Journal of Memory and Language 74 (2014) 1–15

Contents lists available at ScienceDirect

Journal of Memory and Language

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

Working memory updating involves item-specific removal

Ullrich K.H. Ecker [a,*], Klaus Oberauer [b], Stephan Lewandowsky [a,c]

[a] School of Psychology, University of Western Australia, 35 Stirling Hwy, Crawley 6009, Australia
[b] Department of Psychology, University of Zurich, Binzmühlestrasse 14/22, 8050 Zurich, Switzerland
[c] School of Experimental Psychology, University of Bristol, 12a Priory Road, Clifton, Bristol BS8 1TU, United Kingdom

ARTICLE INFO

Article history:
Received 4 December 2013
revision received 12 March 2014
Available online 19 April 2014

Keywords:
Working memory
Updating
Removal
Executive function

ABSTRACT

The ability to keep working memory content up to date is vital for a number of higher cognitive functions such as navigation and reasoning, but it is also crucial for the effective operation of working memory itself. Removing outdated or irrelevant information allows focused processing of relevant information, and minimizes interference. We present evidence from three experiments that (1) people utilize an active removal process to update working memory, (2) that this removal process is an item-specific operation, and (3) that updating subsets of information held in working memory involves switching between maintenance and updating modes of working memory.

Introduction

Working memory (WM) is a severely capacity-limited system serving the configuration, manipulation, and maintenance of mental representations in support of ongoing cognition (Baddeley, 1986; Cowan, 1999; Oberauer, 2009). The flexible configuration of representations relies on ad hoc integration of item and context information. For example, memorizing a telephone number requires the binding of digits to serial positions, and maintaining an accurate representation of one's environment while navigating traffic requires binding of shape, color, and dynamic location information over time (Oberauer, 2005; Wilhelm, Hildebrandt, & Oberauer, 2013, also see Ecker, Maybery, & Zimmer, 2013).

Because of WM's capacity limitations, a fundamental requirement for its smooth operation is a mechanism that ensures outdated and no-longer-relevant information is discarded. Without such a mechanism, clutter from irrelevant information would prevent access to relevant information and would thus ultimately render efficient cognitive processing impossible.

Traditionally, a temporal decay mechanism has been proposed to serve this 'housekeeping' function (Baddeley, 2000; Barrouillet, Bernardin, Portrat, Vergauwe, & Camos, 2007). However, supported by the growing evidence that WM representations do not, in fact, decay (Berman, Jonides, & Lewis, 2009; Jalbert, Neath, Bireta, & Surprenant, 2011; Lewandowsky, Oberauer, & Brown, 2009; Oberauer & Lewandowsky, 2008; Oberauer & Lewandowsky, 2013), we recently suggested that no-longer-relevant information is discarded by an active removal process (Ecker, Lewandowsky, & Oberauer, in press).

In Ecker et al. (in press), we argued that this removal process is the central component of WM updating, as for example when the operands "8" and "3" must be replaced by their product "24" during mental arithmetic. Updating WM is a process that is essential for maintaining a focus on relevant information and replacing outdated with current information, and is thus crucial for mental arithmetic, reading comprehension, navigation, and reasoning (Carretti, Cornoldi, De Beni, & Romano, 2005; Chen & Li, 2007; Garavan, 1998; Gugerty, 1997). By definition, WM

* Corresponding author.
E-mail addresses: ullrich.ecker@uwa.edu.au (U.K.H. Ecker), k.oberauer@psychologie.uzh.ch (K. Oberauer), stephan.lewandowsky@bristol.ac.uk (S. Lewandowsky).

http://dx.doi.org/10.1016/j.jml.2014.03.006
0749-596X/

updating involves the substitution of outdated by new information and may thus be described as a two-component process: outdated information is removed and updated information is encoded to replace it (Ecker, Lewandowsky, Oberauer, & Chee, 2010, also see Artuso & Palladino, 2011).

