) is given, which is represented in the input layer. In this case, the input-output bindings are strengthened as a response is prepared, but that response is actually made before the next cue is inputted, which is different from the retro-cuing situation. The model is not clear about whether the same residual activation remains in this memory span case, or whether the activation is lessened compared to the previously described circumstance and not carried over as strongly to the final cued trial. In the case of the current experiment, no benefit is seen and no residual activation of the previously used information is inferred. Future work will need to examine the possibility that producing a response resets the binding matrix in a way that preparing a response does not.

Additionally, there is a large variability in timing between Rerko and Oberauer's (2013) work and Experiment 2. In one of their experiments for example, there was a memory array, a 1000 ms interval, the cue, another 1000 ms interval, and finally the probe to which they made their response. (The total time per trial stayed the same in each experiment regardless of how many cues they presented). In Experiment 2, the timing depended on the participants' responses, but on average, the participant could be done with an entire memory span sequence within that same timeframe. Perhaps the benefit for defocused trials would appear after a time delay. However, it seems unlikely that simply increasing the time in between responses would result in a selective benefit for the defocused trials without also affecting the unfocused trials. In a future experiment, the timing and general procedure could be made more equivalent to Rerko and Oberauer's work by having the participants learn an entire sequence, then after a delay a particular serial position could be cued (or several serial positions) and they would need to make their response to that cued item.

Finally, there were many more potential items that could be activated by cues in the Rerko and Oberauer (2013) work than there were in the current experiment. In the memory span, there were only two possible task names, height and shape. Perhaps there is no benefit for defocused compared to unfocused trials when there are only two items. Rerko and Oberauer say that focusing on an item results in a stronger bond of that item and its context in the focus of attention[7]. After that item becomes irrelevant or the focus moves to a different item, the increased strength of binding remains in the region of direct access. In the current data, there is no difference in the memory span between immediate repetitions (focused), having done the same task on trial n-2 (defocused), or having done the same task on trial n-2 with a different hand (unfocused-deferred). It is possible that there's no chance to see a difference among these three different conditions because there are only two items in the memory span—height and shape. Maybe in the memory span, an "unfocused" trial really is not unfocused at all,

[7] An additional potential difference between the retro-cuing paradigm and the memory span is the nature of the context; for the retro-cuing paradigm the context is spatial location, while for the memory span the context is serial position. Perhaps spatial location results in stronger benefits for defocused information because the cues themselves are actually identical, not just the item that *gets* cued as in the memory span.

because participants are simply representing the sequences as category labels (e.g. height vs. shape) rather than task sets (e.g. height goes with 5, shape goes with 8) (Arrington & Logan, 2004). We see a cost of hand switch because it serves as a reset the mental state, but there is not really a task set switch in the unfocused condition, and that hand switch cost goes away once the new hand is established.

For future directions, one potential way to modify the task span to eliminate the hand switch confound would be to increase the number of tasks from two to three, making the task span more similar to the task switching paradigms used when studying backwards inhibition. In this case, unfocused trials would be those on which a completely different task and task set was performed on trial n-2, whereas defocused trials would be those on which the same task and therefore task set would have been performed on trial n-2. This change would eliminate the disrupting factor of the hand switch that complicated interpretations in Experiment 2. This seems like a promising modification. However, one problem with this change is that in this case the declarative information would be changing along with the procedural information, which was one factor Experiment 2 originally attempted to avoid. This caveat does not mean that this alternative approach should not be taken, just that any results should be interpreted with this declarative change in mind.

Experiment 2 considered the fate of information that was previously in the response focus or focus of attention and then returned to that level of activation, in order to attempt to address the question of what happens to information after it has been activated. An alternative dimension of the bridge and region of direct access is the shift in information between these intermediary stores and activated long term memory, and how capacity influences this shift. This dimension is examined in Experiment 3.

Experiment 3: Set-size Effects in Procedural Working Memory

Common to descriptions of working memory as a functional component of cognition is the assumption that the system allows for flexible adjustment of contents in the face of changing environments or task goals (Bunting & Cowan, 2005; D'Esposito & Postle, 2015; Oberauer, 2009). This flexibility is critical for updating of the contents of working memory particularly considering the variability present in different types of tasks. One mechanism by which working memory may flexibly adapt to different task constraints is by utilizing activated long term memory as a back-up maintenance method for information that is not currently useful but may be later (Cowan, 1995; 1999; D'Esposito & Postle, 2015; Oberauer, 2007). Oberauer suggests that the working memory system can selectively lower the activation of information that is not immediately relevant, decreasing that activation to the level of activated long term memory, and then rapidly retrieving this information at a later time when it is necessary. This mechanism could be useful for flexibly switching the focus of attention between multiple sources of information.

For example, as mentioned in the main introduction, information that is not currently in the region of direct access but has recently been used or encoded is proposed to be part of the activated portion of long term memory (Oberauer, 2002; 2005). As a reminder, in that experiment participants memorized two memory sets (series of numbers) of varying set size that either needed to be manipulated with arithmetic operations (active—proposed to be stored in region of direct access) or not (passive—proposed to be stored in long term memory). Sometimes both lists were active, and sometimes only one list was active. After 9 rounds of updating the active list with arithmetic operations, participants were tested on the sequence of digits of both lists. In terms of the computational model, in the case of the active list, the set layer was different compared to the start, whereas the set layer of the passive list remained the same. While both lists had to be recalled and therefore were stored in some fashion, the way in which active and passive lists were maintained was expected to differ. Support for this hypothesis was found when considering response latencies for arithmetic updating to active lists as a function of set size. There was an effect of set size for active lists but not for passive lists. The effects of having more activated information in the region of direct access (larger set-size of the active list) slowed down response latencies for the updating task related to the active list, but having more activated information in long term memory (larger set-size of the passive list) did not slow down RTs for the updating task. Oberauer (2002) suggests that information that is merely available for later processing is kept in a functionally different state than information that is being maintained, and that storage in activated long term memory is separate from maintenance in the region of direct access. No analogous process has been analyzed in procedural working memory. It is unclear whether storage for procedural information in activated long term memory is similarly separate from maintenance in the bridge.

In Oberauer's (2002; 2005) work, declarative items were the foundation of the lists used in the mathematical updating task. In tasks that require procedural working memory (one example of which is task switching), stimulus-response mappings are the foundations of task sets. Just as the number of declarative items or elements in a list can be varied to change the amount of information that needs to be maintained in declarative working memory, the number of stimulus-response mappings in a task set can be varied to change the amount of information that needs to be maintained in procedural working memory. Oberauer (2002) showed that set size of activated declarative long term memory does not affect task performance. Set size does matter, however, in the region of direct access. Similarly, for perceptual WM, different set sizes for non-active task sets stored in activated long term memory should not influence task performance which is based on the task set currently active in the bridge.

The behavior observed in task switching environments suggests that the bridge may be quite flexible in drawing task sets from activated long term memory—participants are successful in rapidly shifting between tasks whether these tasks are difficult or numerous. However, the costs of this transition from long term memory to the bridge are influenced by the characteristics of stimulus-response mappings. How stimulus-response mappings are presented to participants can change how they behave when switching tasks. For example, a collection of eight stimulus-response mappings that are presented as part of two task sets is qualitatively different than the same eight stimulus response mappings that are not represented as being part of task sets (Dreisbach, et al., 2007). Specifically, there are switch costs when switching between responses that are encoded as part of two different task sets, while there are no switch costs when switching between the same responses not encoded as belonging to different task sets. These authors established this finding by presenting two groups of participants with eight stimulus-response mappings. Each group of participants learned the same stimulus-response mappings. They were not obviously part of a task set, but could be categorized as such when the connection between them was made obvious. For one group this connection was defined, and for the other group it was not. Participants were either shown eight words and asked to memorize the responses associated with each of them, or instructed to categorize them based on their first letter or whether they were an animal or not (these were the categories that were not obvious unless made explicit). Switch costs arose for the group that had the task sets pointed out to them, while there were no switch costs in the group that was naïve to the connection. Considered in light of Oberauer's model, the results of this experiment indicate that the same eight response mappings are pulled into the bridge differently depending on how the stimulus-response mappings are presented to the participant. According to Oberauer's computational model, (Oberauer, et al., 2013), in the case of no task sets, there should be no switch cost because individuals do not need to update the binding matrix represented in the bridge on each trial. This means that the input-output associations will not be updated by the set layer, because there is no activated task set. In contrast, when stimulus-response mappings are represented as task sets, updating is necessary on switch trials, and the switch cost emerges.

Furthermore, Hübner, Kluwe, Luna-Rodriguez, and Peters (2004) compared task switching performance when two tasks had four stimulus-response mappings each to a situation in which one task had four stimulus-response mappings and the other had two stimulus-response mappings. The tasks were “letter” and “color”, and participants had to map four letters and four colors to responses, or four letters and two colors. For the mixed task set condition, participants were faster to respond when there were only two stimulus-response mappings (color task) compared to four (letter task), and there were larger switch costs when there were two-stimulus response mappings compared to four. Participants enjoyed more of a benefit of repeating tasks when there were fewer stimulus-response mappings to select from. While their research does not directly speak to the distinction between the bridge and activated long term memory, their results can be interpreted in that framework. At first blush, these findings appear inconsistent with what would be predicted based on Oberauer’s (2002) declarative list switching tasks, but it is actually hard to determine whether the decreased RTs in the two stimulus-response mappings condition come from a decreased set-size in long term memory, or from the fewer total responses needed to be maintained in the bridge on any given trial, as their experiment was not designed to test questions of set-size effects in procedural working memory.

The current experiment examines this question, asking whether there are effects of set size on activated long term memory for procedural information. Oberauer (2002) found no effect of set size declarative information. The current experiment uses the task span in order to systematically load either the bridge or activated long term memory on each trial, with appropriate controls in place for comparing performance. In the current experiment, participants learned two task sets, while the number of stimulus-response mappings per task set varied among participants. Some participants learned two stimulus response mappings for each task. Some learned two mappings for one task and four for the other. Some learned four mappings for both tasks. This design allowed for a direct comparison with the declarative list updating paradigm used in Oberauer (2002). That is, in his experiment, there were always two short lists of numbers learned. This is comparable to the two task sets used here. The number of items in those lists varied, just as the number of stimulus-response mappings varied here. One task set at a time was passive in the current experiment, as it is not being actively applied to the currently presented stimuli. The task sets are recruited on a random basis according to the memorized task span sequence. In Oberauer's (2002) experiment, when a list was passive, it was not recruited for updating, and was presumed to be maintained in activated long term memory, ready to be recalled at the appointed time. The task environments for these two experiments were also similar. No explicit cues were presented to participants to aid in their memory for the sets for either the declarative updating task or the task span.

