The skill-based approach

Hoekstra, Corné

DOI:
10.33612/diss.769566695

The skill-based approach

Developing and applying a modelling method based on skill reuse

Corné Hoekstra

Cover design by Corné Hoekstra, Abigail Toth and Marcus Hoekstra
Printed by Ipskamp Printing

The skill-based approach

Developing and applying a modelling method
based on skill reuse

Proefschrift

ter verkrijging van de graad van doctor aan de
Rijksuniversiteit Groningen
op gezag van de
rector magnificus prof. dr. ir. J.M.A. Scherpen
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

dinsdag 26 september 2023 om 14.30 uur

door

Corné Hoekstra

geboren op 25 januari 1994
te Bolsward

Promotor
Prof. dr. N.A. Taatgen

Copromotor
Dr. S. Martens

Beoordelingscommissie
Prof. dr. D. Salvucci
Prof. dr. J.P. Borst
Prof. dr. D. de Waard

Contents

1
Introduction

Human cognition is extremely flexible and can support a wide range of tasks. From speaking (multiple) languages to playing chess or driving a car. To support the knowledge required to accomplish all these tasks in the limited capacity available to people, there has to be overlap in this knowledge and it needs to consist of multi-purpose blocks of knowledge. Additionally, people are very quick in learning (simple) tasks. This suggests that people do not need to figure out from scratch how to perform a task but can instead draw from already available blocks of knowledge. These insights are widely accepted in the cognitive literature (Anderson et al., 2011; Salvucci, 2013), however current modelling practices rarely reflect it. Almost all models created in the fields of (cognitive) psychology and cognitive science are only developed for a single task with little regard for other related tasks. This implicit assumption that the mechanisms modelled in a single-task model only apply to this one task and context often imposes a large risk of overfitting on models constructed in this manner and it adds to the insularity of cognitive science.

The goal of this dissertation is to develop a modelling approach that allows modelers to put these insights into practice. This dissertation will focus on implementing such an approach in the cognitive architecture PRIMs (Taatgen, 2013); however, our efforts in implementing this in PRIMs will also be informative for how to implement it in other architectures (specifically the highly related architecture ACT-R). When successful, this modelling approach can be very valuable to the field of cognitive science. A modelling approach based on reuse will help modelers to create more generalizable, constrained and cognitively plausible models. Additionally, constructing models with the idea of reuse in mind can provide additional insights into an existing experimental paradigm that cannot be gained when only constructing single-task models.

The central concept in this modelling approach is a skill. A skill is a unit of (procedural) knowledge that accomplishes a basic general cognitive processing step that is applicable to (many) different tasks. For example, consolidating an item into memory, solving an algebraic equation, or determining whether a number is odd or even. These are just a few examples as what can be called a skill is very broad and, theoretically, any cognitive processing step that is reusable can be a skill. The basic idea behind our modelling approach is that any task is a combination of several basic skills and that learning a new task only consists of assembling the correct skills.

1. The cognitive architecture PRIMs

PRIMs is the cognitive architecture at the centre of our efforts of developing a modelling approach based on skill reuse. PRIMs is heavily based on ACT-R and functions largely the same way. PRIMs was developed with the explicit aim of modelling knowledge transfer between tasks and therefore is very well suited to support a modelling approach based on reuse of procedural knowledge. As a basic understanding of PRIMs will facilitate the understanding of the following chapters of this dissertation, we will start with a basic overview of PRIMs.

1.1. The modules

Similar to ACT-R, PRIMs is a modular cognitive architecture. This means that the architecture consists of several independent cognitive modules capable of executing their respective function independently from one another and in parallel. PRIMs consists of five modules as depicted in Figure 1: (1) the visual module, (2) the declarative memory module, (3) the working memory module, (4) the task control (or goal) module, and (5) the manual module. These modules are capable of performing a task from start to finish and they communicate through the central workspace via PRIMs.

Firstly, the visual module. This module is also called the input module since its functionality is not limited to only visual information, it can be used to represent any type of sensory input (in practice it mainly is used for visual or auditory input). This module is very basic and consists of several buffer slots which represent the sensory input by means of symbols. For example, a red triangle would be represented by this module in two separate buffers, one buffer containing the symbol 'red' and the other containing the symbol 'triangle'. In this manner any type of input can be provided to the central workspace.

Secondly, the declarative memory module, this module is usually referred to by its abbreviation as the DM module. This module is more detailed and it represents the declarative memory (DM) system of the architecture. This module is responsible for the storage and retrieval of memory chunks and therefore plays a central role in many models. It functions very similarly to the ACT-R DM module. It is populated by chunks which can be retrieved by means of a retrieval request. The chunks activation-levels behave almost identically to the chunks in ACT-R's declarative memory; it uses the same formulas to calculate the current activation of a chunk and the associated retrieval time. The main difference is that a currently active skill spreads activation to associated chunks The DM module takes as input retrieval requests which are partly completed chunks and it outputs the entire chunk with the highest activation that completes the input pattern.

Thirdly, the working memory module. This module is often referred to as the imaginal buffer. This module is responsible for the temporary storage of highly relevant information. Similar to all the other modules it consists of several buffer slots. Relevant information can be placed in one of these slots by means of a PRIM without any penalty and it will be kept in this buffer slot until it is removed (also by a PRIM). Therefore, the imaginal buffer is capable of perfect storage for (theoretically) an unlimited amount of time. In practice, this module will only hold on to information for a few seconds, however this is mainly a result of modelling convention and not an architectural constraint. This module plays a central role in many models as it provides a crucial role as the keeper of short-term relevant information (i.e., working memory). In chapter 4 and chapter 5, the imaginal buffer will be discussed in more detail and its plausibility and role in supporting skill reuse will be more closely examined.

Fourthly, the task control module. This module is almost exclusively referred to as the goal module as its place in the architecture is very similar to the ACT-R goal module. Similar to ACT-R, the PRIMs goal module is responsible for the 'big picture' control of a model. It lays out what the general next step of a model should be and strongly influences which operator (PRIMs' version of production-rules) will be selected. However, the PRIMs' goal module functions rather differently from its ACT-R counterpart. The goal module does not directly determine which operator will be selected by matching the left-hand side of an operator. Instead it only spreads activation to the operators that are associated with the current goal which makes it more likely that one of these operators is selected, but does not guarantee it. The goal buffer is instrumental for the central concept of this dissertation: the skill. Because of the close relationship between goal and skill, the terms are often (sometimes confusingly) used interchangeably, however they are not exactly the same. A goal (i.e., the symbol representing this goal in the goal buffer) is the abstract 'intention' or objective that the model wants to accomplish while a skill is the collection of operators that will achieve this objective.

Finally, the manual module. This module is often referred to as the action module since it is responsible for the execution of the (often manual) action of a model (i.e., the model 'output'). This module is fairly simple and is very flexible. It can execute any action that is defined in the model script, these are usually simple actions such as pressing a button, doing a saccade, or saying a word out loud. This module achieves this high level of flexibility because all important characteristics of an action can be specified, including the time it takes to execute it (including a noise parameter) and the 'product' of an action (i.e., which button is pressed on the keyboard or what word is being said).

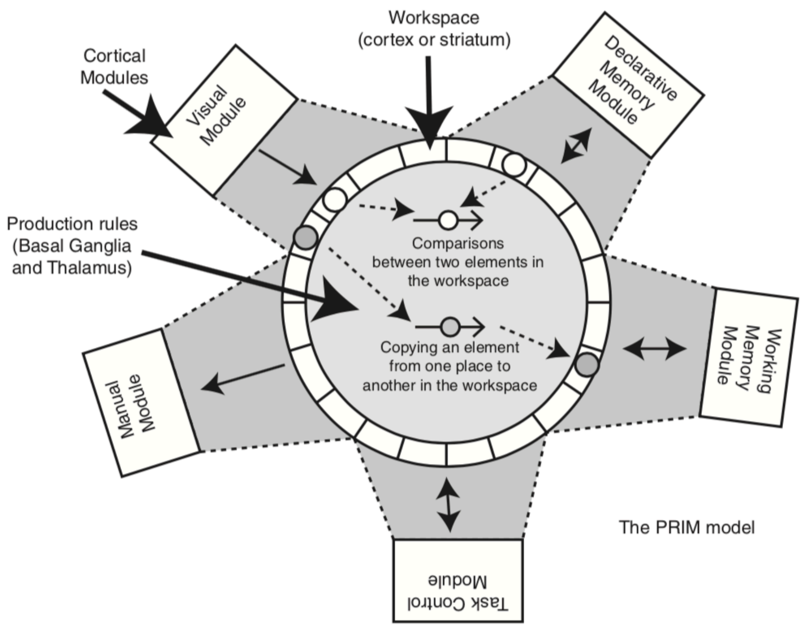

Figure 1. *Overview of the PRIMs cognitive architecture.*

1.2. The productions

The modules can function independently, however, to be able to perform complex cognitive functions and accomplish tasks, the modules need to interact and communicate. In all production-based cognitive architectures such as PRIMs and ACT-R, this is done by productions. The productions are the 'instructions' that govern the behaviour of any model in a similar way as computer code governs the behaviour of a computer program. Constructing a model in any such architecture usually mainly revolves around creating the appropriate productions that will prompt the model to behave in the intended way.

In PRIMs, these instructions are specified by creating operators. Operators contain the instructions for a certain model about how to complete the modelled task. Operators consist of two separate parts, the conditions and the actions. The conditions specify in what context an operator should fire. For example, an operator that should fire when there is a red triangle in the visual buffer would include conditions specifying that the first slot of the visual buffer should be *red* and the second slot should be *triangle*. In PRIMs terminology, this would be specified by writing *V1 = red* and *V2 = triangle*. The action part specifies what actions should be taken when the conditions are true. For example, move what is currently in the visual buffers to the declarative memory module into the working memory module so that the last seen stimulus will be remembered. This would be specified by writing *V1 -> WM1* and *V2 -> WM2*.

An interesting aspect about PRIMs (which is crucial for our goal of designing a modelling approach based on skill reuse) is that every operator consists of individual PRIMs. PRIMs are considered to be the basic building blocks of cognition and are therefore universally applicable for any task that is modelled using (the architecture) PRIMs. Only two types of PRIMs exist, condition PRIMs which can compare two values in the workspace and action PRIMs which are capable of moving a piece of information from one place in the workspace to another. This is why the conditions in the previous example have to be specified piecemeal, the presence of a red triangle in the visual buffer cannot be tested with a single 'command' checking for the presence of a red triangle as a whole. Because it consists of two characteristics (at least in this example), two separate PRIMs are needed: one checking for the presence of *red* in *V1* and a second checking for the presence of *triangle* in *V2*. The same applies to the action PRIMs, in order two move two pieces of information, two separate PRIMs are needed.

In turn, operators can be organized into skills, the central concept of this dissertation. Skills are collections of operators that are capable of accomplishing a certain cognitive processing step. Skills are very flexible as the operators of a skill do not need to be executed in a certain order. This allows skills to achieve a certain goal relatively independent of the initial state of the workspace. Additionally, skills are fairly insensitive to the exact state of the workspace since they can include multiple operators performing the same action but with slightly different conditions. These characteristics make skills very powerful for reuse since they are flexible enough to function in different circumstances but at the same time they are still specific enough to accomplish their intended goal.

To summarize, procedural knowledge in PRIMs is organized in the following way. At the lowest level, it is specified by individual PRIMs which consist of one simple instruction to either compare two values or move a piece of information. At the middle level are the operators which are collections of PRIMs. Operators are comparable to ACT-R production rules and they accomplish very specific and small cognitive steps. Finally, operators can be organized into skills. Skills are the largest unit of procedural knowledge in PRIMs and they consist of several operators that taken together can reliably accomplish a larger goal.

2. Theoretical background of the modelling approach

As was mentioned before, the basic idea on which our modelling approach is based is that people continually reuse previously learned blocks of already learned procedural knowledge. In our approach these blocks are termed skills. This collection of skills continually evolves over someone's lifetime; however, at any specific moment (e.g., when in the lab as a participant), a person merely applies the collection of skills that they currently have available to the challenges presented by the current task. This idea is developed from theories of skill acquisition in the line of research started by Fitts (1964). The original theory was later expanded on and refined by several researchers (J. R. Anderson, 1982; Kim, Ritter, & Koubek, 2013; Proctor & Dutta, 1995; Rasmussen, 1987) resulting in a consensus that skill learning occurs in three distinct stages. (1) A first stage responsible for learning basic declarative and procedural knowledge, (2) a second stage responsible for consolidating this knowledge and, finally, (3) optimizing this knowledge for the current task.

In the context of PRIMs and our modelling approach, the first stage is represented by learning combinations of PRIMs and combining them into operators. The second stage is represented by learning which operators are associated with a certain skill and completing a basic level of production compilation on these operators. The third stage is represented by finalizing production compilation and (if enough practice is available) optimizing the contents of a skill by pruning unnecessary operators or adapting the existing operators to the specific requirements of the current task. Fully completing this final stage would result in a very specific skill which cannot easily be applied to other situations anymore.

We believe that, for adults in practice, most learning occurs on the second and third stages and that this type of learning is thus the most crucial learning that should be accounted for in cognitive models. For example, the learning that occurs when a participant enters a lab to perform a simple psychological task, most often happens between the second and third stages. A participant enters the lab with a certain amount of already available operators (i.e., they have fully completed first stage learning). Depending on the exact task they have likely also already learned a large number of basic skills (i.e., they have learned which operators successfully go together and for which purpose) but probably not all of the required skills are fully compiled (in chapter 5 we will explore what different levels of compilation can do to individual performance). Therefore, most of the learning that participants experience during an experiment and that should be accounted for in models occurs on the second stage (finalizing this stage) and the third stage (starting the process of optimizing the basic skill for the current task).

3. Chapter overview

In this dissertation we discuss our efforts in arriving at a modelling approach capable of reusing skills. We termed this approach the skill-based approach.

In chapter 2, we present the basic ideas and design of the skill-based approach. The skill-based approach is based on the idea that when people are confronted with a new task, they do not need to figure out from scratch how to accomplish this task but can, instead, rely on previously learned procedural knowledge. In this chapter we propose the basic steps involved in creating a model using the skill-based approach and test the feasibility of these by creating a model of the attentional blink.

In chapter 3, we slightly deviated from directly developing the skill-based approach and we focused on the attentional blink model. In this chapter, we further develop the attentional blink model created in chapter 2 and perform two experiments testing its predictions. The main prediction of this model is that the attentional blink is caused by selection of a sub-optimal skill. In order to test this prediction, we perform an experiment aimed at manipulating the skill used by the participants during the task. Our results suggest that the attentional blink task can be performed with two different skills and that one of these skills leads to much better performance.

In chapter 4, we returned to focus on the skill-based approach and discuss three limitations to the skill-based approach and the cognitive architecture we used while attempting to model the nine basic executive function tasks in Miyake et al. (2000). Two of these limitations were limitations to the initial design of the skill-based approach and one limitation concerned the cognitive architecture PRIMs. In this chapter, we discuss the details of these limitations and their relationship to the general issue of improving generalizability in cognitive modelling. Finally, we propose initial ideas as to how these limitations can be overcome.

In chapter 5, we built on the initial suggestions from chapter 4 about how to overcome the limitations and propose and implement solutions. The solutions were tested by creating models of the same executive function tasks that initially gave rise to the limitations. The proposed solutions were successful and allowed for the creation of the model of the executive function tasks using the skill-based approach. Furthermore, the models suggest that executive functioning heavily relies on learned skills (i.e., procedural knowledge) and it proposes procedural and automatic mechanisms for all three basic executive functions.

2

A skill-based approach to modelling the attentional blink

People can often learn new tasks quickly. This is hard to explain with cognitive models, because they either need extensive task-specific knowledge or a long training session. In this article we try to solve this by proposing that task knowledge can be decomposed into skills. A skill is a task-independent set of knowledge that can be reused for different tasks. As a demonstration, we created an attentional blink (AB) model from the general skills that we extracted from models of visual attention and working memory. The results suggest that this is a feasible modelling method, which could lead to more generalizable models.

This chapter has previously been published as:

Hoekstra, C., Martens, S., & Taatgen, N. A. (2020). A Skill-Based Approach to Modeling the Attentional Blink. Topics in Cognitive Science, 12(3), 1030-1045.

1. Introduction

Humans have the impressive ability to learn certain relatively simple tasks with minimal instruction and in a very short period of time. The experimental tasks used in (cognitive) psychology are particularly good examples of these types of tasks. Participants have often never encountered these tasks before, yet are quickly able to work out what to do. This quick learning suggests that people reuse previously learned skills and apply them to new contexts (Salvucci, 2013; Taatgen, Huss, Dickison, & Anderson, 2008). For example, if a task requires a stimulus to be remembered for later recall, people do not have to work out how to remember the stimulus, but they can simply reuse the already learned 'remembering-skill'. It would be unnecessary, in this case, to reinvent the wheel. Learning how to do a new task simply means selecting the appropriate skills, assuming all these skills have already been acquired.

Skill is a commonly used term in cognitive psychology and is used to convey many (slightly) different meanings. In the context of this paper, skill refers to the largest unit of procedural knowledge that can be reused in different tasks. There can be many instances in which the same procedural knowledge (i.e., skills) can be reused. Traditionally, the idea of skill-reuse was often implemented in cognitive architectures in the form of subgoaling (Newell, 1990). A main task goal could be decomposed in subgoals, each of which could in theory be reused for other tasks. However, the subgoaling mechanism proved to be too brittle to support flexible reuse. Moreover, psychological data turned out to be inconsistent with models using a goal stack (Anderson & Douglass, 2001). The skill-based approach shares similarities with subgoaling, but does not use the goal-stack mechanism.

Reusing skills speeds up learning, but it can also have negative side effects that lead to sub-optimal performance even though the cognitive system is, in principle, capable of optimal performance. That is, it is sub-optimal strategy that underlies the impaired performance, not a fundamental information processing limit (e.g., Taatgen, Juvina, Schipper, Borst, & Martens, 2009). One factor underlying the sub-optimal strategy-choice might be the selection of the wrong skills, either because the "right" skill is not available, or because the interpretation of the task cues the wrong skill. A well-known instance of this is the Stroop effect (Stroop, 1935). Because people are so used to reading words, this automatically triggered skill interferes with the task of naming the colour of the word. In this case, words trigger the 'reading-skill', which leads to worse performance. Another, less obvious, instance where this can happen is the attentional blink (AB).

The AB is a well-studied phenomenon in cognitive psychology (Martens & Wyble, 2010). It refers to the finding that the second of two to-be reported targets in a stream of distractors presented at a rate of 100 ms per item is often missed when it is presented within an interval of 200-500 ms after the first target (T1) (Raymond, Shapiro, & Arnell, 1992). Interestingly, the second target (T2) is hardly ever missed if it is presented directly (i.e., within 100 ms) after the first target (lag-1 sparing). This suggests that the cognitive system does possess the processing capacity to identify both targets, but that the chosen strategy prevents the second target from being reported.