To investigate this proposed decomposition, Ecker et al. (in press) introduced a novel updating task that allowed the experimental separation of removal and encoding processes. Previous investigations of WM updating, using traditional WM updating tasks, focused either on the effectiveness of WM updating (e.g., Friedman et al., 2006) or its overall duration (Artuso & Palladino, 2011; Kessler & Meiran, 2008). Traditional WM updating tasks present a set of to-be-remembered items (e.g., the letters B–K–F) and then repeatedly replace one or more items (e.g., B–H–F . . . W–H–J, and so on) for participants to keep track of before they recall the most recent set at the end of a trial. In tasks of that type, removal of outdated items can only begin when the new items are presented. The time to update WM will thus include both time for removal and time for encoding. The novel task used by Ecker et al. (in press) presented a cue, indicating which item was about to be replaced, before presenting the new to-be-encoded stimulus. This allowed participants to use the cue-target interval (CTI) to selectively remove an outdated item from the memory set. Varying this CTI thus varied the time available for removal. Ecker et al. (in press) found that longer CTIs led to faster updating, and argued that people used those long CTIs for removal, thus supporting the notion of a two-stage updating process comprising removal and encoding operations. Additional evidence for a removal process came from trials in which the to-be-replaced item in WM was similar, or even identical, to the new item replacing it. Ecker et al. (in press) found that with a short CTI, similarity or identity between the old and the new item reduced updating RTs, but these RT benefits were strongly reduced with a long CTI. This finding provides additional evidence that the long CTI was used for removing the old item, and not just to search for the to-be-updated item in the list (e.g., cf. Lange, Cerella, & Verhaeghen, 2011).

In the present paper, we present further evidence that an active removal process is central to WM updating, and expand on the previous work by demonstrating that (1) this removal operation is item-specific and goes beyond finding a to-be-updated item, and (2) how the removal process fits into a recent computational model of WM updating, which assumes that updating subsets of information held in working memory involves switching between maintenance and updating modes of working memory (cf. Kessler & Oberauer, in press).

We begin by revisiting a finding from the literature that suggests the involvement of multiple process in WM updating. Kessler and Meiran (2008) used an updating task that presented n items in a set of frames, and substituted between 1 and n items repeatedly with new items. The authors measured the time it took their participants to complete this updating process, and found (in their Experiment 3) that updating RTs increased with the number of to-be-updated items up to $n - 1$ items, but that updating was much faster again when all n items were replaced on a given updating step. Thus, updating latencies depended in a non-monotonic fashion on the number of to-be-updated items.

Kessler and Meiran (2008) explained this non-monotonicity by proposing a distinction between local and global updating processes. Local updating refers to changes made to individual items, whereas global updating refers to the integration of all items in the current memory set after individual items were updated. The authors argued that partial updates require a complex sequence of (1) "unbinding" of the integrated representation of the previous memory set, (2) substitution of some but not all items (i.e., the actual local updating), followed by (3) re-binding the new set as part of the global updating process. In contrast, when the entire set is updated, steps (1) and (2) can be omitted, the old set is simply discarded and a new memory set is encoded and bound.

Our interpretation of the non-monotonicity of updating latencies is a specific instantiation of the ideas of Kessler and Meiran (2008), in light of the results of Ecker et al. (in press) and in the context of the SOB (serial order in a box) model of WM (Farrell & Lewandowsky, 2002; Lewandowsky & Farrell, 2008; Oberauer, Lewandowsky, Farrell, Jarrold, & Greaves, 2012). SOB computationally implements the informational integration that is key to WM processing in a two-layered neural network that associates items (represented in one layer) with positional context markers (represented in the second layer) via Hebbian learning. To this end, the two layers are connected by a weight matrix that is continuously adjusted by the learning algorithm. For instance, in an updating task such as the one by Kessler and Meiran (2008), SOB would associate each item to its frame through Hebbian learning. Forgetting in SOB occurs purely because of interference (there is no temporal decay mechanism). Interference from outdated information is prevented by an active removal mechanism (see in particular Oberauer et al., 2012). Removal of a specific item in SOB involves cuing the item with its position marker to retrieve it and then "unlearning" the association between the item and its position via Hebbian anti-learning (cf. Anderson, 1991; Lewandowsky, 1999). Thus, the removal of outdated and the encoding of new information is accomplished by two separate processes in SOB.

Kessler and Meiran (2008)'s notion of unbinding refers in SOB to the unlearning of selected items from their position markers. Like encoding of an item into the network (Jolicoeur & Dell'Acqua, 1998), the act of removing an individual item takes time (cf. Fawcett & Taylor, 2008; Oberauer, 2001). By contrast, wholesale removal of an entire memory set can be achieved in SOB by simply resetting the entire weight matrix, which we assume to be a very rapid process. It follows that updating an entire memory set does not require active, item-wise removal, and updating RT will mainly comprise the time needed for encoding the new memory set. This explains why updating an entire memory set is faster than partial updating of just one or two items within a larger set. This distinction between partial, item-wise updating and complete updating of an entire memory set is supported by neuro-cognitive evidence for selective recruitment of a neural network by partial updating as opposed to total set replacement (Murty et al., 2011).