Experiment 3 is designed to test how capacity limitations of the bridge impact working memory performance, and whether the capacity limitations in the intermediary bridge store are different from those of activated long term memory as Oberauer (2013) proposed. The question of the nature of capacity limitations in procedural working memory is important to address considering the finding that there is no set size effect for passive information in activated declarative long term memory (Oberauer, 2002). Is it true that increasing the contents of the bridge slows performance but there is no effect of having more activated procedural information in long term memory, just as there is no effect of having more activated declarative information in activated long term memory? Such a result would provide support for the claim of a functional analogy between these two subsystems of working memory.

Methods.

Design. In Experiment 3 participants in three groups performed the task span with two tasks of varying set size, which was intended to change how the bridge and activated long term memory were loaded on any given trial. This experiment had a 2 x 2 mixed design, with a within subjects manipulation nested within the two factors. One factor was the active set size (2 or 4 stimulus-response mapping) and the other factor was the passive set size (2 or 4 stimulus response mappings). The main contrast involved a within-subjects comparison of effects of active load and passive load, with unequal set sizes in each case, theoretically resulting in varied load in the bridge and long term memory. For a group of participants with unequal task set loads there were either two or four stimulus-response mappings theoretically in the bridge and either two or four stimulus-response mappings theoretically in long term memory. The position of the load changed over the course of the block depending on which task was being performed.

There were also two further groups of participants: one performing two tasks with two stimulus-response mappings each and another performing two tasks with four-stimulus response mappings each. These groups served as between-subjects comparisons—there was no variability in the set size of the active and passive tasks for them. Conditions in the design are outlined in Table 3. Local transition and a global complexity measure—switch frequency were also manipulated, just as in Experiments 1 and 2.

		Active Set Size	
		2	4
Passive Set Size	2	2_2	4_2
	4	2_4	4_4

Table 3. Summary of conditions in Experiment 3. There were either 2 or 4 active stimulus-response mappings and 2 or 4 passive stimulus-response mappings. The greyed cells were represented by one participant group. For this group whether the active or passive set was larger varied on a trial by trial level. For the other two groups the active and passive sets were always the same size.

Participants. Experiment 3 involved 86 Lehigh University participants who completed a single one-hour session for partial course credit or $10/hr. compensation. 80 participants were included in the final results: 20 participants for each of the comparison groups (2-2 and 4-4) and 40 in the critical variable task load group (set sizes of 2 or 4 active and 2 or 4 passive). All participants had normal or corrected to normal vision and none participate in Experiments 1 or 2.

Materials and procedure. Participants in this experiment completed only the task span. The structure of the task span was the same as in Experiments 1 and 2, but the nature of the two tasks (shape and fill, described below) was different, and varied between groups. One group of participants learned two tasks with two possible categorizations each. Another group learned two tasks with four possible categorizations each. The final group learned two tasks, but one task had two categorization options and the other had four categorization options. Regardless of group, all participants received 36 trials of practice with each task before completing four practice blocks of the task span procedure described above. The tasks included were shape and fill—height was not selected for this experiment because having four levels of height for such simple stimuli was expected to be perceptually challenging for participants to discern. Shape options were oval, rectangle, pentagon, and triangle. Fill options were dotted, hashed, full, or empty. Therefore there were 16 possible stimuli options in this experiment (full oval, hashed oval, dotted oval, etc.). Participants in the 4-4[8] group saw all possible stimuli. Participants in the 2-2 and 4-2 groups saw only the selection of stimuli relevant for their categorization options. There were four possible combinations of stimulus selection for each of these two groups (e.g. stimuli relevant for only pentagon, triangle, hashed and dotted categorizations), and participants were assigned to one of those possibilities randomly. Participants encoded and performed all 20 possible sequence orders (table 1) of two tasks three times each a total of 5 times over the course of the experiment, for a total of 100 study-test blocks.

For the 40 participants comprising the critical 4-2 group in this experiment, which task was assigned to each number of mappings was counterbalanced within the group. These participants made up two cells of the design, as on some trials they had

[8] In mention of condition names, the first number always refers to the number of stimulus-response mappings for the task believed to be represented in the bridge (currently active), and the second number refers to the number of stimulus-response mappings in the task believed to be represented in activated long term memory (currently passive).

two stimulus-response mappings in the bridge (2-4 condition) and on some trials they had two stimulus-response mappings in activated long term memory (4-2 condition). The two other groups of 20 participants had consistent loads in the bridge and long term memory on every trial, and served as baseline comparisons for the variable conditions described above. A mostly between-subjects design (for these comparison conditions) was selected to prevent carryover of task sets between the two conditions. The load believed to be stored in long term memory is critical to answering the current research question; therefore it was important to ensure there is no possibility of residual activation.

The task span proceeded as described in previous experiments. Participants encoded the presented sequence, and then performed either the shape or fill task on presented stimuli, matching the order of their task performance with the sequence they had just memorized. All that varied between the groups was the number of response options relevant for each trial of task performance. Participants used the *a, s, d,* and *f,* and the *j, k, l,* and *;* keys to respond to targets in this experiment. For the 4-4 group all eight keys were used. For the 2-2 group only the *d, f* and *j, k* keys were used. For the variable 4-2 group, four keys for one hand and two keys for the other hand were used. Key assignments were counterbalanced. As an additional change from previous experiments, the response to stimulus interval (RSI) was modified. Previous task span work has used an RSI of 100ms at performance (Logan, 2004) but in this version of the experiment, this RSI was elongated to 500ms, as Oberauer (2002) suggests that the disappearance of the list length effect takes some time—that is, the transfer of previously used task sets from the bridge to long term memory and vice versa will take some time to complete. All other timing was the same as in Experiments 1 and 2.

Results.

As in Experiments 1 and 2, data were first analyzed on a participant level to evaluate basic accuracy and memory performance. Participants were excluded from further analysis if their task accuracy (their object categorization performance) was lower than 80% or if their task memory was lower than 75%. Six participants were excluded from analysis based on low accuracy (*n* = 1), low task memory (*n* = 2), or both (*n* = 3). Therefore data from 80 participants were included in the final analyses—20 of whom were from the 2-2 group, 20 from the 4-4 group, and 40 from the 4-2 group. Trials for which RTs were greater than 3500ms or shorter than 150ms or on which errors were made were removed from analyses. Accuracy data can be found in Appendix C.

Target RT. Participants' RTs to targets were first analyzed as a function of set size condition. Several sub-comparisons were made without completing an omnibus ANOVA because the 4-2 and 2-4 condition comparison was within-subjects, while the 2-2 and 2-4 and 4-2 and 4-4 comparisons were between-subjects. RTs for the 4-4 condition (1124 ms) were not significantly different from the 4-2 condition (1083 ms) as evidenced by an independent samples t-test, *t*(58) = .860, *p* = .393. The 2-2 condition (862 ms) was not significantly different from the 2-4 condition (925 ms) as evidenced by an independent samples t-test, *t*(58) = -1.33, *p* = .189. A paired samples t-test for the within subjects comparison between the 2-4 and 4-2 condition was significant, *t*(39) = -7.24, *p* <.001.

Repetition and switch trials were also considered separately for all four conditions. These data are presented in figure 17. Again, sub-comparisons between the different conditions are presented here, first for repetition trials and then for switch trials. For repetition trials, RTs for the 4-4 condition were not significantly different from the 4-2 condition as evidenced by an independent samples t-test, *t*(58) = -.19, *p* = .848. RTs for the 2-2 condition were not significantly different from the 2-4 condition as evidenced by an independent samples t-test, *t*(58) = -.91 *p* = .368. A paired samples t-test for the within subjects comparison between the 2-4 and 4-2 condition was significant, *t*(39) = -8.16, *p* < .001. For switch trials, RTs for the 4-4 condition were significantly different from the 4-2 condition as evidenced by an independent samples t-test, *t*(58) = 1.47, *p* = .148. RTs for the 2-2 condition were also significantly different from the 2-4 condition as evidenced by another independent samples t-test, *t*(58) = -1.49 *p* = .140. A paired samples t-test for the within subjects comparison between the 2-4 and 4-2 condition was also significant, *t*(39) = 4.84, *p* < .001. These data show that repetition trials are in line with predictions according to Oberauer's (2002) declarative data, while switch trials are not.

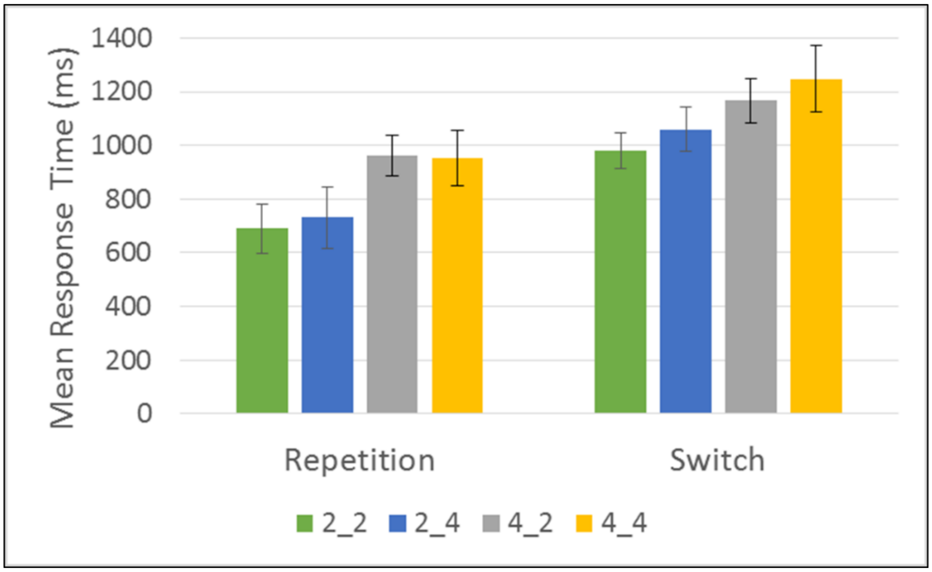

Figure 17. Mean RT to targets in Experiments 3 as a function of task transition and size of the active and passive task sets.

Next, RT data were analyzed as a function of switch frequency. These data are depicted in figure 18. Data are again evaluated with sub-analyses comparing specific condition pairs independently. A 2 x 5 mixed factor ANOVA with set size condition (4-4, and 4-2, between subjects) and switch frequency (1, 2, 3, 4, & 5 switches, within subjects) on target RT yielded a significant effect of switch frequency, *F* (4,232) = 79.94, $p < .001$, $\eta^2_{partial} = .580$. Participants' RTs to targets generally slowed as the switch frequency increased. The effect of condition was not significant, *F*(1,58) < 1. The interaction between these two factors was also not significant, *F*(4,232) = 1.39, *p* = .243. A 2 x 5 mixed factor ANOVA with set size condition (2-2, and 2-2, between subjects) and switch frequency (1, 2, 3, 4, & 5 switches, within subjects) on target RT yielded a significant effect of switch frequency, *F* (4,232) = 140.21, $p < .001$, $\eta^2_{partial} = .707$. The effect of condition was not significant, *F* (1,58) = 1.91, *p* = .172. The interaction between these two factors was also not significant, *F* (4,232) = 2.97, *p* = .09. Finally, a 2 x 5 repeated measures ANOVA with set size condition (2-4 and 4-2, within subjects) and switch frequency (1, 2, 3, 4, & 5, within subjects) on target RT yielded a significant effect of switch frequency, *F* (4,156) = 91.19, *p* < .001, $\eta^2_{partial}$ = .70. The effect of condition was also significant, *F* (1,39) = 60.64, *p* < .001, $\eta^2_{partial}$ = .609. Participants were slower in the 4-2 condition than the 2-4 condition. The interaction between these factors was significant, *F* (4,156) = 4.75, *p* = .035, $\eta^2_{partial}$ = .109.