The crucial aspect of the strategy that most participants use can be the selection of a sub-optimal skill to consolidate the targets in memory. Many theories of the AB assume that consolidation of T1 into memory underlies the AB (Akyürek, Abedian-Amiri, & Ostermeier, 2011). Memory consolidation is thought to be a serial process, meaning that only one consolidation process can occur at a time and that the consolidation has to be completed before a new item can be consolidated. This means that T2 cannot always be consolidated straight away, but sometimes has to wait for the consolidation of T1 to be completed. This leads to the AB when consolidation of T1 has not yet been completed before T2 has disappeared from visual short-term memory. However, such theories all assume that targets are consolidated as separate memory items, whereas in other areas of memory research it is assumed that multiple items are consolidated in a single chunk.

The strongest indication that strategy underlies the AB phenomenon is an experiment by Ferlazzo and colleagues (Ferlazzo, Lucido, Di Nocera, Fagioli, & Sdoia, 2007). In their experiment, participants were instructed to report two target letters (which were always a vowel and a consonant) either separately or as a single syllable. In the latter condition participants did not exhibit an AB. A possible explanation is that the original instruction cues a strategy in which all targets are consolidated separately, while the syllable instruction encourages consolidation of both targets in a single chunk. We will explore this difference by creating two versions of an AB-model that only differ in their consolidation strategy.

To create the model, we have used a novel approach. Instead of creating the model specifically for the AB, we built a model from general skills that we have constructed as parts of other models. In other words, the AB model only links together existing skills. We chose this approach because it mirrors how participants performing an AB-task work out what to do. They do not start from scratch, but they tie skills they already possess together in such a way that allows them to perform an AB-task.

Many tasks share similarities and, therefore, many tasks require the same skills. The tendency of people to utilize this overlap between tasks calls for the creation of more generalizable models to reflect this approach. Moreover, creating models that use generalizable elements allows for the mechanisms used in these models to be placed in a larger cognitive context because each mechanism is essentially part of a generalizable skill. Currently, phenomena are usually modelled by specific cognitive machinery that captures the phenomenon found in empirical data. It is not common to place this machinery in a larger cognitive context and describe how it relates to other cognitive processes (i.e., specifying “where” or “when” it takes place in cognition). However, this leads to models with very specific mechanisms created to explain findings in one particular experimental paradigm. In contrast, creating models with a skill-based approach forces the mechanisms to be placed into a larger cognitive context because it is created as a part of a general skill that is also used in other tasks and cognitive processes. Because its underlying mechanism is the same for these tasks, it should also be predictive of behaviour outside of the initially modelled paradigm.

We believe that creating cognitive models in this manner can be a promising contribution to the field in general. In particular because it could aid generalization among the many different models created in the highly specialized and compartmentalized fields of cognitive science. The goal of the skill-based approach is a continuation of one of the fundamental goals of cognitive architectures. Cognitive architectures have been developed in order to create a basic framework which can be applied to model a large variety of tasks. This ensures a certain amount of correspondence between models created using the same architecture, independent of the specific task modelled. The skill-based approach extends this idea. Applying this approach, in combination with a cognitive architecture, will allow for not only the basic architecture to be considered but also previously acquired knowledge (skills).

We created our AB-model in the cognitive architecture PRIMs (Taatgen 2013; 2014). PRIMs (abbreviation for primitive information processing elements) is based on ACT-R (Anderson et al., 2004) and works in a highly comparable way. The architectures of both ACT-R and PRIMs consist of a 'central workspace' and a number of modules capable of performing specific cognitive functions. The modules can communicate (i.e., exchange the results of their cognitive operations) with each other through the central workspace, which is subdivided in buffers. This exchange of information between the modules in PRIMs is controlled in largely the same way as it is in ACT-R. In ACT-R this is done by productions, and in PRIMs it is done by operators, but they have similar functionalities. For a more extensive discussion of PRIMs, see (Taatgen, 2013). A crucial difference between ACT-R and PRIMs is that in PRIMs operators are by default further organized into skills. A skill is a collection of operators capable of accomplishing a certain goal or processing step. Skills, therefore, form the bridge between single operators and the complete task that is being modelled. A task is built up from a certain number of skills and a skill, in turn, consists of a certain number of operators. This distinction is helpful because it allows for more flexibility while maintaining a high level of organization. Flexibility is improved because the model can diverge from the beaten path if the situation asks for it. Additionally, it allows for connections between operations that cannot be executed in a single operator (e.g., because they use the same buffers).

Skills are combined into tasks by instantiating variables that are part of the skill (and, in turn, the underlying operators). For example, a skill for visual search may be instantiated by the type of item that we search for. Variable instantiation is also used to link skills together. For example, the visual search may need a more elaborate criterion, which in itself is another skill (e.g., search for a sheep with five legs). Therefore, building a task model that can be composed of existing skills entails instantiating the variables in those skills (and nothing else).

The generalizability of skills makes it possible to use the same skills in models of different experimental tasks. The organization into skills thus allows us to employ a novel approach to constructing cognitive models, placing them in a context of related models, tasks, and skills. We had two main goals in this project. Firstly, we wanted to investigate the feasibility of creating a cognitive model by tying together general skills. Secondly, we wanted to create a model of the AB which is capable of capturing the most commonly found effects in the AB-paradigm, including differences due to instruction.

2. Method

Instead of creating operators specifically for the attentional blink, we first considered which general skills are required to perform an AB-task and assembled the AB-model from these skills. In other words, we assembled the model from skills (we assume) participants have already acquired before entering the lab.

Based on previous work and other models of the attentional blink, we identified four basic skills (cognitive processing steps) which had to be performed by our model of the AB. In short, these four skills are visual search, consolidation, retrieve, and report. More extensive discussions of them will follow. We developed these four skills by first creating models of other tasks which share (some) of these same basic skills. This step was done to get a better idea of what these general skills should be capable of and to test the plausibility of these skills.

First, we will describe the three models that provided the building blocks for the AB-model. These three models are: (1) a visual search model, (2) a model of a simple working memory (SWM) task and (3) a model of a complex working memory (CWM) task. Not all parts of all three models will be used for the AB-model, but all three contain at least one of the four basic skills needed to perform an AB-task.

The first model, the visual search model, is very straightforward. The goal of this model is to find a vowel on a screen filled with other letters (see Figure 1). The main search skill processes the current visual item and determines its category through memory retrieval. If it does not match the target category (vowel in this case), it transfers control to another skill which focuses on the next search item. In visual search this is a shift of attention to another item. If it does match the target category, it transfers control to a third skill, in this case a skill that clicks on the target with the mouse. Finally, if it runs out of items to attend to, it transfers control to yet another skill, which is not instantiated in the visual search model.

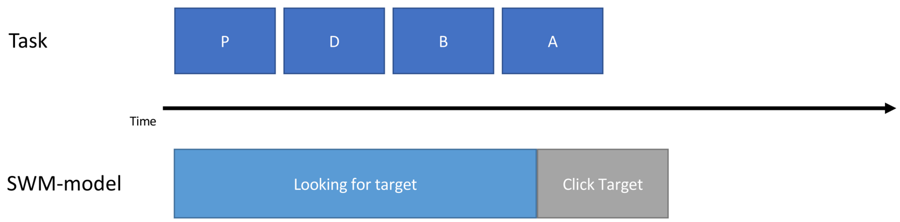

Figure 1. Visual representation of the Visual Search model. The model uses the 'Look for target' skill to discriminate between non-targets (consonants) and the target (vowels) and clicks on the target when it discovers one. The search skill determines the category by a memory retrieval and is reused in both AB-models.

In the AB-model, we will reuse the search skill to find targets, but we will instantiate it differently. To illustrate, Figure 2 lists the operators that make up the search skill, slightly abbreviated for clarity. In these operators V*x* refers to a slot in the visual buffer, RT*x* refers to a slot in the retrieval (declarative memory) buffer, and G*x* refers to a slot in the goal buffer. In these operators, values that are preceded by an asterisk are variables that need to be instantiated for a particular task. For visual search in the context of the previously mentioned example, we instantiate `*fact-type` with `vowel`, `*next-stim` with the `attend-next` skill, and `*after-found-target` with the `click-item` skill.

The second and third basic model are strongly related and provide the final basic skills. Both models deal with working memory tasks which require the participants to remember presented items and, after presentation of the items, recall which items have been seen. Although they both include a consolidation step, they accomplish this step with a different skill. Both build a chunk in working memory, however they differ in the moment of consolidation. The "consolidate-separate" skill, used in the CWM-model, starts consolidation immediately after an item is encountered. In contrast, the "consolidate-chunk" skill, used in the SWM-model, only starts consolidation after all items have been presented. Using these two consolidation skills, we created two versions of the AB-model, a "consolidate-separate" version and a "consolidate-chunk" version.

```
operator look-for-target {
       V1 <> nil  // if there is a visual input
==>
       *fact-type -> RT1  // build a
       V1 -> RT3  // retrieval request
       nil -> V1  // and clear the input
}

operator keep-looking {
       V1 = nil
       RT2 <> *target-type // if it is not a target
==>
       *next-stim -> G1  // change to the skill that
                         // selects the next stimulus
}

operator found-target {
       RT2 = *target-type  // if it is a target
==>
       RT3 -> G8 // Store the target in the goal
       *after-found-target -> G1  // and
             // switch to the skill to handle a target
}
```

Figure 2. Example operators of the visual search skill.

Finally, these two working memory task models provide the "retrieve" skill and the "respond" skill. The "retrieve" skill retrieves the appropriate consolidated item from memory and the "response" skill gives the appropriate response based on the retrieved item, completing the skills needed to create the AB-model (Figure 3).

The four skills described above form the basic building blocks of both versions of our attentional blink model. To finalize the AB-model, the basic skills were put together in one model and were instantiated to fit the context of an AB-trial. This procedure was the same for both versions of the AB-model. In the AB-model, after presentation of a stimulus, the "search" skill checks whether this is a target or a distractor. In other words, the `*fact-type` variable is instantiated with `letter`. If the stimulus is a distractor, it is ignored and the model waits for the next stimulus (`*next-stim` is instantiated with `wait`). If the stimulus is a target it switches to the consolidate skill (by instantiating `*after-found-target` with that skill), which moves the stimulus into a working memory slot.

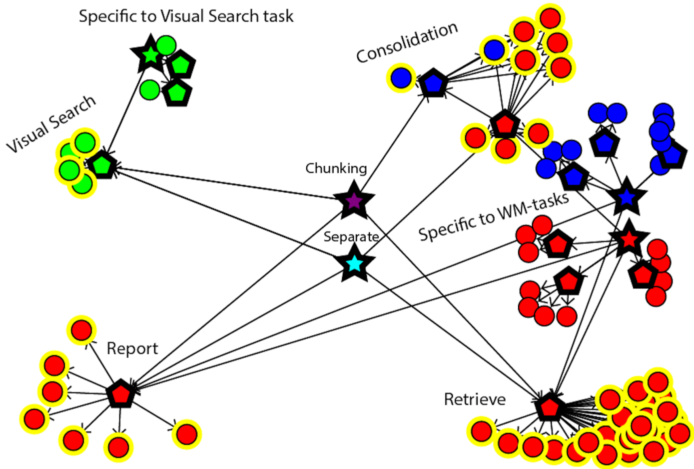

Figure 3. Visual depiction of the AB-models and basic models. The 'consolidate-chunk' AB-model is represented by the purple star and the 'consolidate-separate' AB-model is represented by the light blue star. A yellow circle around a skill indicates that this skill is used in both a basic model and one of the AB-models.

The consolidate skill is the source of the attentional blink in our model. Depending on which skill is used to accomplish consolidation, the model either starts consolidating directly after encountering the first target or postpones consolidation until the second target is encountered. If the chunk is consolidated, no other operator can be executed for a period of, on average, 200 ms (the imaginal delay parameter in ACT-R), leading to a possible attentional blink (see Figure 4). If consolidation is postponed until the arrival of the second target, no attentional blink will occur at this point and the model will keep performing the task normally (see Figure 5). After all stimuli are presented, the model will retrieve the targets that were consolidated on this trial (the "retrieve" skill) and then, after the retrieval, responding to the retrieved items (the "respond" skill).

3. Results

We compared the behaviour of the models with human performance. This was done in order to verify the feasibility of the basic models and to check how well the final AB-model could model the AB phenomenon. The comparisons were made with existing data from the literature, except for the visual search model as we had found no suitable data to compare it with. This is likely due to the fact that our visual search model is very simple and does not have any other functionalities besides what is described in the method section. Furthermore, the visual search model was not our primary interest, as it is not responsible for creating the AB.

For the SWM-model a specific task was modelled, requiring participants to remember a certain number of digits and report them at the end of a stream (Anderson, Bothell, Lebiere, & Matessa, 1998). The critical manipulation in this experiment was that the digits were presented in multiple groups. This grouping was thought to influence chunking of the digits, digits grouped together during presentation would also be grouped together in memory (i.e., chunked together). The findings supported this expectation, such that participants showed longer reaction times during recall for the first item of a group, indicating that the groups were remembered (and recalled) as one chunk. The data from the simple working memory model showed this same pattern in reaction times as reported in Anderson et al. (1998). Just as in the reported data, reaction times in our model to items relating to the start of a new chunk were significantly longer, as tested by linear regression ($\beta = 0.43$, $SE = 0.003$, $t = 128.7$, $p < 0.001$).

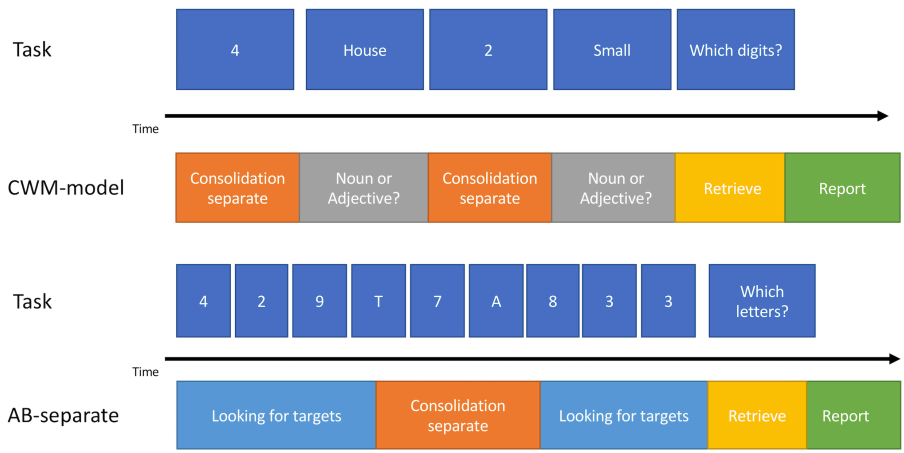

Figure 4. Visual representation of the CWM-model and the "consolidate-separate" AB-model. The CWM-model consolidates the presented numbers separately because working memory is engaged by the secondary task. The "consolidate-separate" AB-model starts consolidation directly after detecting T1 and therefore misses T2 on the lag 2 trial pictured here.

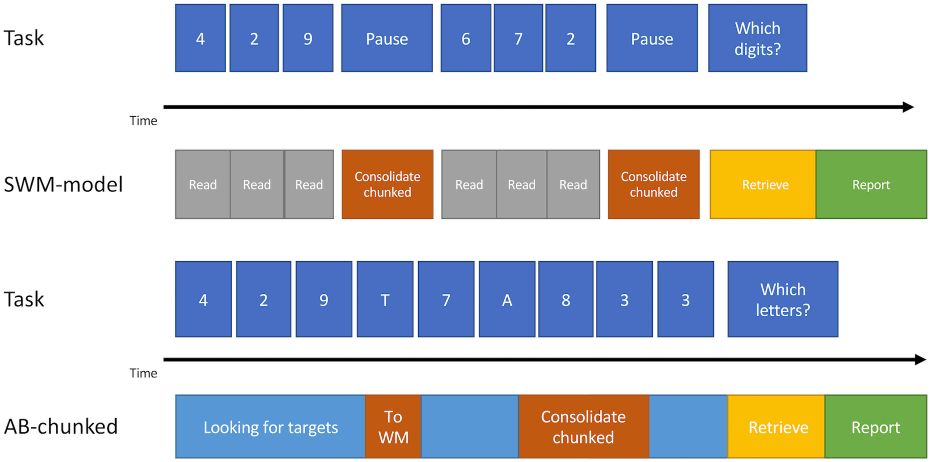

Figure 5. Visual representation of the SWM-model and the "consolidate-chunked" AB-model. The SWM model consolidates the 3 numbers as one chunk. The "consolidate-chunked" AB-model moves T1 into working memory directly after detecting it and only consolidates both targets after detecting T2. This leads to the successful consolidation of both targets even at the Lag 2 trial pictured here.

As can be seen in Figure 6, the reaction times produced by the model show the same typical pattern as the human participants. This reflects the strategy used by the model (and presumably the participants) of recalling the remembered digits. The digits are stored in memory in chunks of three in memory and this influences how the recall occurs. Firstly, the full chunk containing all three digits is retrieved from memory and, subsequently, the three responses are given without any further memory retrieval. Note however that the model is unable to capture the extra-long reaction times at the start of the recall-phase. These increased reaction times are likely due to processes relating to getting started on a new task, an aspect of the task unrelated to working memory so we chose not to model it at this moment. In addition to the reaction times, we also compared the accuracy of our model with the accuracy as reported in the original study (not pictured). The original paper only reports a significant effect of the length of the to-be learned list ($F(9,621) = 128.05$; $p = 0.001$) in the direction that longer lists are harder to recall. Their data also shows a clear effect of serial position, in that earlier presented letters are recalled more accurately than later presented letters (although there is also a small recency effect). We tested our model on these same two effects of list length and serial position with a linear mixed effect model. Our AB-model shows a significant effect of both list length ($\beta = 0.05$, $SE = 0.01$, $z = 3.7$, $p < 0.001$) and serial position ($\beta = 0.13$, $SE = 0.04$, $z = 3.4$, $p < 0.001$) which is in line with the data presented in the original study.

The SWM-model data was collected over 15 runs with 350 trials per run (total of 5250 trials). Most of the parameters were the default PRIMs parameters (which are the same as the ACT-R default parameters for a big part). Only the retrieval threshold (rt was 0.6 instead of 0) and the latency factor (lf was 0.15 instead of 1.0) differed from default. This latency factor of 0.15 was also used in a previous model of the attentional blink (Taatgen et al., 2009).

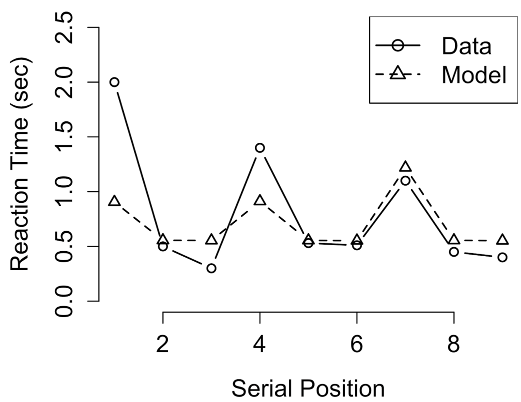

Figure 6. Model fit for reaction times in the SWM-task. Figure depicts the RTs produced by the model (dashed line) and human data (solid line).