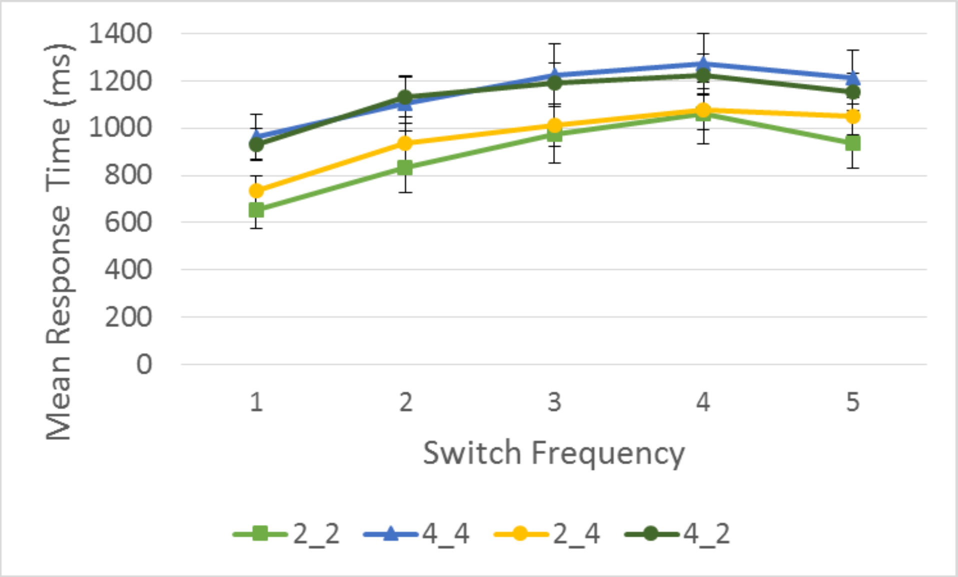

Figure 18. Mean RT to targets in Experiments 3 as a function of switch frequency and size of the active and passive task sets.

Task Memory. Participants' memory for individual tasks was considered next—which required both memory for the task and a correct categorization of the stimulus. List memory was not examined in Experiment 3 as in the variable load conditions (2-4 and 4-2), the active/passive-content distinctions are made on a trial by trial level. Since the list memory variable must be calculated using data from the entire sequence, it was not appropriate here and task memory was considered instead.

Task Memory, or memory for each individual task in a sequence, was analyzed as a function of switch frequency for the same paired conditions analyzed above. A 2 x 5 mixed factor ANOVA with set size condition (4-4, and 4-2, between subjects) and switch frequency (1, 2, 3, 4, & 5 switches, within subjects) on task memory yielded a significant effect of switch frequency, $F(4,232) = 46.98$, $p < .001$, $\eta^2_{\text{partial}} = .448$. The effect of condition was not significant, $F(1,58) = 2.18$, $p = .145$. The interaction between these two factors was not significant, $F(4,232) < 1$. The 2 x 5 mixed factor ANOVA with set size condition (2-2, and 2-4, between subjects) and switch frequency (1, 2, 3, 4, & 5 switches, within subjects) on task memory yielded a significant effect of switch frequency, $F(4,232) = 48.51$, $p < .001$, $\eta^2_{\text{partial}} = .455$. The effect of condition was also significant, $F(1,58) = 7.45$, $p = .008$, $\eta^2_{\text{partial}} = .114$. The interaction between these two factors was not significant, $F(4,232) = 1.56$, $p = .217$. Finally the 2 x 5 repeated measures ANOVA with set size condition (4-2, and 2-4, within subjects) and switch frequency (1, 2, 3, 4, & 5 switches, within subjects) on task memory yielded a significant effect of switch frequency, $F(4,156) = 69.24$, $p < .001$, $\eta^2_{\text{partial}} =.640$. The effect of condition was not significant, $F(1,156) < 1$. The interaction between these factors was also not significant, $F(4,156) < 1$.

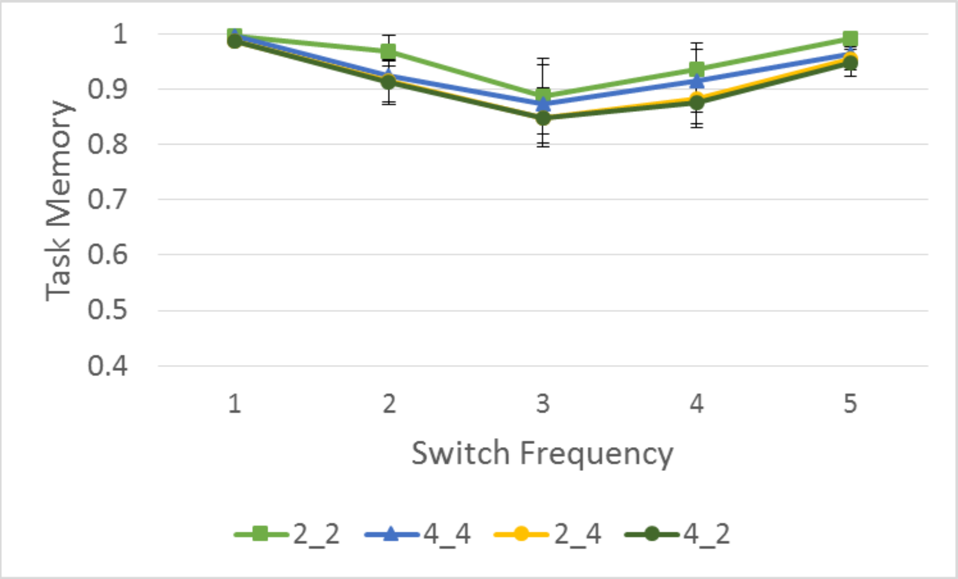

Figure 19. Mean task memory for items in Experiments 3 as a function of switch frequency and size of the active and passive task sets.

Discussion.

In Experiment 3, three separate groups of participants engaged in the task span with two tasks of varying stimulus-response mapping numbers. One group had two mappings for each task, another group had four mappings for each task, and the critical group had two mappings for one task and four mappings for the other task. The critical group allowed for an examination of the effects of a larger load in the bridge and a smaller load in activated long term memory or vice versa, because the size of their active task set changed on a regular basis depending on which task was being performed. According to comparable tests of declarative working memory (e.g. Oberauer, 2002), there should be no set size effects in activated long term memory, and therefore participants' performance should depend only on the contents of the bridge on any given trial. This implies that the 2-2 and 2-4 conditions and the 4-4 and 4-2 conditions should be identical, and the 2-4 and 4-2 conditions should show significant differences from one another.

RT Results were consistent with predictions based on Oberauer’s model, as performance varied depending on the active contents of the bridge and not based on the total number of stimulus-response mappings. Whether examining RT as a function of transition (for repetition trials) or complexity, the 2-4 and 4-2 conditions were different from one another while the 2-2 and 2-4, and 4-4 and 4-2 conditions were not different from one another. In both of the variable conditions (4-2 and 2-4), there was the same total number of mappings. What differed between them was the active contents of the bridge and passive contents of long term memory. Notably, performance in the 2-4 condition was similar to performance in the 2-2 condition, and performance in the 4-2 condition was similar to the 4-4 condition for RT. This finding suggests there is no set size effect for procedural information in activated long term memory, just as there was no set size effect for declarative information (Oberauer, 2002; 2005). Finally, in Experiment 3, the 4-4 and 4-2 conditions elicited longer RTs overall than the 2-2 and 2-4 conditions. When there are more items in the bridge in candidacy for selection, it takes more time to select among them (Oberauer, 2002). However, what is critical for the interpretation of this work in the context of Oberauer’s model is that in most cases the inactive set size did not change performance.

When examining RT for repetition and switch trials separately, repetition trials exemplified the predicted pattern while switch trials were inconsistent with predictions based on declarative results (Oberauer, 2002). This difference in transition effects can be explained when thinking about the nature of switch or repetition trials, particularly in the context of the computational model. On repetition trials, the active task set it stable from trial *n*-1 to trial n. There is no updating of the binding matrix between the input and output layers. Under this circumstance we see clear separation between the 2-4 and 4-2 conditions, while the 2-2 and 2-4 and 4-2 and 4-4 conditions do not differ from one another. On switch trials, participants have either 2 or 4 s-r mappings active in the bridge, but these mappings are not the same mappings that were active on trial *n*-1. Whether participants are switching from 2 s-r mappings to 4, 4 s-r mappings to 2, or between two task sets of the same size, there is significant restructuring of the input-output bindings that must occur before task performance can be completed. This restructuring takes time, and based on the data from Experiment 3 it appears that this retrieval of information from activated long term memory into the bridge takes more time with increasing set size of both active and passive lists. Based on this explanation it makes sense that the repetition trials are clearly demonstrating the predicted pattern while the switch trials are not.

Capacity of the Bridge. According to Risse and Oberauer (2010), in declarative working memory, only objects active in the region of direct access contribute to competition for selection, not those active in long term memory. That assertion is consistent with the current data: not presently relevant task sets do not interfere with bridge processing. The non-active task set is represented in a separate way (activated long term memory) that produces little interference for on-going task execution (Oberauer & Gothe, 2006). It appears based on these data, that there is a distinct capacity limit for the bridge, whereas there is either a much larger capacity, or no capacity limit for activated long term memory. This assertion fits with Oberauer's conception of capacity in working memory. He suggests that capacity is limited because of the limitations of bindings between items and their contexts, or conditions and appropriate actions (Oberauer, et al., 2013). As explained in the context of the computational model, bindings between the input and output layers are limited. Capacity limitations arise because with an increased number of bindings, there is increased competition and interference between those bindings. Ultimately with more bindings there is less distinctiveness between each of them, which results in effects such as longer reaction time or increased errors as the interference and competition must be overcome.

The nature of this capacity is quite interesting to consider. Evidence points to the idea that the bridge maintains one task set at a time, while other information is stored with lower levels of activation (Buchler, Hoyer, & Cerella, 2008). This capacity may arise from the maximum activation in associations between the input and output layers—with more associations, there is more competition and interference between potential stimulus-response mappings. If the working memory system is organized as Oberauer proposes, which is an assertion the current data would support, storage and processing can occur independently when processing is not occurring on those particular stored items. That is, there can be maintenance of particular items (i.e. task sets in the task span) and simultaneous processing of other items at the same time without detriment to processing performance.

Flexibility in working memory. While the previous section highlights the capacity limitations of the working memory system, the current data also highlight another feature: flexibility. When considering the current data as an indication of the flexibility of the working memory system, it also makes sense to consider other ways in which flexibility has been demonstrated. Previous task switching work using varied numbers of stimulus-response mappings is relevant to the current work (Hübner et al, 2004). In their experiment, an imbalanced stimulus-response mappings condition (2 mappings for one task vs. 4 mappings for the other task) was compared to a constant stimulus-response mappings condition (4 mappings for both tasks). Results indicated that there were larger switch costs for the task with fewer mappings, and that participants took longer to respond when there were more mappings. These findings were replicated in Experiment 3. In their work as well as the current experiment, participants benefitted more from repetitions when there were fewer total stimulus-response mappings. In Experiment 3, however, owing to the 2-4 and 4-2 condition separation, we can directly compare switch costs when the total number of mappings is constant, but there is more information in the bridge or more information in activated long term memory. In this case, there were much larger switch costs in the 2-4 condition than the 4-2 condition (interaction between 2-4 and 4-2 condition and task transition was significant, $F(1,39) = 26.65$, $p < .001$, $\eta^2_{\text{partial}} = .406$). The nature of this change is that participants lost the repetition benefit in the 4-2 condition. This effect likely occurred because when there were four stimulus-response options to represent, there is a less strong instantiation of that set overall. In that case, repetitions are not as beneficial because there is less persisting activation. In terms of the computational model, this may be implemented in the pattern of activation between the input and output layers. As stimulus-response mappings become more numerous, each of the individual connections becomes weaker (Oberauer, 2007). Therefore even after a response is selected, it takes less competition from other bindings to reduce that activation back to baseline, and the typical benefit of residual activation diminishes.

Additionally, there has been task switching work varying the total number of tasks, rather than the number of S-R mappings (Buchler, et al., 2008; Kleinsorge & Apitzsch, 2012; Kleinsorge, Heuer & Schmidtke, 2004; Kleinsorge & Scheil, 2013). In the work of Kleinsorge and Scheil (2013), participants switched between two tasks and four tasks in an online manner, depending on the cue given to them pre-trial. The cue indicated which two of the four tasks would be relevant for that particular trial. Consistent with the other task switching studies referenced above, Kleinsorge and Scheil found that when the pre-trial cue restricted task selection to two tasks only on a switch trial, there was performance facilitation. When the cue restricted the tasks on a repetition trial, there was no influence on performance. The authors explained this pattern of data by indicating that when switching tasks, it matters a great deal whether there is only one other option to perform, or whether there are multiple options competing for selection. When switching among two tasks, the act of switching itself provides a cue for the other task, because that is the only other option for performance. Therefore as soon as a switch is indicated, preparation can begin because the task is known. When switching among four tasks, there is no way to know for certain which task will be required just by knowing there is a switch, so every element of task preparation must occur in the moment, which slows performance. For repetition trials, the task set remains instantiated so how many other tasks there are is irrelevant.

Together with the current experiment, these studies give evidence for a flexible working memory system that can reconfigure between different tasks and different mental sets in an online manner depending on environmental constraints. In Experiment 3, participants were successful in the critical condition in that they were able to flexibly switch between different tasks with different numbers of stimulus-response mappings. While performance is very slightly impaired when there are more possible mappings (i.e. slower but not significantly slower RTs in the 2-4 condition compared to the 2-2 condition), the size of the unused mapping does not influence performance—only the contents of the bridge, or the currently relevant information. This finding is consistent with Hübner, et al., (2004), and Kleinsorge and Schiel (2013), which together suggest that the working memory and cognitive control systems dynamically change depending on the nature of the required activity. This mechanism supports the ability to maintain large amounts of information in the background of the mind while actively prioritizing other information for more detailed processing.

Interim conclusions. Overall the results of Experiment 3 are wholly consistent with Oberauer's working memory model. There is evidence for analogous processing in declarative and procedural working memory when considering the effects of load in long term memory. The capacity limitations of the bridge appear to be different than the capacity limitations of activated long term memory.

Furthermore, in the current experiment, task memory showed a different pattern of effects from target RT, just as seen in Experiment 1. This finding supports the previous interpretation that memory for items or sequences measuring a distinct aspect of task span performance from target RT. Potentially, task and list memory results from successful implementation of declarative working memory only, and target RT results from successful implementation of procedural working memory. If this hypothesis is supported by future research, it would be evidence for the independence of the two subsystems. This idea will be discussed further in the general discussion.

General Discussion

The current work examined Oberauer's (2009) model of working memory using the task span and memory span paradigms. The task span was first established as a test of procedural working memory in Experiment 1. Then Experiments 2 and 3 tested the assumption of a functional analogy between the declarative and procedural working memory subsystems in two domains. The results of three experiments largely supported Oberauer's model. Experiment 1 provided evidence that the procedural and declarative subsystems are separable, as loading each system independently produced distinct effects. Experiment 2 tested the state of previously activated information in procedural and declarative working memory, but was inconclusive. Experiment 3 tested the capacities of the bridge and activated long term memory, and found evidence for a lack of set-size effects in activated procedural long term memory, just as was previously found in activated declarative long term memory (Oberauer, 2002).

The work presented in this dissertation speaks to the questions of working memory organization that Oberauer's (2009) model proposes to answer. Oberauer's model is different from those that preceded it for several reasons. For one, it is not domain specific like Baddeley and Hitch's (1974) model, but is instead divided based on two types of representations, items or procedures. The model also indicates that representations are activated from within long term memory—i.e. it is an embedded model which sets it apart from at least Baddeley and Hitch (1974). The successive levels of activation result in a highly specific focus of attention, which is selected from an intermediary store, which is in turn selected from an activated portion of long term memory (e.g. Marklund & Nyberg, 2007). Oberauer suggests that the declarative and procedural subsystems are functionally analogous and independent. The current work has three main contributions: For one, it introduces a new methodological approach to the study of this subsystem division. Next, there is support for at least one aspect of the functional analogy between declarative and procedural working memory. Finally, the data provide some indication that the subsystems may also be independent.

Methodological Contributions. The task span procedure was originally designed as a method to marry the working memory demands of complex span tasks with the set shifting requirements of task switching as a test of executive control (Logan, 2004). Logan compared the length of span participants could remember and accurately perform to a simple memory span and found that there was no difference in span length for the two conditions. He discussed the role of long term memory in span performance, suggesting that the sequences are outsourced to long term memory for storage and then retrieved in chunks to be performed. Ultimately, Logan (2004) asserts that there is no trade-off apparent between processing and storage. That is, the same number of tasks can be remembered as can be remembered and performed. This finding suggests that the storage and processing elements of working memory may be distinct, and the current work supports this interpretation. Based on the different effects of declarative and procedural load in Experiment 1, and the different patterns among memory and RT data, storage and processing may rely on different elements of working memory. These different elements appear to map the declarative and procedural elements Oberauer (2009) outlines. Declarative working memory may support memory or storage, and procedural working memory may support manipulation

Based on the contribution of the current dissertation, the task span is established as a method to test the mechanisms of procedural working memory according to Oberauer's model. The current work takes a complimentary approach to that of Oberauer and his group in examining the two working memory subsystems. Typically in examining the procedural subsystem, Oberauer has used the task switching paradigm and then designed a declarative task to be comparable (list switching). Here, a classic declarative task (memory span) is modified with procedural elements to yield the task span. This paradigm has several advantages that make it a good choice for testing procedural working memory. For one, the task span has clear procedural elements with the incorporated task switching requirements. Participants must keep multiple tasks active in a way that they can be rapidly accessed depending on the demands of the environment. The task span and memory span also have declarative elements with regard to the task names that must be kept in memory in a sequential order. The task span requires input from the entire working memory system—maintenance of task names, updating of position information as tasks are carried out, and processing of stimuli and appropriate responses. This advancement of the paradigm will be useful in testing the procedural aspects of Oberauer's model, to extend beyond task switching data to a more dynamic procedure.

Refining the construct of working memory. A critical distinction can be made in working memory theories to separate multi-component storage based systems from more dynamic state-based theories (D'Esposito & Postle, 2015). Multi-component theories of working memory, like that of Baddeley and Hitch (1974), include the ability to buffer and store information, but do so in a very segmented way. State-based theories, in contrast, allow for a more dynamic role of attention in the working memory system. Models such as those of Cowan (1988) and Oberauer (2009), assume embedding of working memory within long term memory, with attention serving as a selection mechanism for particular pieces of information. This prioritization of attended information explains the capacity limitations inherent to working memory, as well as interference. The current trend in working memory research is dominantly heading towards adoption of state-based theories (D'Esposito & Postle, 2015).

Oberauer's (2009) model, as mentioned above, involves successively increasing levels of activation for items ranging from large categories of information in an activated portion of long term memory to a single item or response in the focus of attention or response focus. Critically, these different components represent not a transfer between “containers” but the same information with more or less activation dedicated to it (Oberauer, 2013). The intermediary stores, the bridge and region of direct access, play a critical role in one of the most influential hallmarks of the working memory system: the ability to flexibly manipulate content. In the intermediary stores, a small subset of items bound to cues or serial positions, or a single task set involving multiple S-R mappings, is represented in a way that can be rapidly adjusted. The intermediary store serves to keep several responses or items activated for selection in a state that is less prioritized than ultimately will be the case for selected information. The bindings in the intermediary store allow for this temporary maintenance, but the capacity concept is more complicated than just thinking about a number of slots for maintenance.

Oberauer (2013) suggests that the commonly used metaphor of working memory as a series of containers is helpful in the abstract early stages of developing theories, but has become harmful in developing a fully satisfying account of capacity. Such a discussion of the container metaphor leads to questions of exactly how many items can be “held” by any given component at once. Instead, Oberauer (2013) focuses on the nature of bindings and *mechanisms* of storage and manipulation. With that being said, the current data do not directly indicate that the working memory system is not a “container model”. The data can, however, be interpreted in a way that is consistent with that idea. In particular, in Experiment 3 participants were able to rapidly shift between task sets of varying sizes. Although capacity was of course limited in Experiment 3, it may not have been inherently dependent on the actual number of mappings within task sets (which would be like a certain number of “slots”). Instead, capacity is based on the characteristics of information currently relevant for the task, and this information could be of any type. For example, with more complex bindings, fewer can be represented. While performance does change with the size of task sets, the working memory system itself is able to accommodate varying numbers and types of context-content bindings. If we focus on working memory as a series of embedded containers, it would be easy to stagnate on the question of a task set's or the bridge's maximum or specific capacity, rather than focusing on the actual mechanism of storage and manipulation. Overall, capacity is limited, but this limitation is flexible depending on the nature of bindings.