Secondly, we will discuss the comparison between the CWM-model and human performance. In the task we modelled, a series of 3, 4, 5, or 6 digits were presented to the model. In between presentation of the digits, the model did a word-decision task in which it had to distinguish between nouns and adjectives. We compared the performance of our model on this task to a similar experimental task (Daily, Lovett, & Reder, 2001). In this task, participants were instructed to remember a series of digits (also 3 to 6), but here the digits were presented among letters which they were required to read aloud. Both of these tasks have in common that working memory is required to perform the interrupting task (either deciding between a noun or adjective or reading a letter aloud). This demand on working memory makes it impossible for the participants (and the model) to chunk the items in memory.

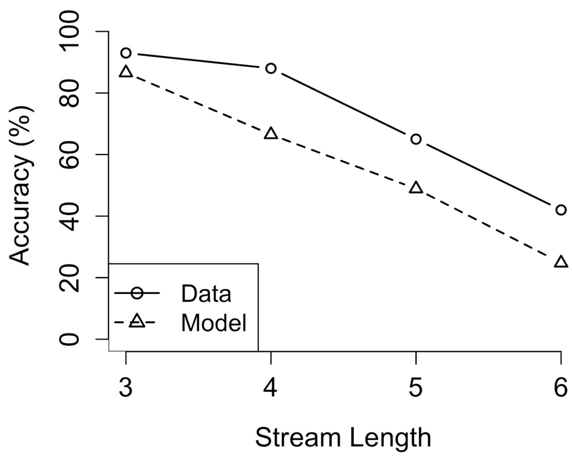

Figure 7. CWM-model fit for accuracy data. The average accuracy as a function of list length for the model (dashed line) and the human data (solid line).

We compared model performance with human performance with respect to accuracy (see Figure 7). This was the only measure we could use because the original paper did not report any other measure (e.g., reaction times). Generally, the model shows a good fit to the human data reported by Daily et al. (2001). Both the model and the participants show decreased accuracy when the length of the presented list is longer. In both the original data ($F(3, 63) = 90.80$, $p = .0001$) (tested with an ANOVA) as in our model ($\beta = -0.2$, $SE = 0.005$, $t = -37.2$, $p < 0.001$) (tested with a general linear model) this was a strongly significant result. This decreased accuracy for longer lists occurs in the model because the presentation of the longer lists takes a longer time to be completed. The longer time required for presentation allows for additional item-decay in memory, leading to reduced accuracy for longer lists. The model, however, generally underestimates accuracy, this is probably due to the model being unable to capture the primacy effect (Murdock, 1962). The primacy effect is often modelled by including a rehearsal mechanism. The fact that we did not include such a mechanism to the model could thus explain the general underestimation of the accuracy.

Rehearsal is not directly related to working memory consolidation, so for reasons of simplicity we did not include this process in the model. The SWM-model data consists of a total of 4500 trials (15 runs with 300 trials per run). Similar as with the CWM-model most of the parameters were kept at the default setting, except for the retrieval threshold and the latency factor. The retrieval threshold had a value of 0.5 and the latency factor was set at 0.15 for our model.

Finally, we compared our AB-model (which resulted from the combination of the above discussed models) with human AB performance (see Figure 8). The specific task we modelled was the classic AB task reported in Chun & Potter (1995). In this standard version of the AB, participants are instructed to identify two digits within a stream of distracting letters and, at the end of the stream, report which digits they have seen. We modelled this experiment with the version of the AB-model that used the "separate-consolidation" skill. The attentional blink, characterized by a strong performance decrement at lags 2 and 3, is nicely captured by our AB-model. Performance at Lag 2 ($\beta = -0.47$, $SE = 0.02$, $t = -25.2$, $p < 0.001$) and Lag 3 ($\beta = -0.41$, $SE = 0.02$, $t = -22$, $p < 0.001$) is significantly lower than at Lag 1. The original paper also reports significant performance decreases at Lag 2 ($t(5) = 6.6$, $p < 0.01$) and Lag 3 ($t(5) = 4.1$, $p < 0.01$). In the model, the AB occurs because consolidation of the first target (T1) is still in progress when the second target (T2) is presented. Therefore, T2 cannot be consolidated and will not be reported at the end of the stream. Our model also shows the typical lag-1 sparing effect. This is because consolidation of T1 often has not started at the moment that T2 is presented at lag 1. Therefore, they can both be consolidated into a single chunk and reported at the end of the stream. Finally, the model shows the slow performance increase for the later lags (lag 4 and higher). This is caused by the slow increase of the likelihood that T1 consolidation is finished by the time T2 is presented.

The AB-model data consists of 15 runs with 600 trials per run (for a total of 9000 trials). Most of the parameters were kept at the default values, except for the latency factor, activation noise, and the imaginal delay factor. The latency factor was again set to 0.15, following (Taatgen et al., 2009). The activation noise was set to 0.35, which is a little bit higher than default, this was done in order to create some extra noise on the behaviour of the model and produce slightly more human-like performance. Finally, the time it took for a chunk to be encoded into the imaginal buffer, the imaginal delay parameter, varied from trial to trial (also following Taatgen et al., 2009). Its value ranged from 50 ms to 350 ms and was drawn from a uniform distribution with an average of 200 ms. This variation made it possible for the model to still detect both targets on lag-2 and lag-3 trials in some instances.

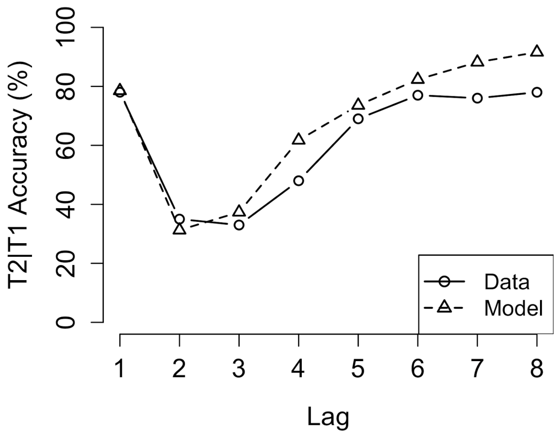

Figure 8. AB-model fit for T2 accuracy. Figure showing T2 accuracy in an AB-task for the model (dashed line) and human data (solid line).

Using the other version of the consolidation skill (the "consolidate-chunk" version) in the AB-model, however, will prompt the model to always try to consolidate both targets into a single chunk, which should prevent the AB to occur. Importantly, both versions of the AB-model were run with exactly the same model parameters. We compared the performance of the AB-model instantiated this way to the data from Experiment 2 in the paper reporting a reduced AB when participants were instructed in a way that promoted chunking (Ferlazzo et al., 2007) (see Figure 9). The model mirrored the general performance level and, crucially, showed no AB. Performance on lag 2 ($\beta = 1.76$, $SE = 0.15$, $z = 11.6$, $p < 0.001$) and lag 3 ($\beta = 1.\ 6$, $SE = 0.14$, $z = 11.6$, $p < 0.001$) was significantly higher in the "consolidate-chunk" version compared to the "consolidate-separate" version. The model, however, shows a slight performance decrease at lag 1 which is significantly lower than lag 6 performance ($\beta = -0.13$, $SE = 0.01$, $t = -2.1$, $p < 0.001$). This was caused by the way in which noise in the visual system was simulated, which meant that occasionally T2 had already disappeared before it was processed fully and therefore it was missed. We do not consider this problematic, because in many AB experiments lag 1 performance is slightly lower than performance on long lags. The “consolidate-chunk” version model data also consists of 15 runs with 600 trials per run (in total 9000 trials). As was mentioned before, the parameters were exactly equal to the parameters used for the “consolidate-separate” version of the model.

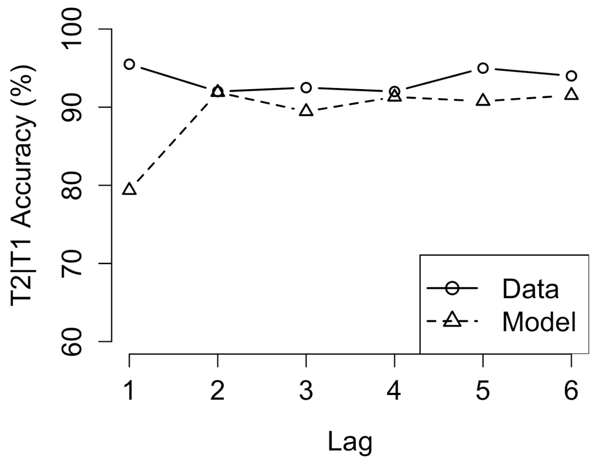

Figure 9. Model fit for the alternative AB model (consolidate-chunk). Figure showing T2 accuracy for the alternative AB model (dashed line) and human data (solid line).

4. Discussion

Computational models of cognitive psychological phenomena are often able to accurately capture one specific phenomenon, however they are often hard to generalize to other tasks and cognition in general (Anderson et al., 2004). In this paper, we attempted to (partly) bridge this gap by employing a novel approach to building cognitive models, which mirrors the way people approach a new task. People do not consider every task in isolation but they use knowledge gained from the past. That is, they reuse skills learned from doing other tasks and apply them to the (new) task at hand (Salvucci, 2013; Taatgen, 2014). This paper describes our attempt to apply a similar approach. However, our approach differs from earlier efforts, because the "building blocks" for a task are much larger. In the earlier work, the level of reuse was operators. Therefore, a new task had to be constructed out of existing operators, which is, even in the case of a simple task such as the attentional blink, still a substantial number (see Figure 3). In the skill-based approach, only four skills needed to be combined. This is not only consistent with the amount of time people need to prepare for the task, but it also sheds new light on the nature of the attentional blink, and the effect of instruction.

The comparisons between our models and human data show that our models are reasonably able to capture human performance. This result demonstrates the basic feasibility of the described modelling approach. It is possible to break a task down into a limited set of skills that are reusable in different tasks. This is an important first step towards creating more generalizable models, because it allows for a method of creating models that are built from the same building blocks. Using existing building blocks when modelling a new task allows for much more integration of any new model into the already existing collection of models and might better reflect the way people approach a new task. Finally, our AB-model can be placed in a larger cognitive context. It provides a clear and generalizable explanation of the AB using terms that can be related to other theories and empirical findings. This is the main benefit of creating models with the skill-based approach. In our case, our model not only provides an account of the AB but, importantly, also suggests some general limitations to memory consolidation. The skill-based approach facilitates the investigation of the mechanisms that lead to experimental findings instead of only focusing on the findings themselves.

Note, however, that the devil is in the details. Building a model using this approach can be challenging, especially when it comes to determining how small differences between tasks can best be handled. Such differences make it difficult to use exactly the same operator (and therefore the same skill). Every operator has a condition-checking part (which checks whether this operator should be activated now) and an action-performance part (which actually executes the 'cognitive action' or PRIM). The action-performance part is relatively easy to generalize across tasks, but the condition-checking part is more challenging. Basically, the condition-checking part checks whether the situation matches the predefined situation in which this operator should be executed. This makes it difficult to generalize the condition-checking across tasks since a different task usually also means a different situation. We solved this problem in the models described in this paper by defining the conditions in such a way that they work for all the modelled tasks. This is a workable solution, but it is time-consuming and a more optimal method for condition-checking is needed.

A further limitation to the models described here is that they did not perfectly capture all aspects of human performance. However, we do not see this as a major issue because we did not set out to create complete models of the described experimental paradigms. Instead we aimed to create models of the main findings only because we were merely interested in the skills that are important for the AB. Although there remain limitations and improvements to be made to the skill-based approach, we consider it a feasible and promising approach to improve the generalizability of models.

The second goal we set out to achieve in this paper was to create a model of the AB that can account for differences due to instruction. The model described in this paper produces most of the basic effects from the classic AB-task, showing lag-1 sparing, the AB itself and the gradual improvement on later lags. Although there are many additional aspects of the AB reported in the extensive literature which we did not discuss, we believe that the model described here is an adequate first attempt that can be built on in future work.

For now, the fact that the model captured the basic AB effects implies that these effects, at their core, may be caused by improper selection of skills. At the start of a new task, a participant has to figure out which skills to combine in order to be able to perform the new task. The models we created suggest that there are (at least) two different skills which can take care of the consolidation into working memory aspect of the task: (1) consolidate every presented target into working memory separately (as in the CWM-task) or (2) consolidate targets as larger chunks (as in the SWM-task). The chunk-consolidation skill as used in the SWM-task would be the optimal pick in this situation, two items can be consolidated into one chunk and there would be no negative unexpected effects. This approach is perhaps employed by participants after receiving the experimental instructions from the Ferlazzo et al. (2007) study. However, given that standard AB instructions consider targets as separate items probably prompts most participants to use the separate-consolidate skill from the CWM-task.

The emphasis put on strategy by our model could explain previous findings in the AB literature that have proven difficult to explain. This includes the effect of instructions as well as the existence of non-blinkers (individuals who do not show an AB) (Martens, Munneke, Smid, & Johnson, 2006; Willems & Martens, 2016; Willems, Wierda, van Viegen, & Martens, 2013), and the reduction of AB-magnitude because of training (Choi, Chang, Shibata, Sasaki, & Watanabe, 2012). All these effects could be explained by the type of consolidation strategy. Different instructions might cue the ‘correct’ consolidation skill, non-blinkers could be more naturally inclined to use the ‘correct’ chunking strategy compared to blinkers, and the training procedure by Choi and colleagues might nudge participants toward using the same optimal strategy.

To summarize, our novel skill-based approach to cognitive modelling resulted in valid models, created using a more natural and human-like method. In addition, we believe it shows great potential to generate more generalizable and thus more flexible models. Therefore, we will continue working on the skill-based approach, by testing the underlying assumptions (e.g., people are able to apply previously learned skills to new tasks) and creating additional models using this approach for other paradigms. Finally, building models with the skill-based approach can lead to interesting new perspectives on well-established cognitive phenomena such as the AB. The choice of consolidation strategy may play an important role in the AB, explaining individual differences as well as instruction and training effects of the AB.

3

Testing the skill-based approach: consolidation strategy impacts attentional blink performance

Humans can learn simple new tasks very quickly. This ability suggests that people can reuse previously learned procedural knowledge when it applies to a new context. We have proposed a modelling approach based on this idea and used it to create a model of the attentional blink (AB). The main idea of the skill- based approach is that models are not created from scratch but, instead, built up from reusable pieces of procedural knowledge (skills). This approach not only provides an explanation for the fast learning of simple tasks but also shows much promise to improve certain aspects of cognitive modelling (e.g., robustness and generalizability). We performed two experiments, in order to collect empirical support for the model's prediction that the AB will disappear when the two targets are consolidated as a single chunk. Firstly, we performed an unsuccessful replication of a study reporting that the AB disappears when participants are instructed to remember the targets as a syllable. However, a subsequent experiment using easily combinable stimuli supported the model's prediction and showed a strongly reduced AB in a large group of participants. This result suggests that it is possible to avoid the AB with the right consolidation strategy. The skill-based approach allowed relating this finding to a general cognitive process, thereby demonstrating that incorporating this approach can be very helpful to generalize the findings of cognitive models, which otherwise tends to be rather difficult.

This chapter has previously been published as:

Hoekstra, C., Martens, S., & Taatgen, N. A. (2022). Testing the skill-based approach: Consolidation strategy impacts attentional blink performance. Plos one, 17(1), e0262350.

1. Introduction

People rarely encounter a task that shares no similarities with tasks that have been done before. A smart strategy, therefore, would be to analyse which components of a task have been done before and which components are new, in order to be able to focus on the novel and more challenging aspects of a task. Such an approach could explain why people can learn certain simple new tasks with impressive speed. It can also explain why performance on some tasks is suboptimal: not because they are incapable of optimal performance, but because combining elements from a prior learning history leads to a suboptimal strategy. Furthermore, it may be a crucial aspect of human cognition and be the reason why human behaviour is so flexible and reliable.

The attentional blink (AB) paradigm is a good example of a simple but novel task. The objective for the participants in an attentional blink experiment is to identify and remember two targets amongst a stream of (to-be ignored) distractors. Although the task is fairly challenging given that items are presented at a rate of about 10 per second, participants only need to distinguish the targets from the distractors, remember the targets and report them at the end of the stream. Therefore, participants need to be provided with very limited instructions and require little time to execute this task properly. This impressive speed suggests that people can use skills they learned outside of the current context and apply them to the new context (Salvucci, 2013; Taatgen et al., 2008). In the example of the AB, in order to be able to remember the target stimulus the participants do not need to figure out from scratch how to remember a stimulus, they can simply use some sort of ‘remembering’ skill they already possess and use it to accomplish the AB task demands. This supports the notion that learning most new tasks does not require any new knowledge, it simply requires combining existing knowledge in a novel way.

In a previous paper (Hoekstra, Martens, & Taatgen, 2020) we created a cognitive model of the attentional blink (AB) based on this idea. This model was able to capture the most important aspects of the data reported in this paradigm and achieved this result using a novel and more human like modelling approach. This approach could provide certain improvements to areas in which models are currently lacking, which we will describe in the following section. The main objective in the current paper is to collect empirical evidence for the predictions made by this AB model on variations of this task. However, for reasons of clarity and completeness, we will first restate the reasoning behind the approach used to create the model and its contributions as well as the theoretical foundation of the AB model.

1.1. The Skill-Based Approach

At its core, the skill-based approach is a theory that explains the fast learning people are capable of when presented with a simple new task. This finding is hard to explain with cognitive models because they either require a long training session or a large amount of task-specific procedural knowledge specified by the modeler. The skill-based approach offers an explanation that does not require either. It does this by assuming that people can apply previously learned procedural knowledge, represented as skills, when these skills are useful in the context of the new task. Skill is a common word in the literature and has been used to refer to many different concepts. In this paper, skill refers to a collection of procedural knowledge that accomplish a certain general processing step and that can be used in multiple tasks. In our models, skills are represented by a set of operators (i.e., production rules) that can be used in different contexts by instantiating the variables depending on the context. Interestingly, when this theory is computationally implemented in this way, additional benefits to modelling in general become apparent because the strategy people employ to facilitate this type of fast learning (reusing of skills) seems to be a crucial element of human cognition underlying more characteristics of human behaviour that have previously been difficult to capture by cognitive models.

The most striking example of such a characteristic is the impressive behavioural flexibility people possess. People are capable of performing a wide range of tasks while cognitive models developed to mirror this behaviour are generally only capable of performing the specific task they are modelling. Presumably, this disparity is caused by the fact that people use general and reusable skills while cognitive models usually do not. In addition to underlying flexible and robust behaviour in humans, the possibility of skill reuse also strongly limits the burden on procedural memory. Instead of a different skill for every different context (e.g., a different skill for remembering names and remembering capitals), only a limited set of reusable skills needs to be stored (e.g., the same ‘remembering’ skill is used for names and for capitals). In summary, reusing skills allows people to behave in an efficient, flexible, and reliable way (Taatgen et al., 2008) and translating this strategy to cognitive modelling would also allow cognitive models to behave in a more efficient, flexible, and reliable way.