Thinking about this coordination of storage and manipulation, on the very first page of this dissertation, the construct of working memory is defined as: "a dynamic cognitive system that allows for the temporary storage of some information while simultaneously processing other information". This definition was introduced as a guiding conception for the current work, but it is not universally accepted. For example, in a recent paper from Cowan's lab working memory is defined as "a system dedicated to maintaining a small amount of information over a brief period of time" (Vergauwe & Cowan, 2014, first page), while Baddeley's group defined working memory as "the ability to maintain information in the face of potential distraction" (Jarrold, Tam, & Baddeley, 2011, p. 688). These definitions are mainly storage focused, and fail to emphasize the dynamic processing power working memory actually has. The definition presented in the current dissertation represents a way of thinking about working memory that is in line with Oberauer's model and highlights the dynamic relationship between storage and processing and how they interact. Findings of flexibility in the current experiments provide more support for this way of defining the construct, although as mentioned above, this flexibility is coupled with limitations in capacity. Differences in definition are not merely matters of wording; they represent the disparate mechanisms underlying the authors' foundations of how working memory functions. Based on the current experiments using the task span and other work, it is possible that processing and maintenance can be completed simultaneously—and the model accounts for this possibility (Buchler, et al., 2008; Oberauer & Gothe, 2006). These two mental actions (processing and maintenance) are inextricably linked in Experiment 3—participants must shift between processing in the bridge and maintenance in activated long term memory on a moment to moment basis (although processing is capacity limited in the intermediary store). The working memory system can certainly perform this dynamic shifting process, so therefore a model of working memory must be able to account for this ability as well, along with the limitations in capacity. Oberauer's model appears to align with these requirements.

As discussed in the introduction, Oberauer (2009) developed his model of working memory in accordance with six principles he deemed necessary for function in that system. As a reminder, these principles were: 1. the ability to create mental representations and maintain them. 2. The ability to selectively manipulate one or some representations. 3. The flexibility to adapt to new and changing goals. 4. The ability to rapidly update previously created representations. 5. The ability to draw the contents of working memory from long term memory. 6. The ability to store representations permanently in long term memory. As the current data support elements of his model, they are also largely in accordance with these principles, particularly numbers one, three, and five. In the described work as well as hundreds of other studies using task switching and similar paradigms, participants were able to create temporary representations of task and memory sets in the experimental context. Participants were also able to switch flexibly between these different task sets depending on what was currently relevant to performance. Even when task sets were different sizes, flexible switching was still possible. These task sets were also stored in activated long term memory when not being used but integrated into the intermediary store when they were relevant. Therefore, the current data support the conclusion that Oberauer's model is in alignment at least with part of the requirements outlined for a successful working memory system that were tested in the current work.

Exploring results in the context of the computational model of Oberauer's three embedded components framework. As mentioned in the discussions of each experiment, the current data can be well-accounted for by the computational model of Oberauer's framework, which can provide a framework for evaluating the theoretical model. Experiment 1 demonstrated separate declarative and procedural systems as well as the nature of complexity in input-output bindings. Experiment 3 demonstrates the different set size effects for items in activated long term memory and the bridge, which supports the distinction between these two components in the working memory model. The computational model would predict these findings.

Oberauer and colleagues' (2013) computational model is also consistent with a long history of theorizing about how executive control is implemented in task switching—which is inherently relevant to the execution of the task span. In particular, researchers have suggested that when switching between competing tasks, individuals must first activate the relevant goal, and then ready themselves for performance (e.g. Allport & Wylie, 2000; Rubinstein, Meyer, & Evans, 2001). In the computational model, activating the relevant goal could mean engaging relevant representations in activated long term memory. This could happen both by connecting the information present in the cue layer to appropriate memory or task sets, and by setting the candidate layer to represent possible memory items or responses. Performance readiness comes from three elements: first, the acquisition of S-R bindings, second, the activation of particular subsystems based on cues, and third, the suppression of irrelevant or competing domains (Allport & Wylie, 2000). These steps can also be accounted for in Oberauer's computational model, making the mechanisms of implementing procedures in working memory harmonious with long standing task switching theory. Acquiring S-R bindings, activating particular subsystems, and suppressing irrelevant ones can all be represented in the bridge or region of direct access. In particular the input-output bindings are responsible for manifesting S-R bindings (the first element) and providing the context (the second element) to select the appropriate one in the face of competition (the third element). Of course the computational model was not designed to account for executive control in task switching, but the fact that it is consistent with the ideas of other longstanding theoretical frameworks is promising for its ability to account for future data, as well as support for Oberauer's approach of using task switching as a paradigm to examine procedural working memory.

Oberauer's computational model can also explain why working memory capacity is limited. A long history of research in the working memory domain makes it clear that one of the most well supported working memory characteristics is its limited capacity (Richardson, et al., 1996). The computational model manages this capacity limit in the expression of bindings between the input and output layers. The bindings between input and output must be limited. There must be some semblance of separation between bindings in order to allow for selection of one without overwhelming competition from all others (Oberauer, 2007). When more candidates are available for selection, access to one particular option is slowed down, which limits the functionality of the working memory system. Limiting the bindings produces less competition and therefore more functionality, but ultimately reduces the complexity of representations that can be maintained and processed resulting in capacity limits (Oberauer, 2007). The limit on input-output bindings is not rigid, however (there is no defined maximum of 4 item-context bindings for example). The number of bindings the working memory system can have while still remaining functional will vary based on task or individual characteristics (Oberauer & Eichenberger, 2013).

Compatibility with other working memory models. Oberauer's model of working memory is a significant departure from those of his predecessors. To recap beyond the simple definitions mentioned above, the Baddeley and Hitch (1974) model makes a critical distinction between auditory and visual information, and posits a central executive that allocates attention to domain-specific storage systems, and a new episodic buffer that can integrate information with long term memory (Baddeley, 2000). Cowan (1988; 1995) proposes an embedded model, with a focus of attention highlighting specific information from long term memory for processing, along with a central executive piece that can manipulate those focused representations. The current data cannot be accounted for by either of those models without significant modification to them.

Primarily, Experiment 1 provides clear evidence for a distinction between the declarative and procedural subsystems, and Experiment 3 for the analogy between them. Neither Baddeley's nor Cowan's model can account for such a distinction in their present forms. Baddeley's model specifically outlines domain specificity in another way (Baddeley & Hitch, 1974; Baddeley, 2000), and Cowan's (1988; 1995) model specifically excludes domain specificity. Both theories of working memory would require significant modification in order to include a distinction between declarative and procedural information in the way for which evidence is presented in the current work. However, Baddeley's and Cowan's models both do include mechanisms for contributions of executive processing, i.e. the central executive. This can sometimes be thought about in a similar way to procedural information (Conway, et al., 2008), but the two are not the same. The central executive performs manipulation and coordination of multiple sources of information in Baddeley's and Cowan's models—it is not described as being responsible for the storage of procedural processes per se. Procedural working memory (Oberauer, 2009) is described as storing and selecting the processes and information for manipulation of content separately from the executive mechanisms. As such, the procedural working memory subsystem cannot be completely accounted for by the central executive. Oberauer also still maintains that there must be executive processing that comes online to allow for the communication between the declarative and procedural subsystems (personal communication, March 12, 2015). But there would not be a place in Oberauer's model for an analogy between declarative information and the central executive, because the central executive has a different role in his view. Since we know that there are examples of declarative and procedural working memory functioning analogously in list and task switching, like switch costs, reduction in switch costs with increased preparation time, and mixing costs, Baddeley's and Cowan's models are incompatible with the current evidence.

Only Oberauer's model of working memory would predict the analogous effects between declarative and procedural information in Experiment 3. Oberauer (2002) discusses how the lack of set size effects in activated declarative long term memory could be accounted for with Baddeley's model. Potentially, one could say that the active list (for which set size is relevant) is kept in the central executive, and the passive list (for which set size is irrelevant) is kept in the phonological loop. The central executive would store the active list, so that manipulation of that list would be possible. But, significant storage capacity is not really a feature that the central executive is expected to have. Additionally, in order to account for the effects of Oberauer (2002) there would need to be an introduction of a focus of attention-like element to select one item from the active list for processing, maintaining separation from other items in the active list.

Baddeley’s model does not formally account for this ability. While this organization is possible, and Baddeley’s model could potentially account for the current data with considerable stretching of parsimony, Baddeley’s model does not *predict* these effects, as Oberauer’s does.

If the same allocations as just described for declarative information were to be used for procedural information like that of Experiment 3, the active task set would be processed in the central executive and the passive task set in the phonological loop (or potentially the episodic buffer). Therefore there would need to be the possibility of task set rehearsal in the phonological loop, since rehearsal is the mechanism for maintenance in Baddeley’s model (Baddeley & Hitch, 1974). This mechanism of maintenance seems unrealistic for task sets in the task span, considering that sequences also need to be rehearsed and would create competition and interference with the task set elements. If the phonological loop is not used for storage of passive procedural task sets, then there would perhaps need to be a newly defined element or outsourcing of the rehearsal to another storage buffer to allow for this maintenance. If the declarative and procedural information are handled so differently, then it becomes hard to explain the analogous findings between these two types of information. Consequently, Baddeley’s model seems unable to handle the current findings, or at least it is an un-parsimonious approach to accounting for them. While ruling out Baddeley’s model may not be possible considering that the elements of his model are quite flexible, the current data raise questions about whether Baddeley’s model is still suitable for the current state of the working memory field.

Cowan's model could account for the active list being maintained in the most active level of working memory—the focus of attention—and the passive list being maintained in activated long term memory. However, in Cowan's focus of attention, multiple items are held at a high level of activation. There must be some ability to prioritize one item or response over others in order to account for repetition benefits such as those seen in task and list switching (Rogers & Monsell, 1995; Souza, et al., 2012). Therefore, Oberauer's model is the best way to account for the present data. Storage and processing can work separately according to this model, which is consistent with the results of this dissertation.

Future research. Future research should specifically address the question of functional independence in Oberauer's model. The current work speaks to the idea of a functional analogy, but this second main assumption that Oberauer outlines was not specifically tested in these experiments. However the current data might point to one way in which these systems are independent despite the fact that that the studies were not intentionally designed to test that assumption. In both Experiments 1 and 3, there were distinctly different data patterns for memory and RT results. This disparity suggests that these two dependent measures are drawing on different cognitive processes. Based on a consideration of the computational model of Oberauer's model (Oberauer, et al., 2013), it is possible that in these paradigms, memory performance is based on successful implementation of the declarative subsystem while target RT performance is based on successful implementation of the procedural subsystem. If the two subsystems were not independent, there might not be as clear a distinction between performance on these two measures. Also consistent with this idea of functional independence, there were distinctly different effects of the declarative and procedural load on the memory span in Experiment 1, suggesting that these two types of information are processed differently. Therefore they could be separate and independent. These findings are not direct evidence for independence, but they are consistent with that idea and suggest that the question should be further explored.