Additionally, the skill-based approach is an important addition to the value of cognitive architectures. Cognitive architectures are general modelling frameworks in which a large variety of tasks can be modelled (see e.g., (Kotseruba & Tsotsos, 2020) for an overview of cognitive architectures). Using a cognitive architecture to create a model offers two important advantages: (1) models will have a high level of cognitively plausibility since they are created within an empirically supported architecture and (2) models will be highly generalizable since they all operate within the same basic architecture. We believe that these two core aspects of cognitive architectures can be strongly improved by considering skill reuse when creating a new model (e.g., by applying some principles of the skill-based approach). Models that are created with such an approach will be more plausible since the pieces of procedural knowledge (i.e., the skills) are validated by using them in other (similar) tasks (Hoekstra et al., 2020) and they cannot be (implausibly) (task-) specific since they need to be general enough to be reused. This will also increase the generalizability of a model since not only will the basic structure of human cognitive system be considered (the "architecture") but also how this system is used (the procedural knowledge).

Besides being important for the design of cognitive architectures, the skill-based approach may also improve the results of modelling efforts that do not make use of a cognitive architecture. Cognitive modelling is a tool used in many different fields answering widely varying questions. This presents a challenge with integrating this multitude of models into a single theory of cognition (Anderson et al., 2004). Possibly, the skill-based approach could aid integration of the many models because it allows researchers to more easily relate the mechanism they are modelling to the general field by explicitly defining the modelled mechanism as part of a more general cognitive process. Additionally, the skill-based approach could support the creation of a collection of skills from which modelers can draw from when building new cognitive models. This would be a huge step towards increasing the consistency between models of similar tasks based on the notion that models that perform the same processing steps should accomplish these processing steps in the same fashion (i.e., with the same skill).

We brought the skill-based approach into practice by creating a model of the attentional blink (AB) (Hoekstra et al., 2020). This model was created using the cognitive architecture PRIMs (Taatgen, 2013, 2014). PRIMs (which stands for primitive information processing elements) is based on ACT-R and has many of the same basic characteristics (Anderson, 2007; Anderson et al., 2004). Cognitive processing in both PRIMs and ACT-R revolves around information exchange within the central workspace by several modules capable of performing specific cognitive functions (e.g., the visual module is capable of visual processing). This way, a cognitive system gets built up from the modules capable of performing specific actions that can share the results of their actions with each other through the central workspace. The modules communicate in such a way that the result of the cognitive actions performed by one module can serve as input for the other modules. This allows for models to be created that are capable of performing a task from start to finish in many different fields (Salvucci, 2006; Taatgen, Van Rijn, & Anderson, 2007; Van Rij, van Rijn, & Hendriks, 2012). The communication between the modules is controlled in PRIMs and ACT-R in largely the same way. In ACT-R this is done by productions and in PRIMs this is accomplished by operators, but they have generally the same functionality. A crucial advantage of using PRIMs over ACT-R is that PRIMs allows for operators (i.e., productions in ACT-R) to be organized into skills. A skill is a collection of operators that, combined, are capable of achieving a certain well-defined cognitive processing step within one model while still being general enough to be reused in other models. The generalizability of skills allows for the same skills to be used in multiple models independent of which exact task is modelled. Additionally, the PRIMs architecture was developed with the intent of breaking up the relatively task-specific processing steps of ACT-R into more elementary and general steps. These elementary processing steps (the PRIMs) are central to the PRIMs architecture and facilitate creating reusable skills because the skills themselves are made up from general and elementary processing steps. This is the main reason why the PRIMs architecture is well suited for creating models based on skill-reuse.

1.2. Skill Selection in the Attentional Blink

Reusing skills is a vital part of human cognition and has many advantages. However, in some cases, this reuse of skills might have unintended negative consequences. Skills that work perfectly in some tasks might lead to sub-optimal performance in other (but highly similar) tasks even though the cognitive system is, in principle, capable of perfect performance (Taatgen et al., 2009). The Stroop effect (Stroop, 1935) might be the most famous instance of such sub-optimal skill selection. People are so used to reading, that the ‘reading’ skill is automatically triggered even when the task is to identify the colour of a word (e.g. “red”) rather than naming the word (e.g. the word “blue”), resulting in prolonged RTs in case of mismatch.

The attentional blink (AB) could be the inadvertent result of a comparable situation. The AB is an intensively studied paradigm in cognitive psychology (Dux & Marois, 2009; Martens & Wyble, 2010). It refers to the finding that the second of two targets (referred to as T2) is often missed when it is presented in an interval of 200-500 milliseconds after the first (referred to as T1). However, when T2 is presented directly after T1, performance is not impaired and participants are able to identify T2 correctly most of the time. This Lag-1 sparing shows that people are, in principle, capable of remembering both targets, but that sub-optimal skill selection might lead to the performance impairment in identifying the second target for somewhat longer lags (e.g., at lags 2 or 3).

The crucial component of the sub-optimal performance may lie in the selection of the particular skill that accomplishes the consolidation of the targets in memory. Although there is no consensus on the exact mechanism behind the AB, memory consolidation has frequently been implicated to play a major causal role in this process (Akyürek et al., 2011) and many theories hold memory consolidation as the main factor underlying the AB (Bowman & Wyble, 2007; Chun & Potter, 1995; Jolicoeur & Dell’Acqua, 1998; Shapiro, Raymond, & Arnell, 1994; Taatgen et al., 2009). These theories differ in the fine details, however they all assume that the AB is the result of a similar two-stage process. This includes a first stage in which stimuli can be processed in parallel followed by a second stage in which only one stimulus can be consolidated into memory at the same time. According to these theories, this serial consolidation process forms the bottleneck responsible for the AB when T1 and T2 are presented in close temporal proximity but not immediately following each other. In these cases, T2 has to wait for T1 to be consolidated and, therefore, runs the risk of being overwritten in short-term visual memory by the following masking distractor, preventing T2 from being consolidated (Giesbrecht & Di Lollo, 1998; Seiffert & Di Lollo, 1997).

Although the attentional blink is often conceived as a fundamental limit to human processing, several studies have reported various categories of manipulations that have led to substantial reductions and sometimes even complete eliminations of the AB, suggesting that the AB does not reflect a structural bottleneck. Some of these studies have manipulated the stimuli directly, e.g., the AB completely disappears when then the T2 is the participant's own name (Shapiro, Caldwell, & Sorensen, 1997), however significant AB reductions have also been reported without manipulating the stimuli. One line of studies manipulating participants' attentional engagement in the task have reported a counter intuitive improvement to AB performance when participants were focused less on the primary AB task. These manipulations include playing music to the participants and encouraging them to be distracted while performing an AB task (Olivers & Nieuwenhuis, 2005), having the participants perform a concurrent secondary task (Taatgen et al., 2009), and distracting the participants with task irrelevant motion and flickering (Arend, Johnston, & Shapiro, 2006). Additionally, a specific type of training has been shown to be beneficial for AB performance (Choi et al., 2012) and, finally, the existence of non-blinkers (individuals who do not display an AB) further questions the fundamental nature of the AB (Martens et al., 2006; Martens & Wyble, 2010; Willems & Martens, 2016).

These manipulations reducing the AB without changing the stimuli strongly imply that strategy plays a crucial role in generating (and eliminating) the AB. We operationalize strategy as sub-optimal skill selection in this paper. The manipulations may have led to a reduced AB by successfully changing which strategy (and therefore which skill) participants were using to consolidate the targets into memory. It is our hypothesis that, instead of consolidating both targets separately, participants have been cued to consolidate both targets as a single chunk following these manipulations. This is similar to an earlier account, which states that the less engaged participants are unable to exert sufficient cognitive control to start consolidating immediately after encountering the first target (Taatgen et al., 2009). The training may nudge participants towards using the chunked consolidation strategy and the non-blinkers might instinctively (or accidentally) employ this strategy. The crucial consequence of using the strategy of consolidating both targets as one chunk is that the bottleneck of stage-two processing as described earlier would not occur. Instead of consolidating T1 into memory as soon as it is identified (and therefore preventing T2 from being consolidated), T1 consolidation is postponed until T2 has been identified as well and both targets are consolidated together in one chunk.

Concrete evidence for the effect of strategy on AB performance was provided by an experiment performed by Ferlazzo and colleagues (Ferlazzo et al., 2007). This paper reported the results of an experiment in which participants were either instructed to report the presented targets (which were always a vowel and a consonant) as two separate letters (the standard AB instructions) or to report them as a syllable. Interestingly, participants did not show an AB in the latter syllable condition. A possible explanation could be that participants in the syllable condition adopted a chunking consolidation strategy and thereby avoided the AB bottleneck, whereas the participants in the separate condition adopted the standard separate consolidation strategy and thus fell into the AB trap.

1.3. Modelling the AB using the Skill-Based Approach

In the paper mentioned at the start of the introduction (Hoekstra et al., 2020) we investigated this effect of strategy on the AB by creating two versions of a cognitive model of the AB that only differed in their consolidation skill. The “consolidate-separate” version of the model consolidated the two targets as separate chunks into memory and the “consolidate-chunked” version of the model consolidated the two targets as a single chunk into memory. The model was created using the skill-based approach, which meant that instead of creating a model specifically for the AB, we composed the model from skills taken from other models. The skills used in both versions of the model were mostly identical, the only difference was that the consolidation skill used by the “consolidate-separate” version was taken from a model of a complex working memory task (in which participants consolidate every target separately) while the consolidation skill used by the "consolidate-chunked" version was taken from a model of a simple working memory task (in which participants consolidate multiple targets as a single chunk).

Both versions of the AB model were capable of capturing the data reported in the literature (see Figure 1). The "consolidate-separate" version of the model successfully showed the most important aspects of data reported in the AB literature: Lag-1 sparing, the AB itself and the steady performance increase on the later lags (Figure 1a). The "consolidate-chunked" version of the AB model, crucially, does not produce an AB. This version of the model avoided the AB because memory consolidation is accomplished by a different consolidation skill. Instead of consolidating a single target into memory as soon as it is encountered, this consolidation-skill postpones consolidation until both targets have been detected and consolidates both targets as a single chunk into memory. We hypothesized that the syllable instruction condition of the study by Ferlazzo and colleagues prompted this consolidation strategy and, therefore, allowed the participants to bypass the AB. The "consolidate-chunked" version of the model indeed showed a good fit with the data reported in the syllable condition of the study by Ferlazzo and colleagues (Ferlazzo et al., 2007) (Figure 1b). Crucially, the only difference between both versions is the consolidation-skill, all other aspects of the model (e.g., model parameters) were held equal.

1.4. Current study

The skill-based approach to model the AB did not only result in a model capable of capturing important aspects of the data reported in the AB literature but also constitutes a more naturalistic and human like modelling approach. This promising first step provided basic evidence for the potential of the skill-based approach. In the current study, we will attempt to collect additional empirical evidence for the central prediction of the AB model that the employed consolidation strategy greatly impacts AB performance.

We will do this by first attempting to replicate the previously mentioned study conducted by Ferlazzo and colleagues (Ferlazzo et al., 2007). Secondly, we will perform an experiment centred around the same prediction using different targets than commonly used in AB studies. These two experiments allow us to test the effect of consolidation strategy on AB performance. Additionally, it provides an opportunity to test the flexibility of the models created by the skill-based approach, because the task demands of the second experiment are slightly different compared to the original AB task, requiring small adjustments to the model.

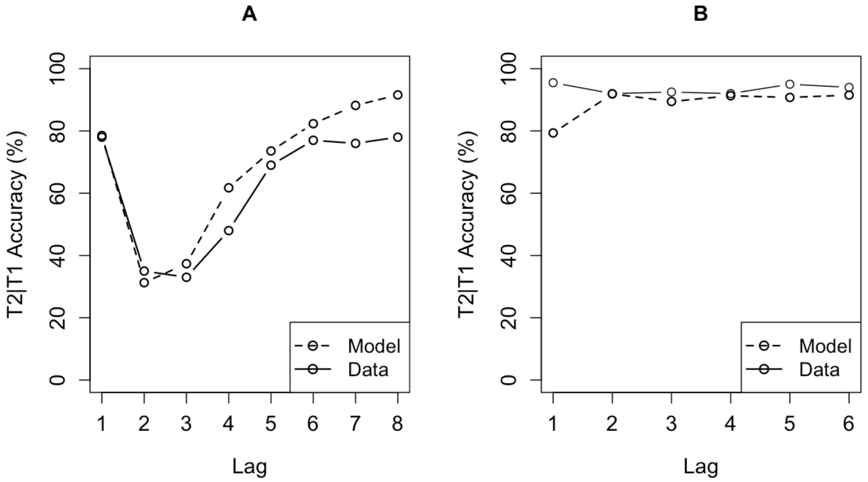

Figure 1. Model fits for both versions of the AB model. (a) the “consolidate-separate” model with data reported in a classic AB study (Raymond et al., 1992). (b) the “consolidate-chunked” version of the AB model with the data reported in Ferlazzo et al. (2007).

2. Experiment 1

Experiment 1 was a replication attempt of the experiment in the study conducted by Ferlazzo and colleagues (Ferlazzo et al., 2007) in which the targets (a vowel and a consonant) had to be reported as a single syllable. The goal was to verify the original findings and to create a complete data set which would allow for a more detailed model fit. Although we did not carry out an exact replication of the experiment reported in the original paper, the results should be comparable since the crucial manipulation was identical in both studies.

2.1. Method

2.1.1. Experimental setting

The research took place in the lab of the Artificial Intelligence department of the Bernoulli Institute at the University of Groningen. The experiment was run and programmed with OpenSesame (Mathôt, Schreij, & Theeuwes, 2012) using the backend PsychoPy (Peirce et al., 2019) on a MacOS computer. The participants were seated approximately 0.5 meter away from the computer screen, which was a 24 inch LCD Benq XL2420-B with a refresh rate of 60 Hz.

2.1.2. Participants

All 18 participants (10 female, aged 18 to 25, mean = 21.2 years) who took part in the experiment were students of the University of Groningen and received a financial compensation of 8 euros for their participation. The sample size was based on the large effect size reported by Ferlazzo and colleagues (Ferlazzo et al., 2007). Additionally, a power analysis was performed utilizing the method provided by (Chow, Wang, & Shao, 2007). This analysis was based on the effect size and standard deviation reported by the original study and indicated that 14 was the recommended sample size for those values. Finally, prior to the experiment, participants signed an informed consent form and ethical approval was acquired from the Research Ethical Review Committee of the University of Groningen.

2.1.3. Stimuli

On every trial a sequence of twenty stimuli was presented consisting of 18 distractor stimuli (digits) and 2 target stimuli (letters). The distractors could be any digit with the exception of 1, 5, and 9. They were randomly drawn with the single rule that two subsequent distractors could not be identical. The targets on every trial consisted of a vowel-consonant pair (creating a syllable). The order of presentation was random but the frequency of appearance was balanced; on half of the trials the vowel was presented first (i.e., as T1), on the other half of the trials the consonant was presented first. The vowel on every trial was randomly drawn from a collection of four vowels: ‘A’, ‘E’, ‘I’, and ‘U’. The consonant on every trial could be any consonant (except for ‘S’, ‘Q’ or the semi-vowel ‘Y’). The stimuli were presented in white at the centre of the screen on a black background. The font used for both the targets and distractors was ‘droid sans mono’ which is the default font used by OpenSesame. The size of both the distractors and the targets was about 1° of visual angle.

2.1.4. Procedure

The experiment was set up with a between-subjects manipulation of instruction. The participants were instructed to either remember the two target letters as two separate letters (the ‘separate’ condition) or as a single syllable (the ‘syllable’ condition). Participants in either condition were not aware of the other condition. Participants were randomly assigned to a condition, which due to a technical oversight led to a small imbalance between the conditions. At the end of data collection, 10 participants had been assigned to the ‘syllable’ condition and 8 participants had been assigned to the ‘separate’ condition. Additionally, the serial position of the second target (T2) relative to the first target (T1) was varied (referred to as lag). The lags included in this study were lag 1, 2, 3, 4, 5, and 6. All 6 lags were presented 70 times (420 experimental trials in total).

Before the experiment started, participants received a short verbal instruction from the experimenter. This verbal instruction was given in addition to further written instructions on the computer screen which participants could read at their own pace. The verbal instructions were given because participants’ understanding of the instructions was a crucial part of the experimental manipulation. Finally, participants performed 18 practice trials, 3 trials per lag. The participants received feedback about their performance during the practice trials, but they did not receive any feedback during the experimental trials.

All trials in the experiment consisted of a rapid serial visual presentation (RSVP) stream containing 20 items presented at a rate of 10 Hz (see Figure 2). The RSVP stream was always preceded by a fixation cross presented for one second in the centre of the screen. The stimulus onset asynchrony (SOA) in this study (100 ms) was longer than was reported in the original study (80 ms). This was decided after an unsuccessful pilot study with low accuracy indicating that an SOA of 80 ms was too fast in the current design. One reason for this discrepancy with the original study could be that the targets were more easily distinguishable from the distractors in the original study (e.g., because of the font), but this cannot be verified from the reported information. Finally, at the end of every trial, two white dots appeared in the centre of the screen prompting the participants to type in the two letters they had seen during the RSVP stream. This response screen was identical in both conditions. The participants provided their answers with the letter keys on the keyboard without time pressure. Participants required approximately 45 minutes to complete the experiment.

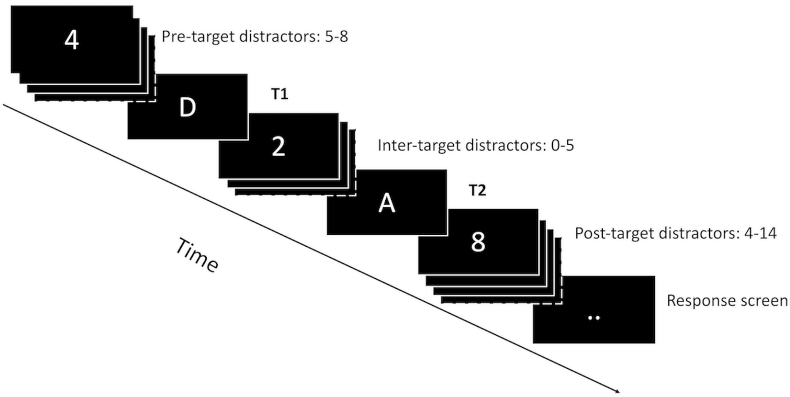

Figure 2. Schematic representation of a trial in Experiment 1.

2.2. Results

The main goal of Experiment 1 was to replicate the original findings as reported by Ferlazzo and colleagues (Ferlazzo et al., 2007) regarding the effect of instruction on AB performance. They reported a strongly reduced AB in the 'syllable' condition. We conducted a highly comparable experiment and thus expected that the participants in the 'syllable' group would show a strongly reduced AB while the participants in the 'separate' group would show a standard AB.

However, the data do not support this hypothesis. As can be seen in Figure 3a, there were no large differences in T2|T1 accuracy between the two conditions at any lag. T2|T1 accuracy refers to the T2 accuracy on trials in which T1 was correctly identified. The largest difference exists at Lag 2 where average performance in the 'syllable' condition was slightly higher than in the 'separate' condition. However, this difference was not statistically significant as tested with a logistic linear mixed effects regression model (β

= 0.22, SE = 0.42, z = 0.5, p = .59). Furthermore, the data from our 'syllable' condition differs strongly from the original data and does not show the strong AB reduction reported in the original study. In short, the data does not support the conclusion that the AB magnitude differs significantly between the two instruction conditions and thus failed to replicate the findings of the original study.

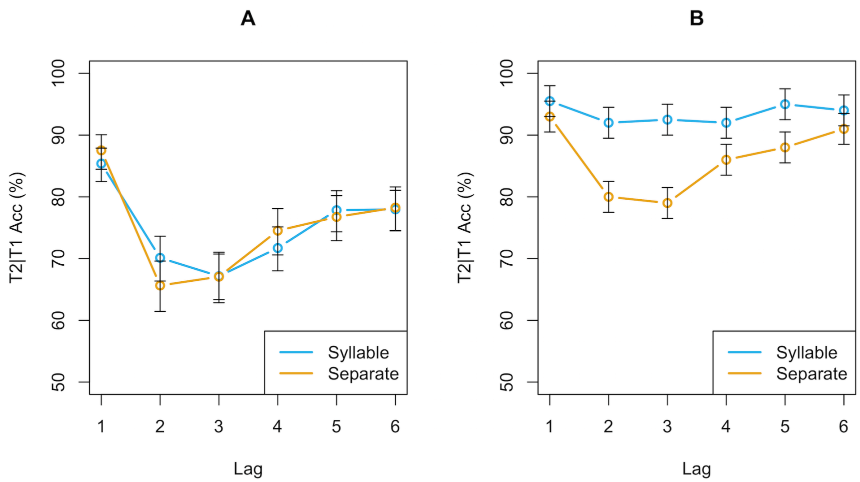

Figure 3. Comparison of replication and original study. (a) the mean T2|T1 accuracy per instruction condition over the six lags in our study. (b) the mean T2|T1 accuracy of the original study by Ferlazzo and colleagues (Ferlazzo et al., 2007). In contrast to the original, performance in the 'syllable' condition (blue line in 3a) did not significantly differ from the AB in the 'separate' condition (orange line in 3a) in our study. Additionally, the difference in general accuracy between our data and the 'separate' condition of the original study (orange line in 3b) implies a difference in difficulty level of the two studies.

2.3. Discussion

Ferlazzo and colleagues (Ferlazzo et al., 2007) reported the results of an AB experiment in which participants showed a strongly reduced AB when instructed to report the two targets as a syllable. We explained these results by assuming that these participants employed a different consolidation strategy than commonly employed in the standard AB task. Instead of consolidating both targets separately into memory, they may have consolidated both targets as one chunk and therefore managed to avoid the bottleneck of memory consolidation thought to underlie the AB.

In Experiment 1 we attempted to replicate these results. The design of Experiment 1 was highly similar to the design of the original study involving the same manipulation of instructing the participants to report the targets either as a syllable or as separate letters and used similar stimuli. Nevertheless, this experiment failed to replicate the results of the original study: there was no significant difference in AB magnitude between the two instruction conditions.

The main reason for the failed replication may be that we were unable to effectively manipulate the participants' consolidation strategy. This could perhaps have been caused by subtle differences in how the instructions were phrased. Additionally, the participants in our sample might have been less sensitive to the syllable instructions. Possibly, due to differences in linguistic and cultural background, the concept of a syllable may not have been as clear to our Dutch participants compared to the original Italian participants, and, therefore, may have been ineffective in spurring the chunked consolidation strategy (Ferlazzo, personal communication, June 26, 2019). Note also that the SOA was 20 ms longer than in the original study, which may have played a role in the effectiveness of inducing the chunked-consolidation strategy. In addition, the targets in the original study may have been substantially easier to distinguish from the distractors, which could have improved performance and facilitated in the chunking of the targets. This possibility is supported by the fact that the participants in the 'separate' condition of the original study outperformed the participants in both of our conditions. Note, however, that the reduced AB in the original study could not simply have been an artifact of the stimuli used in the original study. The original study also included a 'separate' condition which showed a mostly standard AB (as can be seen in Figure 3b).

Although the original results were not replicated, it seems unlikely that those results were merely a chance finding. There were small differences between the original study and the replication which might have prevented us from successfully manipulating the participants' consolidation strategy. Furthermore, the original paper reports two additional experiments that both supported the results of the experiment we attempted to replicate. To conclude, although the failed replication suggests that the manipulation of the original study does not reliably result in the same effect, one may also argue that it does not categorically reject the original results but rather shows how difficult it is to manipulate participant strategy. Therefore, we conducted Experiment 2 involving a stronger manipulation of participant strategy in order to accurately test the predictions of our AB model.

3. Experiment 2

The manipulation we included in the design of Experiment 1 seemed to be insufficient to cue participants to use the chunked consolidation strategy. An additional manipulation enforcing this strategy is thus necessary to create a suitable data set to test the predictions of our AB model. Therefore, Experiment 2 was set up, containing targets that were expected to promote chunking in addition to the instruction manipulation.

3.1. Method

3.1.1. Participants

In total, 82 participants (45 female, average age: 20.9) took part in the experiment. All participants were students of the University of Groningen who received a financial compensation of 8 euros for their participation. Two participants were removed from the final data analysis because too many trials had to be excluded (see ‘data preparation’ below for more details). The sample size was increased in response to the small effect of the instruction manipulation found in Experiment 1. Ethical approval was acquired from the Research Ethical Review Committee of the University of Groningen and participants signed an informed consent form before taking part in the study.

3.1.2. Stimuli

The same experimental setting and apparatus was used as in Experiment 1. On each trial a sequential stream of twenty stimuli was presented, including multiple distractor non-targets and one or two targets. The distractors were letters, randomly drawn with replacement, with the additional constraint that two sequential stimuli were never identical. The targets in this study were chosen in order to promote chunking. They consisted of corners of a square, as used in a study by Akyürek and colleagues

(Akyurek et al., 2012), shown in Figure 4a. Every corner of the full square was 7 pixels wide and both the horizontal and the vertical side were 23 pixels long with 8 pixels of white space in between two neighbouring corners. The white space in between the corners gave the impression that the corners were distinct but, because of the total configuration, still part of a bigger figure (e.g., forming a square). On most trials, two targets were presented consisting of one or two corners. The exact make up of these two targets was determined randomly with the single rule that there could be no overlapping corners (e.g., if T1 consisted of the bottom left and the bottom right corner, then T2 could not contain any of these corners), see Figure 4b and 4c for examples of targets. Both the targets and the distractors appeared in a similar size of around 2 degrees of visual angle on the screen. The font used for the distractors was 'droid sans mono'.

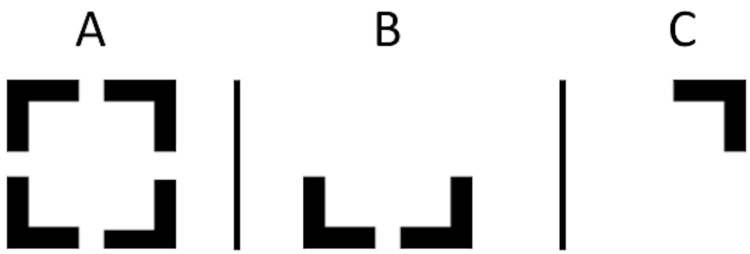

Figure 4. Examples of the stimuli used in the study. (a) the full square that was used as the base for the creation of the targets. (b) an example target of the maximum size (i.e., 2 corners). (c) an example target with only one corner. The two targets in one stream could consist of any combination of these targets as long as there were no overlapping corners.

3.1.3. Experimental design and procedure

The experiment was set up with a between-subjects manipulation of instructions. Participants in the 'separate' group were instructed to report both targets separately whereas participants in the 'combined' group were instructed to report the two targets as a single unit. Participants were not aware of the existence of the other condition. Additionally, the lag of T2 was varied. In this study lags 1, 2, 3, 4, 6, and 8 were included in the design. All

6 lags were presented equally often (50 trials per lag) and the order was determined randomly (without replacement).

Before the experiment started, verbal instructions were provided in addition to written instructions on the screen in a similar manner as in Experiment 1 to ensure adequate understanding of the task. Additionally, participants performed 24 practice trials with two targets (4 for each lag) and 4 trials in which only one target was presented to further ensure correct understanding of the task (28 in total).

The experiment consisted of 378 trials in total (including practice) and took around 45 minutes to complete. Every trial in the experiment consisted of an RSVP stream of 20 items with a presentation rate of 12 Hz (i.e., every item was on screen for 83.3 ms) preceded by a fixation cross which was on screen for 1 second. The presentation rate was slightly faster than in Experiment 1 and most AB tasks (frequently presented at 10 Hz) to make the task sufficiently challenging for the participants. On most of the experimental trials (300 out of the 350 experimental trials) two targets were presented. On the remaining 50 trials only one target was presented and the place of the T2 was taken up by an additional distractor.

Finally, at the end of every trial one or two response screens appeared (depending on the instruction condition). The response screen displayed a 4 by 4 grid with all response options (16 options in total). These response options consisted of all 15 possible targets and an option to indicate that there was no second target present or that the target in question was missed. The response option screen was identical in both conditions. Each response was given by pressing the appropriate key on the keyboard that was associated to it (a key was displayed underneath every response option). There was no spatial regularity between the location of the response options on the screen and the location of the keys on the keyboard. The keys with which the participants responded were the numerical keys at the top of the keyboard and the letters ‘Q’ up to ‘T’. The ‘Enter’ key was used to indicate that the participants had not seen the target. There was no time constraint on the responses.

Note that participants in the ‘separate’ group gave two responses while participants in the ‘combined’ group were only required to give a single response (Figure 5). This difference may have resulted in some of the participants in the ‘separate’ condition to also attempt to remember the order of the two targets. They were not instructed to report the two targets in the order in which they appeared and the feedback they received during the practice trials also did not depend on this order. However, the participants were also not explicitly instructed to report the targets in any order, so it is possible that some participants did attempt to remember the order of the targets. This might have made the task slightly more difficult for some of the participants in the 'separate' condition. However, we expected that this difference would only influence general accuracy and not AB magnitude. Furthermore, we took measures to limit the influence of this difference which are described in the next section. The crucial aspect of these measures is that the order was not relevant in either condition during data analysis.

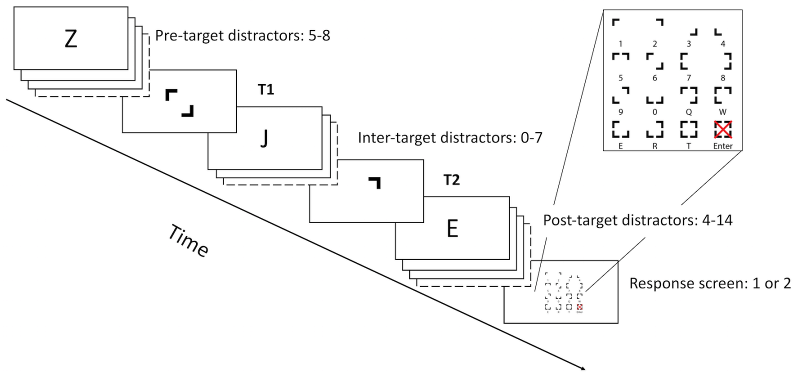

Figure 5. Schematic overview of a single trial with two targets. Every trial started with the presentation of a fixation cross for one second (not shown). The presentation rate of the stimuli was 12 Hz with no inter stimulus interval.

3.1.4. Data preparation

The stimuli used in the study and our hypothesis required a somewhat different way to collect the participants' responses than is common in AB tasks. The main difference between our study and most AB tasks is that the participants in the 'combined' instruction group only provided a single response. Because of this, we needed a method to be able to distinguish between T1 and T2 accuracy based on this single response. Additionally, in order to match the difficulty of responding correctly in both conditions of our study, the two responses given in the 'separate' condition had to be treated as a single response, ignoring the order of the responses, similarly to the 'combined' condition. To resolve these two issues, we transformed our data in the following way. Firstly, we combined the two responses given on every trial by the participants in the 'separate' condition into a single response. After this step, we treated the data from both conditions in the same way. Extracting the T1 and T2 accuracy from these single responses was subsequently done by comparing the presented T1 and T2 on a trial to the response given on that trial. If the response included all the corners belonging to one of the targets, this target was counted as correct. On the other hand, if the response did not include all the corners presented as one of the targets, this target was considered to be incorrect. Trials in which the participants reported more corners than were presented were excluded from the data analysis (7% of trials). Exclusion was chosen over simply judging the response incorrect because it cannot be retroactively determined whether the error should be attributed to T1 or to T2. Finally, as mentioned before two participants were completely excluded from the analysis because more than half of their trials were excluded after this step, leaving a total of 80 participants in the analysis.

3.2. Cognitive Model

3.2.1. Original Cognitive Model

A large portion of the model has already been created as part of a previous effort (Hoekstra et al., 2020). For clarity, we will first repeat and summarize the methodology of that work before describing the adjustments we made in order to adapt the model to the current task demands.

The initial cognitive model was constructed using the skill-based approach as outlined in the introduction. This meant that instead of creating a model specifically for the attentional blink it was assembled from general skills, mirroring how participants would approach a new task. Based on previous research and other models of the AB, we identified four basic skills required to successfully perform an AB task. These four basic skills were created by developing three models of other tasks that also made use of one or more of these skills. These three models were: (1) a visual search model, (2) a model of a complex working memory (CWM) task, and (3) a model of a simple working memory (SWM) task. Complete descriptions of the three basic models and more details on the modelling methodology can be found in the previously mentioned paper, where we also show that the models fit the appropriate experimental data for the memory tasks (Hoekstra et al., 2020).

After creating the three basic models, all the building blocks needed to create the final AB model were present and the AB model was assembled in the following way. The visual search model provided the first skill for the AB model: the "search" skill, used to distinguish targets from distractors in the AB stream. The simple working memory model and the complex working memory model provided the other skills. Firstly, the CWM model supplied the "consolidate-separate" skill, consolidating one individual item into memory. The reason for using this strategy is that in CWM experiments, the items that have to be memorized are interleaved with another task. This consolidation skill was used for the version of the AB model that modelled the standard consolidation strategy during AB tasks. The model of the SWM task provided the alternative consolidation strategy. This consolidation skill is capable of consolidating multiple items as a single chunk into memory, based on evidence in the literature that people do indeed use chunking strategies in such experiments (Anderson et al., 1998). This skill was used for the alternative AB model. The CWM and SWM models provided two additional skills, one responsible for retrieving the consolidated items from memory (the "retrieve" skill) and the other responsible for responding with the retrieved items (the "respond" skill). Thus, both working memory task models and both versions of the AB models used the same "retrieve" and "respond" skill.

3.2.2. Modified Cognitive Model

A modification to one of the skills described above had to be made in order to align the model with the demands of the task used for Experiment 2. This new task is largely the same as the previous task with the exception that the stimuli were different. Because of this, the visual search skill had to be instantiated differently.

Instantiation plays a crucial role in facilitating skill reuse because it allows for the task-specific information used by a skill to be adjusted without changing the procedural rules (i.e., the operators). In PRIMs models, skills often contain a certain number of variables which can be specified differently depending on the concrete task the model will be performing. For example, the same visual search skill is used for the model of Experiment 1 and

Experiment 2, but the variable "*distractor-type*" (specifying the nature of the distractors) is instantiated differently (among other variables). In the model of Experiment 1 this variable is instantiated as "*number*" and in the model of Experiment 2 this variable is instantiated as "*letter*". Because all task-specific symbolic knowledge is represented by such variables it becomes possible to reuse skills in different contexts without changing the skills themselves.

In addition, a variable can also be instantiated with another skill. The main advantage of instantiating variables with a skill is that for every skill it can be specified which skill to perform after it. This is crucial for skill reuse since it allows for the general skills to be carried out in any order (e.g., in some tasks it is required to immediately respond after retrieving the correct answer while in other tasks it might be required to perform some additional processing before responding, but this does not change the skill itself). We used this characteristic to adapt the four basic skills we created according to the demands of the AB task. For example, in the visual search skill a subskill determines how to get to the next stimulus. In a standard visual search task this is done by an eye movement to the next unattended stimulus, but in the AB task it is to wait for a new stimulus in the same location.

The changes to the visual search skill had one important consequence for the performance of the model. Instead of using a declarative memory retrieval as selection criterion to distinguish between letters (the targets) and numbers (the distractors), a perceptual judgement was used as criterion to distinguish between the corners of a square pictured in Figure 2 (the targets) and letters (the distractors). The main impact of this change is that the perceptual judgment is faster than a memory retrieval.

After modifying the visual search skill, all skills were put together in the AB model which carried out the task in the following way. If a letter (i.e., a distractor) is presented, the model ignores this stimulus and waits for the next. When a corner of a square (i.e., a target) is encountered, the model switches to one of the "consolidation" skills (depending on which version is run). If the "consolidate-separate" skill is performed, the model instantly consolidates the target into memory and no other operator can be executed for, on average, 200 milliseconds (the imaginal delay parameter in ACT-R and PRIMs), possibly leading to an attentional blink. The duration of this period varied between 50 and 350 ms, randomly determined using a uniform distribution. If the "consolidate-chunk" skill is performed, the model postpones consolidation of the target until the second target is encountered and keeps performing the task normally, such that no blink occurs in this version of the model. After all stimuli have been presented, the model uses the “retrieve” skill to retrieve the consolidated items from memory and, finally, uses the “respond” skill to respond with the retrieved items.

The underlying cause for the AB in our model is very similar to an AB model created in ACT-R (Taatgen et al., 2009) in the sense that the AB is caused by a wrong consolidation decision and not by fundamental information processing limits. The crucial aspect of the consolidation decision revolves around when the contents of the imaginal buffer (i.e., working memory) are committed to longer term storage. In our model this moment of encoding is directly linked to the consolidation skills. The ‘consolidate-separate’ skill encodes an item into longer term memory as soon as it encounters a target, while the ‘consolidate-chunked’ skill will only start consolidation when both targets have been detected. That the encoding moment can depend on the context is suggested by AB studies that use three sequential targets (Di Lollo, Kawahara, Ghorashi, & Enns, 2005; Olivers, van der Stigchel, & Hulleman, 2007). In these studies, the third target (presented at the ‘lag 2’ position in normal AB studies) is reported just as frequently as the first target, completely eliminating the classic AB effect. These results support the idea that the moment of memory consolidation varies depending on context or perhaps strategy. In this conception, the AB is caused by the fact that the ‘consolidate-separate’ skill is the default consolidation skill for most participants in the AB context and that it can be avoided when participants are compelled to use the ‘chunked-consolidation’ skill.

The predictions of the model (see Figure 6) are similar to the predictions by the original PRIMs model shown in Figure 1. These predictions were made by running both versions of the model 20 times with 500 trials per run for a total of 20000 trials (10000 per version).