Additionally, as mentioned in the discussion of Experiment 2, there are several modifications to that experiment that could result in the ability to draw inferences from the memory and task span results with respect to the question of how previously used information is maintained. For one, modifications of the number of tasks could allow for better operational definitions of the unfocused and defocused trials, without adding the complexity of a hand switch in addition to a task switch. There also could be selective probing of sequences for recall or performance, rather than completely sequential performance, to create a scenario more comparable to Rerko and Oberauer (2013). Finally, modifications to the nature of the declarative tasks can be undertaken in two ways. The retro-cuing paradigm could be adapted to require a response for memory items (rather than just activation) before moving on to additional items in a way more similar to the memory span. If the defocused benefit disappears for retro-cuing in this case, there would be support for the idea that making a response to items produces some sort of resetting of the input-output bindings in the region of direct access. This resetting could cause an elimination of residual activation that would ordinarily provide a benefit in response time—explaining the null effects in Experiment 2. Similarly, the memory span can be modified to a recognition task instead of recall to reduce procedural demands which might interfere with the ability to achieve a defocused-trial benefit.

Lastly, as mentioned above, there is very little discussion in Oberauer's body of work about how declarative and procedural working memory interact and communicate. There must be communication between the two subsystems; individuals need declarative information on which to perform procedures, and procedures must be performed on some declarative mental object. In fact, in the examination of the task span in the context of the computational model, the iteration of the procedural subsystem required input from the declarative subsystem in order to complete processing. However, Oberauer and his colleagues are silent in published work on a specific mechanism for this communication. This critical element of a functional working memory system must be identified and tested.

With some limitations in mind, the current work can be used as a foundation for future studies examining these additional areas. Through the examination of cognitive processes in the task span, we can develop a more complete understanding of the working memory system. This understanding has relevance more broadly for cognition; working memory is potentially the foundation of individual variation in a large array of cognitive processes. For example, individual capacity in the working memory system is related to general fluid intelligence (Conway, Kane, & Engle, 2003), reading comprehension (Daneman & Carpenter, 1980), problem solving (Wiley & Jarosz, 2012), planning (Reiman, & Arrington, in prep.), reasoning (Engle, Tuholski, Laughlin, &

Conway, 1999; Kane, Hambrick, Tuholski, Wilhelm, Payne, & Engle, 2004) and language processing (King & Just, 1991). If we have a better understanding of the mechanisms of working memory, we may be able to more completely understand how variation in this system is associated with such a broad swath of cognitive abilities, leading to greater understanding of those domains as well (Smith & Kosslyn, 2006). Therefore the exploration of working memory models, as in the current work, can have a substantial impact on the field of cognitive psychology more generally.

Conclusions

The current work tested Oberauer's three embedded components model of working memory. Working memory is a highly studied topic, and the number of distinct models about its organization, including Oberauer's, speaks to the complexity of the system. This dissertation solidifies one of the newest models by testing one of its main assumptions. Many of the predictions made at the conception of these three experiments suggested that the assumption of a functional analogy between declarative and procedural working memory would not hold true. However, Experiment 1 provided the groundwork for discussing independence between the two subsystems, and Experiment 3 provided support for the idea that they are functionally analogous in terms of the nature of capacity in the intermediary store and activated long term memory.

This project is also empirically valuable in connecting an experimental paradigm never before used to test a particular model of working memory with a model gaining popularity in the working memory field. The task span and memory span have considerable empirical advantage over the list and task switching paradigms used by Oberauer and colleagues, and this advantage has not been previously utilized. Developing these empirical approaches as valid tests of declarative and procedural working memory is useful in furthering the theoretical and empirical grounding of this working memory model.

In conclusion, the results of this dissertation speak to the claim of a functional analogy between declarative and procedural working memory. The comparisons tested in this project were potentially the areas most likely to show differences between declarative and procedural working memory based on past research in related domains, so failures to find differences in these studies is good support for Oberauer's model.

References

Allport, A. & Wylie, G. D. (2000). Task switching, stimulus-response bindings, and negative priming. In S. Monsell & J. Driver (Eds). *Control of Cognitive Processes, Attending and Performance XVIII* (pp. 35-70). London: A Bradford Book.

Arrington, C. M. & Logan, G. D. (2004). Episodic and semantic components of the compound –stimulus strategy in the explicit task-cuing procedure. *Memory & Cognition, 32,* 965-978.

Baddeley, A.D. (2000). The episodic buffer: A new component of working memory? *Trends in Cognitive Science, 4,* 417-423.

Baddeley, A. D., (2010). Long-term and working memory: How do they interact? In L. Backman, (Ed.), *Memory, Aging, and the Brain: A Festschrift in honour of Lars-Göran Nilsson*. Hove: Psychology Press.

Baddeley, A., Chincotta, D., & Adlam, A. (2001). Working memory and the control of action: Evidence from task switching. *Journal of Experimental Psychology: General, 130,* 641-657.

Baddeley, A.D., & Hitch, G.J. (1974). Working memory. In G.A. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (pp. 47-89). New York: Academic Press.

Buchler, N. G., Hoyer, W. J., & Cerella, J. (2008). Rules and more rules: The effects of multiple tasks, extensive training, and aging on task switching performance. *Memory and Cognition, 36,* 735-748.

Bunting, M.F., & Cowan, N. (2005). Working memory and flexibility in awareness and attention. *Psychological Research, 69*, 412-419.

Conway, A. R. A., Kane, M. J. & Engle, R. W. (2003). Working memory capacity and its relation to general intelligence. *Trends in Cognitive Sciences, 7,* 547-552.

Conway, A. R. A., Moore, A. B., & Kane, M. J. (2009). Recent trends in the cognitive neuroscience of working memory. *Cortex, 45,* 262-268.

Cowan, N. (1988). Evolving conceptions of memory storage, selective attention, and their mutual constraints within the human information processing system. *Psychological Bulletin, 104,* 163-191.

Cowan, N. (1995). *Attention and memory: an integrated framework.* New York: Oxford University Press.

Cowan, N. (1999). An embedded-processes model of working memory. In A. Miyake & P. Shah (Eds.), *Models of Working Memory: Mechanisms for Active Maintenance and Executive Control,* Cambridge: Cambridge University Press.

Cowan, N. (2005). The focus of attention as observed in visual working memory tasks: Making sense of competing claims. *Neurospychologica, 49,* 1401-1406.

D'Esposito, M. & Postle, B. R. (2015). The cognitive neuroscience of working memory. The *Annual Review of Psychology, 66,* 115-142.

Dade, L. A., Zatorre, R. J., Evans, A. C., Jones-Gottman, M. (2001). Working memory in another dimension: functional imaging of human olfactory working memory. *Neuroimage, 14*, 650-660.

Daneman, M. & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning & Verbal Behavior, 19,* 450-466.

De Jong, R. (2000). An intention-activation account of residual switch costs. In S. Monsell & J. Driver (Eds.) *Control of Cognitive Processes: Attention and Performance XVIII (*pp. 357-376). Cambridge: MIT Press.

Dreisbach, G., Goschke, T., & Haider, H. (2007). The role of task rules and stimulus-response mappings in the task switching paradigm. *Psychological Research, 71,* 383-392.

Engle, R. W., Tuholski, S. W., Laughlin, J. E., & Conway, A. R. A. (1999). Working memory, short-term memory, and general fluid intelligence: A latent variable approach. *Journal of Experimental Psychology: General, 128,* 309-331.

Ericsson, K. A. & Kintsch, W. (1995). Long-term working memory. *Psychological Review, 102,* 211-245.

Gade, M., Druey, M.D., & Oberauer, K. (2012, November). Declarative and procedural working memory—Two separate systems? Poster presented at the annual meeting of the Psychonomic Society. Seattle, Washington.

Garavan, H. (1998). Serial attention within working memory. *Memory & Cognition, 26,* 263-276.

Gilbert, S. J., & Shallice, T. (2002). Task switching: A PDP model. *Cognitive Psychology, 44,* 297-337.

Gilchrist, A., & Cowan, N. (2011). Can the focus of attention accommodate multiple, separate items? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 37,* 1484-1502.

Harris, J. A., Miniussi, C., Harris, I. M., & Diamond, M. E. (2002). Transient storage of a tactile memory trace in primary somatosensory cortex. *Journal of Neuroscience, 22,* 8720-8725.

Hübner, M., Dreisbach, G., Haider, H., & Kluwe, R. H. (2003). Backward inhibition as a means of sequential task-set control: Evidence for reduction of task competition. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29,* 289-297.

Hübner, M., Kluwe, R. H., Luna-Rodriguez, A., & Peters, A. (2004). Response selection difficulty and asymmetrical costs of switching between tasks and stimuli: No evidence for an exogenous component of task-set reconfiguration. *Journal of Experimental Psychology: Human Perception and Performance, 30,* 1043-1063.

Hurlstone, M. J., Hitch, G. J., & Baddeley, A. D. Memory for serial order across domains: An overview of the literature and directions for future research. *Psychological Bulletin, 140,* 339-373.

Jarrold, C., Tam, H., & Baddeley, A. D. (2011). How does processing affect storage in working memory tasks? Evidence for both domain-general and domain-specific effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 37,* 688-705.

Kane, M. J., Hambrick, D. Z., Tuholski, S. W., Wilhelm, O., Payne, T. W., & Engle, R. W. (2004). The generality of working memory capacity: A latent-variable approach to verbal and visuospatial memory span and reasoning. *Journal of Experimental Psychology: General, 133,* 189-217.

King, J. & Just, M. A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language, 30,* 580-602.

Kleinsorge, T., & Apitzsch, N. (2012). Task preparation based on precues versus memory: precues l;ead to superior performance with two tasks but not with four tasks. *Journal of Cognitive Psychology, 24,* 140-156.

Kleinsorge, T., Heuer, H., & Schmidtke, V. (2004). Assembling a task space: Global determination of local shift costs. *Psychological Research, 68,* 31-40.

Kleinsorge, T. & Scheil, J. (2013). Task switching among two or four tasks : effects of a short-term variation on the number of candidate tasks. *Psychological Research, 79,* 163-173.

Lien, M-C., Ruthruff, E., & Kuhns, D. (2006). On the difficulty of task switching: Assessing the role of task-set inhibition. Psychonomic Bulletin & Review, 13, 530-535.

Logan, G. D. (2002). Parallel and serial processes. In H. Pashler & J. Wixted (Eds.), *Stevens' Handbook of Experimental Psychology (3rd ed.), Vol. 4: Methodology in experimental psychology*. (pp. 271-300). New York, NY: John Wiley & Sons.

Logan, G.D. (2004). Working memory, task switching, and executive control in the task span procedure. *Journal of Experimental Psychology: General, 133,* 218-236.

Logan, G. D. (2007). What it costs to implement a plan: Plan-level and task-level contributions to switch costs. *Memory & Cognition, 35,* 591-602.

Logie, R. H. (1995). *Visuo-spatial working memory.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Marklund, P. & Nyberg, L. (2007). Intersecting the divide between working memory and episodic memory. Evidence from sustained and transient brain activity patterns. In N. Osaka, R. H. Logie, & M. D'Esposito (Eds). *The Cognitive Neuroscience of Working Memory* (pp.305-331). Oxford: Oxford University Press.