3.3. Results

3.3.1. Effect of instruction

Our main hypothesis concerned the effect of instruction on performance in an AB task (Figure 7). We predicted that the participants in the ‘combined’ condition would show a smaller AB than the participants in the ‘separate’ condition. We chose to define AB magnitude by the slope with which T2|T1 accuracy improved over the lags. This method was recommended by MacLean and Arnell (MacLean & Arnell, 2012), especially for studies investigating modulations of the AB, and has been used before in a similar fashion (e.g., (Beanland & Pammer, 2012)). The main advantage of this method over taking difference scores (e.g., Lag 8 – Lag 2) is that it factors in all intermediate lags, not just the extreme lags. Furthermore, this method allows for quantification of the effect, since it not only provides information about whether the slopes are significantly different (i.e., the p-value), but also provides an indication of how different the slopes are (i.e., the effect size).

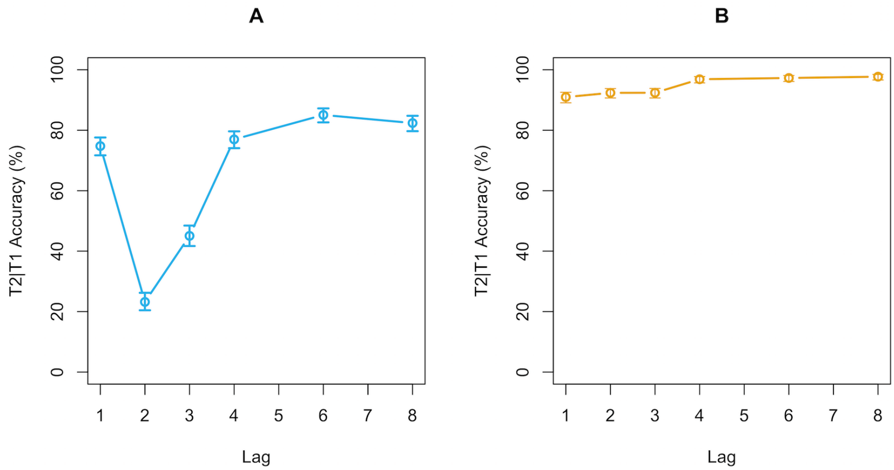

Figure 6. Model predictions for both version of the modified AB model. The predictions of the 'consolidate-separate' version of the model include a mostly standard AB with the deepest point at lag 2 (a). The predictions of the 'consolidate-chunked' version include no AB and a very high overall performance level (b). Confidence intervals were calculated using the Agresti-Coull method (Agresti & Coull, 1998).

To test our hypothesis, we fitted a logistic generalized linear mixed effects model on the T2|T1 accuracy data. This was done with the statistical software 'R' (R Core Team, 2015, 2017) using the package 'lme4' (Bates, Mächler, Bolker, & Walker, 2015). P-values were extracted with the package 'lmerTest' (Kuznetsova, Brockhoff, & Christensen, 2017) using the Satterthwaite method (Giesbrecht & Burns, 1985; Hrong-Tai Fai & Cornelius, 1996).

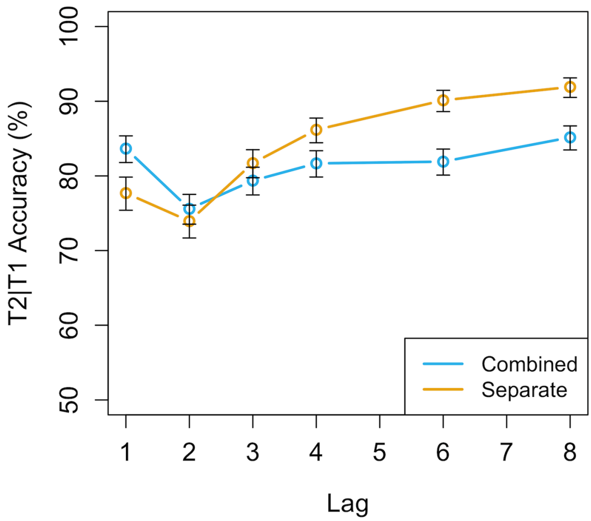

Figure 7. T2 accuracy as a function of instruction and lag. T2 accuracy in the 'combined' condition (blue line) has a significantly smaller slope over lag than T2 accuracy in the 'separate' condition (orange line). Confidence intervals were calculated using the Agresti-Coull method (Agresti & Coull, 1998).

Our final model, testing the effect of instruction on T2|T1 accuracy, included two fixed effects (Instruction and Lag) and a random intercept for subject. Instruction was a categorical factor with 'separate' as the reference level. Lag was a numerical factor starting at Lag 2. Lag-1 trials were not included in this model because of our definition of AB magnitude (see above). The analysis revealed no significant main effect of Instruction ($\beta = 0.0.18$, $SE = 0.35$, $z = 0.5$, $p = .6$) indicating that there was no difference in T2 performance between the two instruction conditions at the first level of Lag (i.e., Lag 2). The analysis did reveal a significant main effect of Lag ($\beta = 0.29$, $SE = 0.02$, $z = 15.9$, $p < .0001$) with the positive beta coefficient of 0.29 indicating that T2 performance improved on later lags. Finally, the analysis revealed a significant interaction between Lag and Instruction ($\beta = -0.14$, $SE = 0.02$, $z = -5.6$, $p < .0001$) implying that the slope of Lag is less steep in the 'combined' condition which indicates that the participants in the 'combined' condition showed a relatively smaller AB.

We, additionally, performed a post-hoc analysis of the unexpected difference between the two conditions at lags 4-8. In order to perform this analysis, we ran the same model but with Lag as a categorical factor instead of as a continuous variable. Entering Lag as a categorical factor allowed us to test for an interaction between Lag and Instruction for every level of Lag separately instead of testing for an interaction over all levels of Lag collectively. This analysis revealed a significant interaction between Instruction and Lag at lag 4 ($\beta = -0.34$, $SE = 0.14$, $z = -2.3$, $p = .02$), lag 6 ($\beta = -0.74$, $SE = 0.15$, $z = -4.9$, $p < .0001$), and lag 8 ($\beta = -0.67$, $SE = 0.16$, $z = -4.2$, $p < .0001$) indicating that performance was significantly lower in the 'combined' condition at these lags.

In addition to analysing the T2|T1 accuracy, we also conducted statistical tests on the effect of Instruction and Lag on T1 performance in a similar fashion. Our final model included Instruction and Lag both as categorical factors and a random intercept for subject. Average T1 performance differed slightly per condition: 94% in the 'separate' condition compared to 91% in the 'combined' condition. However, this difference did not reach significance at any of the lags (all $ps > .05$).

Finally, we analysed the performance on the trials where only one target was presented. Participants performed very well on these trials, correctly reporting the 'T1' in 96% of the trials. Performance on these trials was almost identical across the conditions with 97% accuracy in the 'separate' condition and 96% accuracy in the 'combined' condition.

3.2.2. Model predictions for the Effect of Instruction

The model predicted different performance patterns for the participants in the 'combined' condition and in the 'separate' condition. In particular, it predicted that the participants in the 'separate' condition would show a strong blink characterized by a serious decline in performance at Lag 2 and a quick recovery at Lag 3. In contrast, the model predicted that the participants in the 'combined' condition would not show an AB, performing close to ceiling at all lags. Although the model predictions were in the correct direction (there was a relatively smaller blink in the 'combined' condition), the difference between conditions was much smaller than anticipated by the model. Therefore, the model predictions regarding the effect of instructions were not fully supported by the data.

The model also predicted that the targets that were used in Experiment 2 (see Figure 4) would have an impact on performance. Because it was relatively easy to distinguish the targets from distractors in comparison to the ones typically used in regular AB designs, the task in Experiment 2 did not require a memory retrieval. The first step done by the model (i.e., detecting T1 from the stream of distractors) could therefore be accomplished faster, which subsequently caused the model to be faster in completing consolidation of the targets. This meant that the entire AB occurred slightly earlier, resulting in the atypically quick recovery in model performance at lag 3. This prediction was supported by the participant data: the AB was deepest at lag 2 and recovered quickly at lag 3.

Although the significant difference in AB magnitude between the two groups shows an influence of the instruction manipulation, its impact was not as large as predicted by our model. This weaker than predicted effect of the experimental manipulation may have been caused by an only partial success of the strategy manipulation. The unpredictable nature of strategy manipulations, as suggested by the results of Experiment 1, implies that the instruction manipulation may have had a different effect on individual participants. Some participants in the 'combined' condition may have still used the more natural separate consolidation strategy, furthermore, some participants in the 'separate' condition may have been prompted by the nature of the targets to adopt the chunked consolidation strategy.

Additionally, the data analysis revealed an unexpected difference between the instruction conditions at the later lags (lags 4-8). Before data collection, we expected that the 'separate' condition may be slightly more difficult because participants might attempt to also remember the order of the targets. However, the data analysis showed that this relationship was the other way around: rather than better, performance was worse in the 'combined' condition. This finding is remarkable because the 'combined' condition should always be less difficult than the 'separate' condition. After all, if it is possible to report the two targets separately, it should also be possible to report them as one combined unit.

In order to better understand these two issues, a closer look at the individual performance patterns of the participants in our study is required. To accomplish this, we performed a cluster analysis. This method allowed us to extract the most commonly occurring performance patterns in the data and to group similarly performing participants together based on the observed patterns of behaviour.

3.2.3. Cluster analysis

Cluster analysis is a collection of data driven statistical methods able to create groups of participants within a single data set. In our case, we wanted to group the participants in our experiment based on their T2|T1 accuracy for each lag. Many different clustering methods exist, differing in complexity and their applicability to certain data sets. We used the k-means clustering algorithm, which groups the data based on various sets of numbers (vectors) with which the squared distance from one of these sets is minimized for all sets that were observed (in our case, each observed set refers to a single participant). The cluster analysis was carried out with the R-package 'factoextra' (Kassambara, 2017).

Before a k-means cluster analysis can be performed, the k (i.e., the number of clusters resulting from the analysis) has to be determined. We accomplished this using a common approach actualized in the R-package 'NbClust' (Charrad, Ghazzali, Boiteau, & Niknafs, 2014). This method calculates 30 different measures calculating the optimal number of clusters, with the most optimal number of clusters being the number that received the most support from all 30 measures. For our data, 12 of these 30 measures suggested that 3 was the optimal number of clusters. The next most supported number of clusters was 2, with 5 measures supporting it. Therefore, we performed a k-means cluster analysis on the T2|T1 performance data for all lags included in our study with $k = 3$. This means that our cluster analysis will result in three groups of participants with different performance patterns.

The cluster analysis resulted in three differently sized groups of participants who showed distinctly different T2|T1 accuracy patterns (see Figure 8). The three clusters together explained 76.8% of the total variance in the data. The first and largest cluster consisted of 52 participants of which 28 came from the 'combined' condition and 24 from the 'separate' condition. It contained the participants who performed remarkably well in our study, performing at approximately 90% accuracy and showing no or little AB. The second cluster contained 16 participants of which 12 were in the 'separate' condition, showing a mostly classic AB pattern, but demonstrating little lag-1 sparing. The final and smallest cluster contained 12 participants of which 10 had been in the 'combined' condition and showed a performance pattern that was quite different from the previous two. Whereas these participants performed relatively well at Lag 1 with 70% accuracy, they performed quite poorly at the subsequent lags, hardly reaching 40% accuracy.

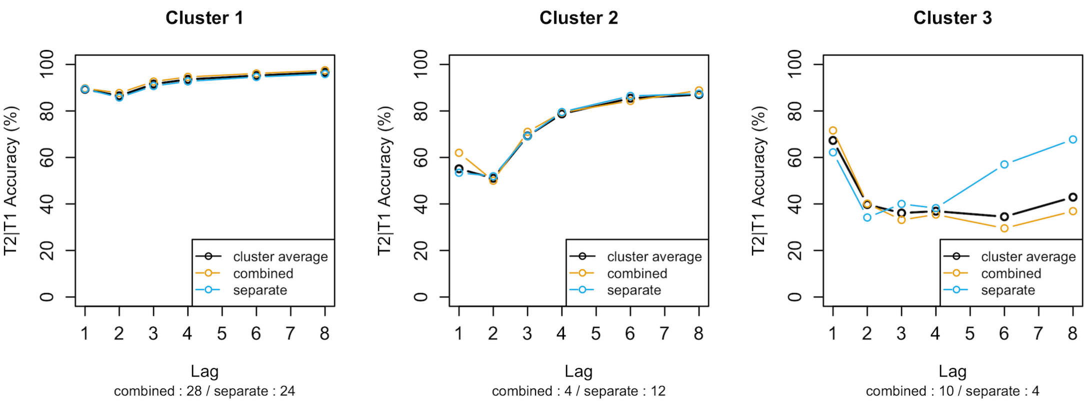

Figure 8. The three clusters present in the T2|T1 accuracy data. Indicated underneath the x-axis label are the number of participants included in this cluster per condition. Additionally, the dashed lines depict the cluster averages separately per instruction condition.

An interesting aspect of the results of the cluster analysis was that participants in the 'separate' condition seemed more likely to be assigned to Cluster 2 while participants in the 'combined' condition were more likely to be assigned to Cluster 3. Using a Pearson's chi-squared test on the distribution of participants from the two conditions assigned to either Cluster 2 or 3, we found a significant difference ($\chi2$ (1) = 7.1; p = .008), confirming that condition was an important factor in determining which cluster a given participant was assigned to. A comparable result was achieved when the distribution of participants from both conditions over all three clusters was tested ($\chi2$ (2) = 9.5; p = .009).

In addition to examining the differences between the instruction conditions across clusters we can inspect the differences between the conditions within a cluster. Figure 8 additionally shows the average performance of participants assigned to each cluster separately for both conditions. It reveals that the participants in both conditions performed very similarly to the cluster average in Cluster 1 and Cluster 2, indicating that the cluster analysis was successful in grouping comparable participants in those clusters. However, the participants in Cluster 3 differ in one crucial aspect.

The participants in the 'separate' condition show a performance pattern characterized by a deep and long AB including a return to normal performance levels at the later lags, while the participants in the 'combined' condition show consistently poor performance. This supports the idea that the instruction manipulation led participants to adopt different strategies. Specifically, it appears that the participants in the 'combined' condition rarely used the standard 'consolidate-separate' skill indicated by the very small number of participants showing a regular AB. However, it also suggests that quite some participants in the 'separate' condition used the 'consolidate-chunked' strategy (reflected in Cluster 1). Finally, it is important to repeat that the instructions did not include any specific directions on how the targets should be consolidated but merely concerned how the targets would be reported. Therefore, the participants were free to consolidate the two targets in any way they saw fit without breaking any of the instructions, which may have contributed to the unpredictability of strategy choice.

To conclude, the two previous analyses suggest that the instruction prevented participants in the 'combined' condition from adopting the standard 'consolidate-separate' skill, but that it did not prevent the participants in the 'separate' condition from spontaneously also using the same chunked consolidation skill. This again shows the unpredictable nature of strategy manipulations and supports the necessity of the cluster analysis to get an accurate understanding of the data.

3.2.4. Model Fit Cluster Analysis

The unreliable change in strategy following the instruction manipulation prevents direct comparison of the model predictions with participant performance since the model assumes that the manipulation will result in a perfect separation of strategies (i.e., that every participant dutifully uses the instructed strategy). Therefore, we compared the model prediction to the patterns revealed by the cluster analysis. An important side note is that we did not perform any model fitting after data collection, the presented model outcomes are purely predictions made before the experiment was conducted. The performance of the participants in Cluster 1 was compared with the version of the AB model that consolidated the two targets as one chunk and the performance of the participants in Cluster 2 was compared with the version of the AB model that consolidated both targets separately.

This strongly improved the fit of the model predictions with the participant data (Figure 9). The performance of the 'consolidate-chunked' version of the AB model lines up well with the performance of the participants in Cluster 1. Both the model and the participants performed very accurately with an accuracy level of around 90% combined with a very small AB (Figure 9a). Additionally, the 'consolidate-separate' version of the model shows the same pattern of performance as the participants in Cluster 2 (Figure 9b). Both the model and the participants show a clear AB with lowest performance at lag 2 and a quick recovery at lag 3.

Although the model predicted the correct direction of the effect, it did strongly overestimate the size of the effect (i.e., the AB magnitude). This might be due to the decreased T1 difficulty in Experiment 2 which might have a mediating effect on the AB (Visser, 2007). We chose not to account for this in our model because we cannot conclude that T1 difficulty is indeed the reason for the smaller AB and because incorporating a potential mechanism that might be able to explain it would diminish the generalizability of our AB model. A final interesting finding in Cluster 2 was that these participants did not show Lag-1 sparing. Although the cause of this cannot be conclusively determined, it may be due to the quicker target detection in our study relative to other AB studies. This faster target detection may have shifted the AB to the left, leading to worse performance at lag 1 but improved performance at lag 3.

Cluster 1 and Cluster 2 are in line with the model predictions, however our model does not account for the pattern shown by the participants in Cluster 3. The surprisingly low accuracy in this cluster was not predicted by the model and the data and design of the current study cannot give a definite answer as to what caused this performance pattern. However, it might be related to the combined consolidation strategy since it is displayed mainly by participants in the 'combined' condition.

3.4. Discussion

The design of Experiment 1 proved insufficient to change the consolidation strategy employed by the participants during an AB task. Therefore, we conducted Experiment 2 which included targets that promote chunking, in addition to the instruction manipulation already present in Experiment 1 (see Figure 4).

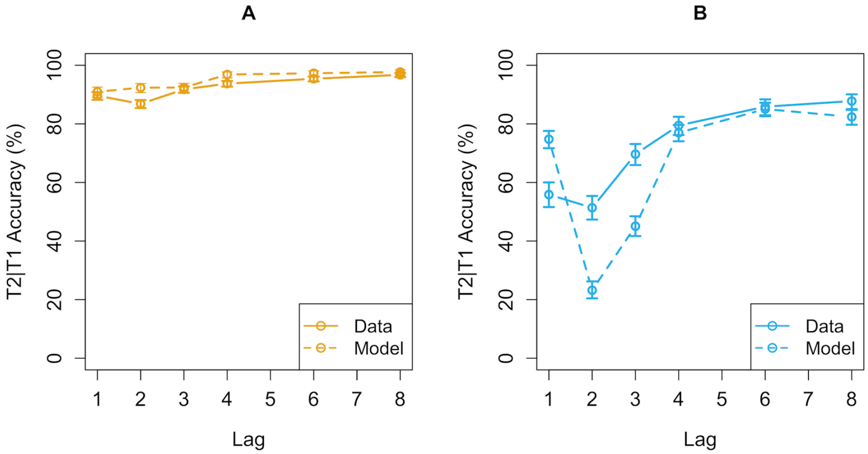

Figure 9. Model predictions and participant data for Cluster 1 and Cluster 2. (a) the predictions of the 'consolidate-chunked' version of the AB model (dashed orange line) which lines up well with the performance of the participants from Cluster 1 (solid orange line). (b) the predictions of the 'consolidate-separate' version of the model (dashed blue line) and the performance of the participants in Cluster 2 (solid blue line). Confidence intervals were calculated using the Agresti-Coull method (Agresti & Coull, 1998).

The results of Experiment 2 revealed a significantly reduced AB for the participants who were instructed to remember both targets as a single chunk (i.e., the 'combined' condition) compared to the participants who were instructed to remember both targets separately (i.e., the 'separate' condition). Although this result is in line with the predictions of the model, the modulation of the AB was much weaker than predicted by the AB model and, therefore, this initial analysis does not (fully) support the model predictions.

However, it is likely (as suggested by the results of Experiment 1) that the instruction manipulation has a different effect on individual participants. Therefore, we conducted a cluster analysis on the data in order to gain insight into these individual differences. The cluster analysis revealed three distinct performance patterns: (1) a large group of participants that performed remarkably well on all lags and hardly showed an AB, (2) a group of participants experiencing a mostly regular AB although without Lag-1 sparing and, finally, (3) a group of participants that performed relatively well at Lag 1 but performed very poorly at all subsequent lags.

The first pattern revealed by the cluster analysis is especially intriguing since it shows a very strong modulation of the AB experienced by a large number of participants (52). This opposes the common assumption that the AB reflects a fundamental limitation of cognition and raises the question of how these participants were able to avoid the AB bottleneck. Although it cannot be concluded with certainty, it is unlikely that this strong modulation is caused by the individual ability of the participants, because of the large number of participants in which it was observed. Additionally, it cannot be produced by the stimuli alone because the cluster analysis also revealed a group of participants showing a mostly regular AB. Therefore, it is likely that these participants managed to avoid the AB by using a particular strategy.

We assume that this strategy is the chunked consolidation strategy, because the use of the strategy seems to be directly linked to the targets used in Experiment 2, which were easily combined into a single chunk of information. This characteristic was initially intended to merely support the instruction manipulation, however after data collection it seems likely that the targets also cued the chunked consolidation strategy and that the instruction manipulation was not the only driving factor. Although our data cannot prove without question why the participants in cluster 1 managed to avoid the AB, their performance suggests they adopted the chunked consolidation strategy, which was triggered by a combination of the stimuli and instructions which allowed them to avoid the memory consolidation step thought to underlie the AB.

The performance pattern of the participants in the third cluster was unexpected and the design of the study does not provide a clear answer to why this occurred. However, significantly more participants from the 'combined' condition were assigned to this cluster. This suggests that these participants might have struggled with executing the 'consolidate-chunked' strategy. A crucial difference between the two conditions is that the two targets need to be integrated into one object in the 'combined' condition while this is not necessary in the 'separate' condition. It could be that this integration proved very difficult for these participants and that this led to the low performance at the later lags. The relatively good performance at Lag 1 in this cluster also suggests that the main difficulty with these participants occurs when there is at least one distractor in between the two targets. In accordance with this idea, the participants from the 'separate' condition in this cluster performed very differently. These participants showed a long and deep blink (as can be seen in Figure 8) which suggests no problems with integration but mainly points to a pronounced AB.

An important limitation of the current study is that the model predictions were only supported in a relatively 'post-hoc' manner. Although we consider the outcome of the cluster analysis to be a better reflection of the data than the partition into the two original conditions, we acknowledge that our initial hypothesis was not confirmed but that we only found support for our model in a different manner than we initially conceived. The post-hoc nature of the cluster analysis increases the uncertainty that the 'chunked' consolidation strategy is responsible for the absence of an AB in Cluster 1. This uncertainty is smaller for Cluster 2 since the 'consolidate-separate' strategy is the strategy that participants are commonly assumed to employ during standard AB tasks (Dux & Marois, 2009; Martens & Wyble, 2010). However, the 'consolidate-chunked' strategy is not commonly used to explain performance during AB tasks (although it has been suggested before (Taatgen et al., 2009)) and the data of the current study cannot unequivocally prove that this strategy was responsible for the absence of an AB in Cluster 1. An experiment using physiological measures might provide more conclusive answers, pupil dilation data combined with the 'deconvolution' method capable of entangling pupil responses to quickly occurring stimuli might be a good candidate for that (Wierda, van Rijn, Taatgen, & Martens, 2012; Willems, Damsma, Wierda, Taatgen, & Martens, 2015).

To conclude, our model does not provide a full explanation of the cognitive mechanisms involved in performing the modelled task. This is common in computational models since it is possible to check the model predictions against all aspects of the data and therefore exposes every aspect of the model that is not fully in line with reality. Following the generalizability ambitions of the skill-based approach we decided against adjusting model details after data collection (e.g., its parameters) because it often reduces generalizability to other tasks. Furthermore, refraining from post data collection adjustments also provides an interesting opportunity to learn from the incorrect model assumptions. One crucial aspect the model did not predict correctly was the large individual variation participants within an instruction condition showed. As previously mentioned, we assume this is due to differences in strategy although the data cannot fully exclude other possibilities or confirm that the strategies responsible are the ones we modelled. A second crucial aspect the model did not account for was that the combined instructions would have such a negative effect on so many participants. It seems that these participants struggled with integrating the two targets when there were one or more distractors in between the two targets. These two factors prevented us from analysing the data in the way we intended and the post hoc nature of the analysis we performed instead reduces the reliability of the conclusions. However, the strong reduction of the AB in such a large group of participants is surprising given the existing literature and the nature of the targets supports the idea that this might be due to a difference in how the targets were consolidated compared to more regular AB tasks.

4. General Discussion

In a previous paper (Hoekstra et al., 2020) a model of the AB was created only using skills (pieces of procedural knowledge) that had been created as part of other tasks. This resulted in a more naturalistic and human like model of the AB that succeeded in capturing commonly reported aspects of the data. Additionally, the model offered an explanation for the strong reduction (Arend et al., 2006; Olivers & Nieuwenhuis, 2005; Taatgen et al., 2009) and sometimes complete elimination (Choi et al., 2012; Ferlazzo et al., 2007) of the AB under certain experimental conditions. The model accounts for these findings by assuming that participants avoid the AB by using a different consolidation strategy (and skill). Instead of consolidating the two targets separately, they are consolidated together in a single chunk. This strategy allows participants to bypass the bottleneck of serial consolidation commonly suspected to be responsible for the AB.

In the current paper we attempted to collect empirical evidence supporting our AB model. We conducted a replication study of an experiment which reported that participants did not experience an AB when instructed to remember the two targets as a single syllable (Ferlazzo et al., 2007). Our replication study failed to show the same effect. This failed replication is valuable to report and shows the unpredictable nature of manipulations aimed at changing participants' strategy. Additionally, the difference in results of two very similar experiments (the original study and our replication) combined with the large individual differences shown in Experiment 2 indicate that the specifics of the design, for example the stimuli or the composition of the sample, may considerably change the outcome of an AB experiment. This strengthens the case for the use of large sample sizes in AB studies and shows the importance of including an investigation into the individual differences present in the sample (e.g., with cluster analysis).

Subsequently, we conducted a second experiment which included targets that facilitated chunking in order to obtain a suitable data set on which we could test our model predictions. The results of this experiment supported these predictions in that a significant AB reduction was found, but not to the extent predicted by the model. We explained this smaller than predicted effect by assuming that the instructions did not have the same effect on every participant. Participants in the 'combined' condition may still have employed the more common 'separate-consolidation' strategy while some participants in the 'separate' condition may have been prompted by the targets to use the 'chunked-consolidation' strategy.

Therefore, we performed a cluster analysis on the data to gain additional insight into these individual performance patterns. This analysis revealed the two performance patterns predicted by our model which are indicative of the two different consolidation strategies. In addition, it revealed a third performance pattern which might be the result of an inadequate execution of the chunked consolidation strategy. In conclusion, the direct manipulation of consolidation strategy only (the instruction manipulation) did not result in full support for our model predictions. However, the experiment included an additional indirect manipulation of consolidation strategy (the nature of the targets), the effect of which was uncovered by the cluster analysis. The combination of the direct and indirect manipulation show that it is possible to circumvent the AB bottleneck when a different strategy than the commonly presumed 'consolidate separate' strategy is adopted.

Most of the components of the AB model were created for a previous paper (Hoekstra et al., 2020) using a novel modelling approach which involves only (re)using skills taken from other models. This approach is based on the idea that people do not consider a new task in isolation but use previously learned procedural knowledge (skills) to perform the new task. The skill-based approach is one way of bringing that idea into practice and the data collected in Experiment 2 supports its predictions. The model we created improved on two of the core advantages of models created within a cognitive architecture: cognitive plausibility and generalizability. Our model has become more plausible than it would have been without considering skill reuse because the skills we used to model the AB have been verified by using them in other tasks besides the AB task. This reuse also shows that the procedural knowledge (the operators/production rules) used by our AB model is not implausibly (task-) specific. Furthermore, the use of general procedural knowledge has made the findings of our AB model more easily generalizable to other contexts. The fundamental limit to item consolidation can be expected in any situation in which people use the "consolidate-separate" skill (and be avoided by using the "consolidate-chunked" skill).

This requirement that the model can only be created from reusable skills encouraged a simplistic approach to building the model and resulted in a more natural and straightforward explanation of the AB. In our model, the AB is simply a consequence of normal cognitive functioning and does not require any specific mechanisms primarily aimed at explaining the AB. Additionally, our model offers a new perspective on the AB by showing the importance of strategy during an AB task. This effect of strategy may be the reason why it has been so challenging to arrive at a consensus on the mechanisms behind the AB. Previous models of the AB only propose a single mechanism responsible for the AB and therefore are unable to account for effects caused by differences in strategy. However, our data and model suggest that strategy plays a crucial role and that a singular explanation of the AB might not be sufficient.

Another advantage of the more simplistic modelling approach imposed by the limitations of the skill-based approach is the improved ability to relate the proposed mechanisms to other (general) theories and models. Our AB model is not as extensive and detailed as other AB models nor does it propose a completely original mechanism to be responsible for the AB. However, this mechanism is explicitly defined as part of a general processing step (memory consolidation), which strongly aids the integration of the insights gained from this model with existing theories. As mentioned before, there is no consensus on the mechanism responsible for the AB (e.g., see

(Olivers & Meeter, 2008; Wyble, Bowman, & Nieuwenstein, 2009) for alternative accounts), and we do not claim that our model resolves all the differences between the many models of the AB. However, using the skill-based approach, we were able to precisely identify a general cognitive process *potentially* responsible for the AB. This link to more general theories generally lacks in other AB models (which, on the other hand, are often more detailed than our model). However, this link is highly valuable. It strongly improves the generalizability of models and theories which can reduce the division between the different models of the AB and, in general, between the sub fields of cognitive psychology.

To summarize, the main benefit of using the skill-based approach is the improved balance between specificity and generalizability. Models created using the skill-based approach are specific enough to explain a certain phenomenon but, at the same time, are general enough to be easily linked to more general theories and other models.

The skill-based approach in its current form can nevertheless be improved in multiple ways. Firstly, creating models with this approach is more cumbersome and time consuming compared to standard modelling practices. The skills need to be built in a way that suits multiple tasks increasing the difficulty of creating these skills and they need to be verified by creating extra models that use the same skills. Secondly, applying the skill-based approach is complicated by certain assumptions made by many cognitive architectures. For example, the strict rule-based firing of production rules in ACT-R makes it very difficult to develop general production rules that can be used in multiple tasks. PRIMs allows more flexibility because operator selection is partly based on activation, however explicit condition checking is still required for reliable behaviour.

In our future work we will work on improving these aspects of the skill-based approach. However, the two main benefits of this approach can also be achieved by partial implementation of the principles of the skill-based approach. Models can become more flexible and human-like by considering that operators/production-rules are likely to be reused in other contexts when building a model. Additionally, the fact that our AB model could be easily related to other models and general theories is largely due to dividing the cognitive processes involved in the modelled task into general processing steps (i.e., skills).

In conclusion, building cognitive models based on the idea of skill-reuse can create novel insights and presents important improvements to certain aspects of cognitive modelling. In the case of the AB, it has shown that the AB can be a simple consequence of normal cognitive functioning and that it can be avoided using an alternative consolidation strategy. Because the AB model was created with general pieces of procedural knowledge (skills), the model reached a level of flexibility and robustness which is difficult to achieve without such an approach. Finally, we have shown that the skill-based approach is capable of producing valid models and new predictions.

4

Obstacles to the skill-based approach: why is skill reuse so hard for cognitive architectures?

Skill reuse is a commonly accepted aspect of human cognition but it has been difficult to translate to cognitive architectures. We developed the skill-based approach which enables modelers to create models composed of skills created for other tasks but it does not (yet) support fully reusable skills. We will discuss three factors that prevent full reusability: inflexible WM, rigid goal selection and all-or- nothing condition checking. The factors are discussed in the context of the architecture PRIMs but they also apply to many other cognitive architectures. Finally, we discuss possible solutions to alleviate these issues.

This chapter has previously been published as:

Hoekstra, C., Martens, S., & Taatgen, N. A. (2022a). Obstacles to the Skill-Based Approach: Why is skill reuse so difficult for cognitive architectures? In T. C. Stewart (Ed.), Proceedings of the 20th International Conference on Cognitive Modeling.

1. Introduction

Many tasks share considerable overlap in the cognitive elements required to complete it (Lee & Anderson, 2001). This cognitive overlap is one of the key fundamental principles underneath the attempt of cognitive architectures to arrive at a unified theory of cognition. In cognitive architectures the overlap in cognitive elements is put into practice by defining the blank slate cognitive system (i.e., the architecture) consisting of modules and buffers that underlies all behaviour (J. R. Anderson et al., 2004). This approach has led to successful modelling of a wide range of tasks and paradigms; however, a crucial additional consequence of the cognitive overlap between tasks has never received much attention. Not only can the same architecture be used to complete many tasks, this architecture can also very often be used in the same way (i.e., with the same procedural knowledge). Incorporating this into cognitive architectures would take into account the fact that huge proportions of our capabilities have been acquired through a long process of development and learning while currently only the innate aspects of cognition are considered. In order to bring this idea into practice, we have developed the skill-based approach to cognitive modelling.

This approach can be valuable for multiple reasons. Firstly, models will mirror human behaviour more closely which will improve model fit (Stearns & Laird, 2018). Secondly, reusing procedural knowledge is a large contributor to the flexibility people possess in executing various tasks. Incorporating it into cognitive modelling and AI could strongly improve flexibility and robustness (Taatgen et al., 2008). Finally, the large range of models created in the different fields of cognitive science can be integrated more easily if they all draw from one pool of basic building blocks.

1.1. PRIMs

We have explored the idea of skill reuse in the cognitive architecture PRIMs (Taatgen, 2013). We will give a short introduction to PRIMs here and in the relevant sections further down the paper. (See Taatgen (2013) for a complete introduction). PRIMs is based on ACT-R and inherits many of its properties. It is a cognitive architecture built up from distinct cognitive modules whose actions are controlled by “production-rules” (operators in PRIMs) and it contains a similarly functioning declarative memory system. An important distinction between the two architectures is that the operators in PRIMs are built up from smaller units than ACT-R's production rules. These smaller units are the **prim**itive information processing element**s** (PRIMs). PRIMs are considered the basic elements of cognition and are only capable of either moving or comparing pieces of information in the workspace. Although a single PRIM is not very powerful, combinations of PRIMs (i.e., operators) are able to execute complex cognition on the same level as ACT-R. These primitive operations are assumed to be universally applicable to any task and therefore can provide low-level mechanisms of transfer. They are also relatively easy to implement in neural architectures (Stocco, Lebiere, & Anderson, 2010). The central concept of the skill-based approach, a skill, is one level above an operator. A skill is a reusable collection of operators that perform a part of a task. Although a skill is larger than an operator, carrying out a skill still only takes a small amount of time in the order of one second or less.

The low-level transfer combined with the higher-level concept of a skill make PRIMs well-suited for exploring the skill-based approach although (most of) its principles can be implemented in other cognitive architectures as well.

1.2. The Skill-Based Approach

The central idea of the skill-based approach is to construct models of tasks in the same way humans would approach a new task. When people are confronted with a new task, they do not need to figure out from scratch how to complete this task but instead can rely on previously learned knowledge which has proven successful (Salvucci, 2013). A good example of this are the experimental tasks typical of cognitive psychology. Participants have usually never encountered these tasks before, yet they are quickly able to figure out what to do. Since they do not have time to learn new procedural knowledge specific to this task, it suggests that they reuse existing procedural knowledge. Concretely, the skill-based approach assumes that learning (almost) any new task merely means composing it from already existing skills.

A fundamental challenge to emulating this human-like flexible behaviour in cognitive models is balancing generalizability with accuracy. Different tasks come with different contexts and the model needs to be general enough to function in all these contexts but also specific enough to produce the same result regardless of that context. The common solution to this challenge is to allow for dynamic variable binding (Greff, van Steenkiste, & Schmidhuber, 2020); that is, allow variables to take on different values depending on the context. Although this solution is commonly adopted across different types of AI, there is no consensus on how it should be implemented (Feldman, 2013). The solution adopted by PRIMs is variable instantiation; a skill is created with general variable names which are only defined (instantiated) when the skill is used in a new context. However, there is no principled way in which this mechanism is implemented in the architecture.

More exact details can be found in our previous publications on the skill-based approach in which we propose the method (Hoekstra et al., 2020) and test the validity of its predictions (Hoekstra, Martens, & Taatgen, 2022b), but in short the skill-based approach works as follows. The first step of the skill-based approach is determining which basic skills are responsible for performing the modelled task based on previous literature. This step comes forth out of the fundamental principle of the skill-based approach that every task is a composition of basic processing steps that have been done (many times) before. For example, in the attentional blink (Martens & Wyble, 2010) model we have constructed (Hoekstra et al., 2020), the four basic skills we included were 'visual search', 'consolidation', 'retrieval', and 'response'. Skills that were reused from other models. This first step increases the generalizability of a model because the ubiquity of its basic building blocks allows it to be easily linked to other models and theories. The second step involves creating and testing the validity of the basic skills. In this step, other models which include (some of) the basic skills are built and these models are compared with human data. In our attentional blink model, we completed this step by creating a model of a simple visual discrimination task and two working memory tasks (a simple working memory task and a complex working memory task). This step is necessary to create the basic skills and it provides evidence for the accuracy of these skills. The final step involves adapting the basic skills to the context of the task of interest. In PRIMs, the cognitive architecture we used, this is done by instantiating the skills.

Following this method, we succeeded in constructing a model of the attentional blink (AB) that consisted of elements (skills) that worked in both the original task (e.g., the complex working memory task) as well as the AB task. This shows that it is possible to create cognitive models out of elements created for other tasks and that models can be created by merely assembling already existing procedural knowledge. However, the process of creating these skills was quite laborious and it often required making modifications to the basic skills that seemed too "AB-specific" to be part of general basic skills (Hoekstra et al., 2020). In short, we succeeded in creating a model with *reused* skills but not with fully *reusable* skills. That is, we managed to create an AB model out of skills that are also parts of other models (and are therefore *reused*) but these skills cannot be freely reused in every other task that includes the same basic skill (i.e., they are not fully *reusable*). However, this is crucial; making the step from *reused* skills to *reusable* skills would realize the full potential of the skill-based approach. It would standardize the knowledge used in cognitive models as well as increasing the ease with which skill-reuse can be implemented during model building.