Mayr, U. & Keele, S. W. (2000). Changing internal constraints on action: The role of backward inhibition. *Journal of Experimental Psychology: General, 129,* 4-26.

McClelland , J. L. (1979). On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86*, 287-330.

Mecklinger, A., Gruenwald, C., Besson, M., Magnie, M-N., Von Cramon, Y. (2002). Separable neuronal circuitries for manipulable and non-manipulable objects in working memory. *Cerebral Cortex, 12,* 1115-1123.

Meiran, N. & Kessler, Y. (2008). The task-rule congruency effect in task switching reflects activated long-term memory. *Journal of Experimental Psychology: Human Perception and Performance, 34,* 137-157.

Miller, G. A. (1956). The magical number seven, plus or minus two. Some limits of our capacity for processing information. *Psychological Review, 63,* 81-97.

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). Control of the contents of working memory—A comparison of two paradigms and two age groups. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31,* 714-728.

Oberauer, K. (2007). Activation, binding, and selective access. In N. Osaka, R. H. Logie, & M. D'Esposito (Eds.) *The Cognitive Neuroscience of Working Memory* (pp. 351-368). Oxford: Oxford University Press.

Oberauer, K. (2009). Design for a working memory. *Psychology of Learning and Motivation, 31,* 45-100.

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

Oberauer, K. (2013). The focus of attention in working memory—from metaphors to mechanisms. *Frontiers in Human Neuroscience, 7,* 1-16.

Oberauer, K. & Eichenberger, S. (2013). Visual working memory declines when more features must be remembered for each object. *Memory and Cognition, 41,* 1212-1227.

Oberauer, K. & Gothe, K. (2006). Dual-task effects in working memory: Interference between two processing tasks, between two memory demands, and between storage and processing. *European Journal of Cognitive Psychology, 18,* 493-519.

Oberauer, K. & Lewandowsky, S. (2011). Modeling working memory: A computational implementation of the time-based resource-sharing theory. *Psychonomic Bulletin & Review, 18,* 10-45.

Oberauer, K., Souza, A. S., 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.

Postle, B. R. (2006). Working memory as an emergent property of the mind. *Neuroscience, 139,* 23-28.

Postle. B, R. & D'Esposito, M. (2003). Spatial working memory activity of the caudate nucleus is sensitive to frame of reference. *Cognitive Affective and Behavioral Neuroscience, 3,* 122-144.

Reiman, K. M., & Arrington, C. M. (n.d.). Planning tasks in sequences: The contributions of working memory and executive control. Manuscript in preparation.

Reiman, K. M., Weaver, S. M. & Arrington, C. M. (2015). Encoding and choice in the task span paradigm. *Psychological Research, 79,* 267-281

Rerko, L., & Oberauer, K. (2013). Focused, unfocused, and defocused information in working memory. *Journal of Experimental Psychology, Learning, Memory, and Cognition, 39,* 1075-1096.

Rerko, L., Souza, A. S., Oberauer, K. (2014). Retro-cue benefits in working memory without sustained focal attention. *Memory and Cognition,* Advance online publication.

Richardson, J. T. E., Engle, R. W., Hasher, L., Logie, R. H., Stolzfus, E. R. & Zacks, R. T. (1996). *Working memory and human cognition.* Oxford: Oxford University Press.

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

Rogers, R.D., & Monsell, S. (1995). Costs of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: General, 124,* 207-231.

Rubin, O, & Meiran, N. (2005). On the origins of the task mixing cost in the cuing task-switching paradigm. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31,* 1477-1491.

Rubinstein, J., Meyer, D. E., & Evans, J. E. (2001). Executive control of cognitive processes in task switching. *Journal of Experimental Psychology: Human Perception and Performance, 27,* 763-797.

Schneider, D. W., & Verbruggen, F. (2008). Inhibition of irrelevant category-response mappings. *Quarterly Journal of Experimental Psychology, 61,* 1629-1640.

Smith, E. E., & Jonides, J. (1999). Storage and executive processes in the frontal lobes. *Science, 283,* 1657-1661.

Smith, E. E., Jonides, J., Koeppe, R. A., Awh, E., Schumacher, E. H., & Minoshima, S. (1995). Spatial versus object working memory: PET investigations. *Journal of Cognitive Neuroscience, 7,* 337-356.

Smith, E. E. & Kosslyn, S. M. (2006). Working memory. In *Cognitive Psychology: Mind and Brain* (pp. 239-279). Pearson: New York, NY.

Smyth, M. M. & Pendleton, L. R. (1989). Working memory for movements. *The Quarterly Journal of Experimental Psychology A: Human Experimental Psychology, 41,* 235-250.

Souza, A.S., Oberauer, K., Gade, M., & Druey, M. D. (2012). Processing of representations in declarative and procedural working memory. *Quarterly Journal of Experimental Psychology, 65,* 1006-1033.

Souza, A. S., Rerko, L., & Oberauer, K. (2014, April 14). Unloading and Reloading Working Memory: Attending to One Item Frees Capacity. *Journal of Experimental Psychology: Human Perception and Performance*, advanced online posting ahead of print.

Squire, L. R. (2004). Memory systems of the brain: a brief history and current perspective. *Neurobiology of Learning and Memory, 82,* 171-177.

Vergauwe, E. & Cowan, N. (2014). Attending to items in working memory: evidence that refreshing and memory search are closely related. *Psychonomic Bulletin & Review,* advanced online posting ahead of print.

Wiley, J., & Jarosz, A. E. (2012). How working memory capacity affects problem solving. In B.H. Ross (Ed). *The Psychology of Learning and Motivation, Vol. 56 (*pp. 185-227). San Diego: Elsevier Academic Press.

Appendix A

Additional analyses and figures—Experiment 1

Experiment 1a accuracy.

Participants' accuracy was calculated as their correct categorizations for the correct task in the task span. For the memory span, accuracy is the same measure as memory, so the list memory measure is a reflection of accuracy. Task accuracy data for the task span are depicted in figure A1. There was no significant difference in accuracy in the task span as a function of switch frequency, as evidenced by a repeated measures ANOVA, $F(4, 60) = 2.21$, $p = .158$. These accuracy data support the interpretation that there was no speed/accuracy trade-off for participants in the task span, as switch frequency did not influence accuracy.

Experiment 1b accuracy.

Again, participants' accuracy was calculated as their correct categorizations for the correct task in the task span. Task accuracy data for the task span are depicted in figure A1. There was a significant difference in accuracy in the task span as a function of switch frequency, as evidenced by a repeated measures ANOVA, $F(4, 60) = 15.59$, $p = .001$, $\eta^2_{partial} = .510$. Participants' accuracy was best for the highest and lowest switch frequency sequences. These accuracy data support the interpretation that there was no speed/accuracy trade-off for participants in the task span, as the effects of switch frequency on accuracy was consistent with the effect of that variable on RT.

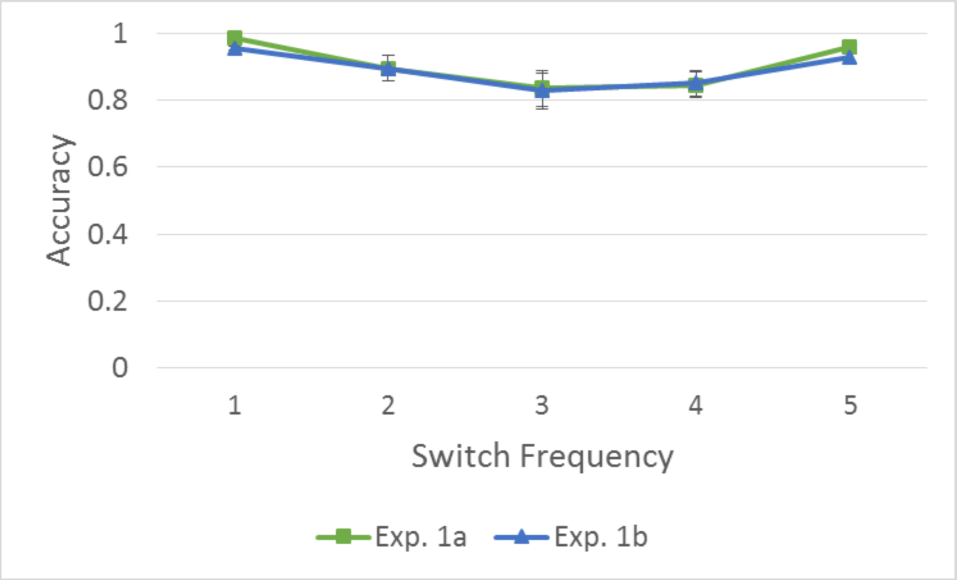

Figure A1. Mean accuracy for items in Experiments 1a and 1b as a function of switch frequency for the task span (TS) condition.

Appendix B

Additional analyses and figures—Experiment 2

Experiment 2 task memory.

Participants' memory for individual tasks was also analyzed in Experiment 2. This memory variable required both correct memory for the task and a correct categorization. These data were evaluated using a 2 x 4 repeated measures ANOVA with span type (memory or task) and focus classification (focused, defocused, unfocused-immediate, and unfocused-deferred) as factors. These data are presented in figure B1. The effect of span type was not significant, *F* < 1, nor was the effect of focus classification, *F*(4, 76) = 2.84, *p* = .108. The interaction between these factors was also not significant, *F* < 1.

Experiment 2 accuracy.

Participants' accuracy was calculated as their correct categorizations for the correct task in the task span. For the memory span, accuracy is the same measure as memory, so the task memory measure is a reflection of accuracy. Task accuracy data for the task span are depicted in figure B2. There was no significant difference in accuracy in the task span as a function of focus classification, as evidence by a non-significant repeated measures ANOVA, *F* < 1. These accuracy data support the interpretation that there was no speed/accuracy trade-off for participants in the task span, as focus classification did not influence accuracy.

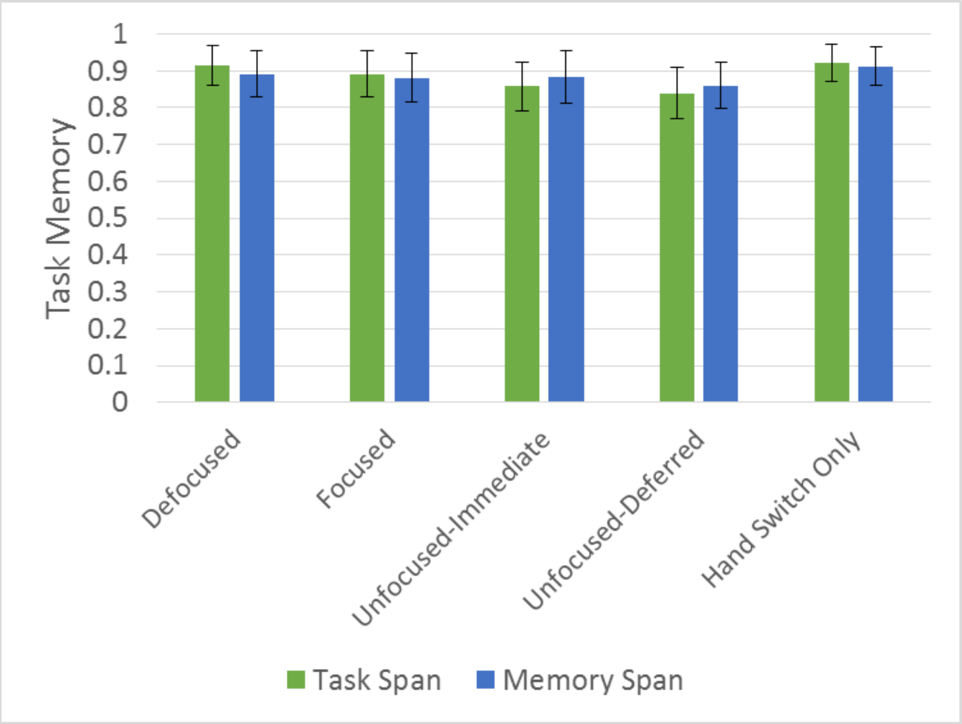

Figure B1. Mean task memory for targets in Experiment 2 for memory and task span as a function of focus classification.

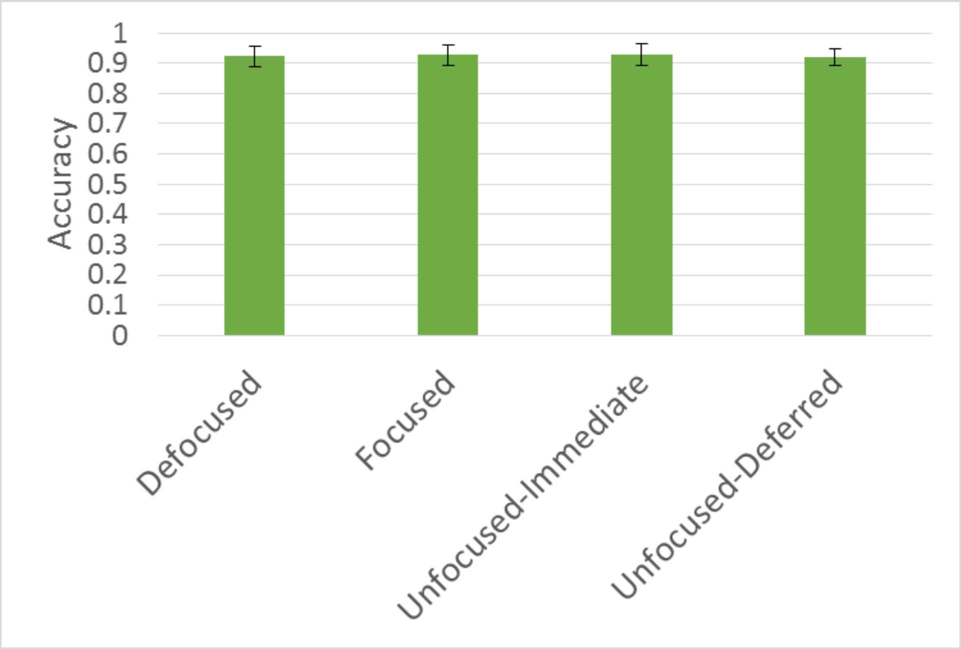

Figure B2. Mean accuracy for items in Experiment 2 as a function of focus classification for the task span.

Appendix C

Additional analyses and figures—Experiment 3

Experiment 3 accuracy.

Participants' accuracy was calculated as their correct categorizations for the correct task in the task span. Accuracy was analyzed as a function switch frequency. A 4 x 5 mixed factor ANOVA with set size condition (2-2, 4-4, 2-4, and 4-2, between subjects) and switch frequency (1, 2, 3, 4, & 5 switches, within subjects), yielded a significant effect of condition, *F* (3,116) = 19.66, *p* < .001, η^2_{partial} = .337. Participants' accuracy was lowest for the 4-4 condition and was higher and equivalent for other conditions. These data are depicted in figure C1. The effect of switch frequency was not significant, *F* < 1. The interaction between these two factors was also not significant, *F* (12,464) = 1.10, *p* = .351.

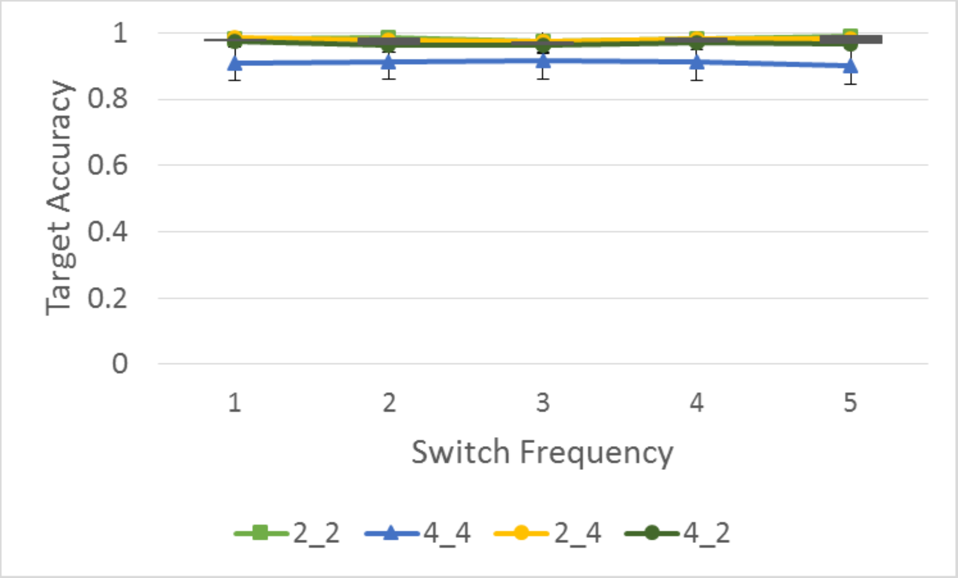

Figure C1. Mean accuracy for items in Experiment 3 as a function of switch frequency and size of the active and passive task sets for the task span (TS).

KAITLIN M. REIMAN

17 Memorial Dr. East • Bethlehem, PA 18015
kmr210@lehigh.edu

EDUCATION

Ph.D., Cognitive Psychology: **Lehigh University,** Bethlehem, Pennsylvania May 2015

Dissertation:
*Are Declarative and Procedural Working Memory Functionally Analogous?
Testing Working Memory Using the Task Span*
Advisor: Catherine Arrington, Ph.D

M.S., Cognitive Psychology: **Lehigh University**, Bethlehem, Pennsylvania
May 2012

Master's Thesis:
*Executive Control, Working Memory and Action Planning:
An Individual Differences Approach*
Advisor: Catherine Arrington, Ph.D

B.A., Psychology: **The College of New Jersey**, Ewing, New Jersey
May 2010
Cum Laude

Honor's Thesis:
Aging and Isolation Effects, a Study of Real World Actions
Advisor: Tamra Bireta, Ph. D

PUBLICATIONS & ACCEPTED MANUSCRIPTS

Arrington, C. M. & **Reiman, K. M**. (Accepted). Task frequency influences stimulus-driven effects on task selection during voluntary task switching. *Psychonomic Bulletin & Review.*

Reiman, K.M., Weaver, S.M., & Arrington, C.M. (2015). Encoding and choice in the task span procedure. *Psychological Research, 79,* 267-281

Arrington, C. M., **Reiman, K. M.,** & Weaver, S. W. (2014). Voluntary Task Switching. In J. Grange & G. Houghton (Eds.), *Task Switching and Cognitive Control.* Oxford University Press.

MANUSCRIPTS IN PREPARATION

Reiman, K. M. & Arrington, C. M. (n.d.). Planning tasks in sequences: The contributions of working memory and executive control. Manuscript in preparation.

Arrington, C. M., & **Reiman, K. M.** (n.d.). Tracking eye movements in voluntary task switching. Manuscript in preparation.

Weaver, S. M., Arrington, C. M., **Reiman, K. M.,** & Wiley, G. R. (n.d.). Differential neural activation associated with storing information for use as part of a plan and for later retrieval. Manuscript in preparation.

PRESENTATIONS AT SCHOLARLY MEETINGS

Reiman, K. M., Braun, D. A., & Arrington, C. M. (2015, March). Choosing to look or looking to choose: Eye tracking in voluntary task switching. Poster presented at the inaugural International Convention of Psychological Science, Amsterdam, The Netherlands.

Reiman, K. M. & Arrington, C.M. (2014, November). Declarative and procedural working memory in the voluntary task span. Poster presented at the annual meeting of The Psychonomic Society, Long Beach, CA.

Weaver, S. M., Arrington, C. M., **Reiman, K. M.** & Wiley, G. R. (2014, November). Planning to use it later: Differential neural activation associated with storing information for use as part of a plan and for later retrieval. Poster presentated at the annual meeting of The Psychonomic Society, Long Beach, CA.

Reiman, K.M., & Arrington, C.M. (2013, November). Voluntary task switching between tasks of unequal strength. Poster presented at the annual meeting of The Psychonomic Society, Toronto, Canada.

Arrington, C.M.& **Reiman, K. M.** (2013, November). The Influence of External Stimuli on Task Selection for Tasks of Unequal Frequency. Poster presented at the annual meeting of The Psychonomic Society, Toronto, Canada.

Reiman, K.M., Arrington, C.M. (2013, March). Executive control and working memory in the task span paradigm. Oral presentation at the annual meeting of the Eastern Psychological Association, New York, NY.

Reiman, K.M., & Arrington, C.M. (2012, November). Executive control, working memory and action planning: An Individual differences approach. Poster presented at the annual meeting of The Psychonomic Society, Minneapolis, MN.

Reiman, K.M., Arrington, C.M. (2012, March). Encoding and retrieval in the task span procedure. Oral presentation at the annual meeting of the Eastern Psychological Association, Pittsburgh, PA.

Reiman, K.M., Weaver, S.M., & Arrington, C.M. (2011, November). Encoding and choice in the task span procedure. Poster presented at the annual meeting of The Psychonomic Society, Seattle, WA.

Weaver, S.M., **Reiman, K.M**., & Arrington, C.M. (2011, November). Predictable stimulus positions decrease bottom-up influences on task choice. Poster presented at the annual meeting of The Psychonomic Society, Seattle, WA.

Reiman, K.M., Weaver, S. M., & Arrington, C. M. (2011, March). The effect of complexity on implementing a plan in the task span paradigm. Poster presented at the annual meeting of the Eastern Psychological Association, Cambridge, MA.

PROFESSIONAL AFFILIATIONS

New York Academy of Sciences
Association for Psychological Science
Eastern Psychological Association
Women in Cognitive Science