1.3. Current paper

In the current paper, we will discuss which factors cause the difficulties in creating fully *reusable* skills. We will describe three open questions that complicate the implementation of the skill-based approach, specifically in PRIMs but some also apply to ACT-R. Although these open questions demonstrate practical problems in implementing the skill-based approach, they also point to fundamental unanswered questions about how flexibility should be balanced with cognitive plausibility as well as learnability. The questions will be illustrated by challenges we encountered while using the skill-based approach to model the updating tasks described by Miyake and colleagues (Miyake et al., 2000).

2. Inflexible Working Memory

In PRIMs and ACT-R the main purpose of working memory (WM) is to keep relevant information quickly available and to support the building of new chunks. WM in ACT-R does not consist of one dedicated system but instead consists of two modules that together function as WM: declarative memory and the problem state (Nijboer, Borst, van Rijn, & Taatgen, 2016). Declarative memory is responsible for storing chunks while the problem state takes care of keeping the chunks immediately available and is capable of creating new chunks.

In PRIMs, WM does consist of a single dedicated module responsible for keeping information readily available and for creating new (long-term) memory chunks. This module is called the imaginal buffer; however, it is often referred to as the WM-buffer and, for clarity, we will follow that convention. The WM-buffer in PRIMs works as any other buffer in the architecture in the sense that it has slots in which information can be placed and retrieved without any penalty. The slots function independently of one another and are numbered starting with one. Information is placed in and withdrawn from WM by a PRIM. For example, placing information presented on the screen in WM can be done by the PRIM *V1 -> WM1* and information can be taken out from WM by a PRIM such as *WM1 -> AC2*. Information can also be moved around within WM, for example *WM4 -> WM1*. The use of numbered slots in WM makes it much easier to reuse skills and operators compared to using named slots such as in ACT-R. However, it is not flexible enough to facilitate full reusability because the numbered slots are often still too rigid.

The inflexible working memory causes two main issues. The first is that the slots that will be used by the skills in the separate tasks need to be calibrated to work together. This requires a lot of effort from the modeler and although it is manageable for smaller and homogenous models, it quickly becomes unwieldy when the model involves many skills and different types of tasks. This is not a fundamental limitation, but it does present an obstacle to the adoption of the skill-based approach, especially when skill reuse is only a secondary interest. The second issue is more fundamental. Reusability of skills depends on the availability of the WM slots used in the original task. When these slots are not available in a different task, the skill cannot be reused. For example, the ‘read’ skill in our updating model stores the newly presented item in WM5 because the first four slots are used to keep track of the previously presented items. This might become problematic if the model would move on to a five-item memory task because the WM5 slot will be used to keep track of the fifth item. This illustrates that WM is not flexible enough unless a skill is designed while keeping every possible combination of tasks in mind and that full *reusability* is not yet possible.

Besides causing practical difficulties in using the skill-based approach, the issues with WM also point to a more fundamental question of how WM should be implemented in a cognitive architecture. The challenge is that WM needs to be extremely flexible on the one hand, but also consistent

with the limitations that have been identified in the literature on working memory.
The buffer-based design of PRIMs' WM has the advantage of being relatively flexible. It can be used in many types of tasks and it can store many types of information, additionally it provides a means of keeping information readily available. However, it lacks some plausibility because it assumes perfect (decay-free) storage of its contents which is not fully in line with the WM literature.

The alternative to using a buffer for WM is to store items in declarative memory. This is an attractive option, because it puts no hard limit on the number of items, but it still imposes a soft limit through memory decay. However, using declarative memory as WM also has a strong limitation in the sense that the information is not readily available, and has to be retrieved first. Given that items can only be retrieved one at a time, it is impossible to interrelate two or more items, which is a necessity for almost all tasks.

In conclusion, the practical issues we encountered while exploring the skill-based approach not only point to implementation issues but also to fundamental questions of how flexibility and plausibility should be balanced in WM.

3. Rigid Goal Selection

The goal module plays a central role in determining which production will fire in both ACT-R and PRIMs. Although the goal buffer plays a similar role in both architectures it does not work in the same way. In ACT-R the goal buffer influences production selection through the goal-state chunk present in the goal buffer and exerts its influence in a very explicit manner. Only production-rules which condition side matches the pattern in the goal-state chunk will be considered for selection. This way, the goal module is largely responsible for guiding the model towards firing the right productions at the right time.

The goal module in PRIMs has the same general role and also is responsible for the broad strokes 'supervision' of the model through a task. However, the goal module in PRIMs executes its role in a different and less explicit way. Operator selection in PRIMs is determined by the activity of the operators in memory. The most active operator gets selected first and its conditions are compared to the current context, if the conditions match the context the operator will fire. If the conditions do not match, the next most active operator will be retrieved and its conditions tested. This process repeats until an operator with matching conditions is found which will then fire. The goal buffer has a large influence on this process by spreading activation to operators that are associated with the current goal. This biases the selection process towards selecting operators that match the goal without guaranteeing that such operators will fire (noise or non-matching conditions can still prevent it). The subtle but forceful influence the PRIMs goal module exerts allows for organized behaviour while still allowing for flexibility within a task and, importantly, between tasks. The limitation related to the goal module is not how the goal module impacts operator selection but instead in how the goal itself is selected.

As is the case with all exchanges of information in PRIMs, goals are also determined by a PRIM. A new goal becomes active by a PRIM updating the value in G1 (the first slot of the goal-buffer). Although it is also possible to create situations in which multiple goals are active, for simplicity sake we will focus on a situation with one active goal. Goals are defined by symbols (similar to ACT-R) and therefore setting 'respond' as the goal can be done by the PRIM *respond -> G1*, if there is a skill with that same name. There are no rules about when or how the goal-determining PRIM needs to fire, however the architecture is designed in such a way that the most logical place for such a PRIM is in the final operator of a skill. This is very useful for simple models because it allows for an easy to understand (and flexible) way in which the model moves from one goal to the next. However, it becomes limiting in more complex models, especially in tasks in which the order of the goals is not always the same.

Determining the next skill within the previous skill essentially means that the next goal is decided by the previous goal. This severely limits full reusability of a skill because the role of a skill differs depending on the task. In some tasks, a certain skill might only be used at the end of a task (and therefore would not even require a *next-skill* operator) while in a different task the same skill might be a central part of the task and be used multiple times within a single trial. Switching skills gets further complicated by condition checking (which will be discussed in the next section) because different conditions might require the same skill to be performed next and, therefore, require separate operators. Often these limitations lead to a large array of different operators whose only function is switching to the next skill in different situations. For example, the ‘update-WM’ skill required four different operators only for switching between skills in the three tasks we modelled due to its centrality in those tasks. Extending the ‘update-WM’ skill to more tasks would only introduce more of such operators even though the basic procedural knowledge of updating WM would remain the same. This puts the cognitive plausibility of this way of switching skills into question, because it implies that every skill includes many operators that are only responsible for switching to the next skill.

This exposes two core limitations that are present in the current conception of PRIMs (and also ACT-R). Firstly, skills take care of two separate aspects of cognition: they perform the cognitive processing steps and are responsible for goal selection. That is, they are responsible for both selecting the goals and ensuring that they are achieved. This makes skill reuse difficult because, as our example shows, the basic procedural knowledge (which takes care of achieving a certain goal) might remain stable in most situations but the goal selection process might be different. Separating goal selection from goal execution will make skill reuse much easier. The second limitation is related to the type of information on which goal selection is based. Currently, goals are purely selected based on declarative knowledge. At the start of a task, by creating the goal-switching operators a ‘plan’ for the task is laid out and the model is practically incapable of deviating from this path. This way of goal selection is too rigid and overlooks the fact that people select goals based on a plan combined with their perception of the current situation (Altmann & Trafton, 2002).

Our modelling suggests that goal selection should be separated from execution and be made more flexible. However, this is not an easy task. The basic assumptions of PRIMs do not consider goal selection a special case of cognition and posit that it should be accomplished by a PRIM. Furthermore, increasing the flexibility of goal selection leads to questions of how this flexibility can be balanced with reliability since a more flexible model will also be more unpredictable.

4. Condition checking

The final factor limiting the creation of fully reusable skills we will discuss here is related to a fundamental aspect of both ACT-R and PRIMs, namely condition checking. In both architectures, productions consist of a condition side (left-hand side) and an action side (right-hand side). The conditions are compared to the content of the buffers before the action side is executed. In ACT-R, the conditions of all productions are evaluated in parallel and when multiple productions match the current contents of the buffer the production with the highest utility factor will be chosen. In PRIMs, condition checking occurs serially starting with the first condition of the most active operator. When one of the conditions does not match, the next active operator will be tested until a matching operator is found. This takes a certain amount of time at first, but after a while most conditions will be compiled into one execution cycle and the most active matching operator will usually be picked without any time cost (comparable to ACT-R).

Conditions are thought to be a fundamental part of procedural knowledge in both architectures. Therefore, full skill reusability means that both the action as well as the condition side need to be reused. Although the action side usually works in both tasks, the condition side is more problematic. After all, a different task usually means a different context to which the conditions will be matched. This often means that the condition side of an operator needs to be adapted to the new task which hinders reusability. Conditions that are especially challenging are those that are related to specific situations in a certain task. For example, in one of the updating models WM needed to be updated based on information in the visual buffer while in a different model it had to be updated based on information in WM itself. In this situation the action PRIMs (the right-hand side) were identical, but a different operator still needed to be created to accommodate the difference in conditions.

This leads to the question to which extent conditions are reused. The quick learning displayed by humans suggests that some previously learned condition-action associations are retained when a new task is performed, however our modelling implies that this does not apply to all of them. Take for example the operator depicted below.

```
operator respond-value-WM1 {
      V1 = *report-instructions
      WM1 <> nil
      ==>
      *action -> AC1
      WM1 -> AC2
      }
```

This operator gives the response (stored in WM1) at the end of a trial by performing an action (e.g., pressing a key on a keyboard). In this case, the second condition can be retained without problem because reporting WM1 would always require WM1 to not be empty. However, the other condition which tests whether the report instructions are currently on-screen should probably not be retained because it depends on the task.

The example suggests that not all conditions are created equal and that some conditions should not be reused. Especially conditions aimed at representing a task-specific situation hinder skill reuse suggesting that conditions might not be the best way to represent task-specific context.

5. Potential solutions

The three limitations we discussed impede the practical usefulness of the skill-based approach but we believe that they will not present a fundamental roadblock to fully *reusable* skills. The limitations we discussed are largely consequences of the reliance of cognitive models on the input of task-specific details from the modeler. Therefore, these issues might be alleviated by implementing learning mechanisms with which the model can figure out task-specific details independently or by providing more principled ways in which the modeler can specify such details.

The first limitation we discussed involved WM. The key issue here is that the inflexible WM demands a lot of coordination from the modeler because the model is not aware of the identity of the WM contents. A possible way to alleviate this would be to store the to-be remembered value together with its meaning (e.g., store the value "four" together with "current-stimulus"). This cannot be done in the current conception of the WM; however, the DM module does possess the required properties. By storing chunks in the DM (such as depicted below) the model would be aware of the value as well as the identity.

```
ISA fact
SLOT1 binding-fact
SLOT2 current-stimulus
SLOT3 four
```

In this situation, the current PRIMs imaginal buffer (i.e., the WM buffer) would be used almost exclusively to facilitate the creation of new chunks and as a problem-state (Borst, Taatgen, & van Rijn, 2010). Importantly, in order to keep the high flexibility of a buffer-based WM, these chunks should be accessible without the need of an explicit retrieval request but instead through means of a PRIM. For example, by allowing a PRIM to directly create bindings (e.g., *four -> *current-stimulus*).

This way of organizing WM provides a better balance of flexibility and plausibility, because chunks are subject to decay and retrieval times, however the information in WM is still easily accessible because it can be directly done by a PRIM. Furthermore, this design of the short-term memory would also provide a mechanism for the variable binding problem discussed earlier in the introduction. The dynamic bindings required to facilitate flexible model behaviour could be stored in this same manner. Ideally, the model would create these flexible binding chunks independently (e.g., when 'reading' the instructions) which would tremendously improve skill reusability as well as model autonomy.

The second limitation we discussed involved the manner in which the next skill is selected in PRIMs. This issue boils down to how the next goal is placed in the goal buffer. In the current situation, the previous skill usually places the next skill in the goal buffer but this method creates a large amount of procedural knowledge only aimed at switching between skills.

There is a possible solution that fits the PRIMs philosophy. Instead of having one active skill, two skills can be active: one skill for execution, and one skill for planning. The execution skill carries out the actions required to achieve a particular subtask, and then terminates itself. The planning skill is then responsible for selecting a next skill. This would be a big improvement over the current situation because it allows for goal switching separate from goal execution based on both a pre-made plan as well as the current context. Additionally, it allows for a flexible representation of task-specific information without the need to include such information in the general skills.

The final limitation we discussed concerned condition-checking. The limitation to skill reusability associated with condition-checking is that every task has a different context which makes it likely that the original conditions will not apply. Additionally, our modelling showed an important distinction between generally applicable conditions and task-specific conditions and raised the question whether conditions are the best way to represent task-specific context.

Testing conditions is one way to establish a mapping between the current state of the cognitive system and the action to be taken, but not the only one. Neural network approaches to modelling operators often use inspiration from the basal ganglia. The basal ganglia are considered to be central to forming context-action mappings and recent modelling efforts have created models capable of creating such mappings. These mappings provided reusable context-action associations while retaining flexibility by means of small changes to the connection weights in the network (Stewart, Bekolay, & Eliasmith, 2012; Taatgen, 2020). Such functionality could be incorporated in production-based architectures by specifying (or learning) connections between certain items in the workspace and operators. For example, the first condition of the previously mentioned example could be replaced by specifying a positive connection (through spreading activation) between the report-instructions and this operator. This would make it more likely that it gets picked when such instructions are on the screen but it does not prevent the operator from firing when they are absent. This functionality is already possible in PRIMs but it might be helpful to explicitly make it part of an operator definition (in addition to conditions) which is not only practical but also highlights that these connections are reused.

6. Conclusion

The skill-based approach is a promising addition to the arsenal of a cognitive modeler; however, the previous discussion has shown that there are still some important limitations. The inflexible WM demands a lot of coordination from this modeler, the unnatural goal selection requires a large amount of inefficient procedural knowledge and the all-or-nothing condition checking severely hampers the versatility of operators. Resolving these issues will require some substantial modifications to the cognitive architecture we employed and to production-system architectures in general. We proposed some solutions in this paper which we will explore in a subsequent study.

The current paper resulted from attempting to apply the skill-based approach to a series of basic tasks that make use of skills that are widely used. The difficulties we experienced show that current cognitive architectures do not support the creation of fully *reusable* skills. This does not mean, however, that the skill-based approach is completely ineffective, current architectures do support the use of *reused* skills and capitalizing on this characteristic will already result in more valid and generalizable models.

5

A skill-based model of executive function

The skill-based approach is a promising modeling approach which allows modelers to create models based on the idea of skill reuse. However, as we discussed in the previous chapter there are three limitations that prevent full adoption of the ideas of the approach and the creation of fully reusable skills. In this chapter we propose solutions to all three of these limitations. These solutions will be tested by creating models of the executive function tasks described in Miyake et al. (2000). The solutions made it much easier to create models using the skill-based approach and, additionally, the resulting model of executive function (consisting of eight smaller models) provided interesting insights into the nature of the EFs as described in Miyake et al. (2000). Our model suggested a fundamental cognitive strategy for each of the three EFs. Additionally, it showed that the individual differences present in executive functioning might be caused by differences in procedural knowledge instead of differences in the (more) stable architectural mechanisms that this procedural knowledge acts upon.

1. Introduction

The skill-based approach is a novel modelling approach with great potential to improve generalization and integration across models made for different tasks and in different fields. It accomplishes this by allowing modelers to mirror the way in which humans learn to accomplish a new task during model creation. The fundamental aspect of human behaviour that the skill-based approach attempts to integrate into modelling practice is skill-reuse. People can learn new tasks very quickly, which suggests that they reuse already existing procedural knowledge and apply it to the new situation (Hoekstra, Martens, & Taatgen, 2020). In this view, learning a new task does not require (a lot of) learning of new procedural knowledge but merely consists of assembling the correct blocks of knowledge that are already available. In the previous chapter we discussed three limitations we encountered in PRIMs while applying the skill-based approach to the executive function tasks described in Miyake et al. (2000) and proposed solutions to these limitations. In this chapter, we will describe how we implemented these solutions in PRIMs and we will test whether they have the intended effect by creating a model of the tasks described in Miyake et al. (2000) using the skill-based approach. Additionally, the model that would result from this effort could shed light on certain unknown aspects of executive function.

All our efforts using the skill-based approach have been based on the cognitive architecture PRIMs (Taatgen, 2013). Although PRIMs was developed with skill-reuse in mind, there were still certain limitations present in the architecture and in how we used the architecture to create reusable skills. The previous chapter contains a more detailed discussion of the limitations and the situations in which they become problematic. Here we will shortly repeat these limitations and subsequently discuss the worked-out solutions and how they were implemented in PRIMs. We will first shortly discuss the PRIMs syntax in order to facilitate understanding the examples containing PRIMs model code.

1.1. PRIMs syntax

The central elements of any PRIMs model are the operators. Operators consist of a left-hand side with conditions and a right-hand side specifying the actions taken when the conditions are true. The conditions can be specified using two symbols which are also often used in other programming languages. A test for equality is specified with a single equals sign (=) and a test for inequality is specified by a less than and a larger than sign (<>). The border between condition and action PRIMs is indicated by two equal signs and a larger than sign (==>). An action PRIM is indicated with a minus sign and a larger than sign resembling an arrow (->), this arrow represents the moving of information from one buffer-slot to another. For example, the PRIM `V1 -> WM1` represents the copying of information from the first slot of the visual buffer to the first slot of the working memory buffer. Finally, PRIMs allows for the use of symbolic constants. These are indicated by plain text identifying the constant in question. For example, the concept of 'one' is represented by the constant `one`. However, when a constant is preceded by an asterisk (*), it represents a variable. This means that the meaning of this 'constant' depends on a binding and can therefore be different in different models or even change at different times within a single model. Depicted below in Table 1 depicts an example operator which includes all six elements of the basic PRIMs syntax.

Table 1. Example operator adapted to include all six basic syntax elements.

PRIMs code	Meaning
`operator left-hand`	Name of the operator
`V3 = *target`	Check whether V3 is the value associated with *target*
`WM1 <> left`	Check whether WM1 does not contain left
`==>`	Indication that the following lines will be the action PRIMs
`V1 -> WM1`	Copy the current content of V1 to WM1
`nil -> WM2`	Place *nil* in WM2, meaning that it will be emptied
`Respond -> G1`	Copy the constant *respond* to G1

1.2. Overcoming the limitations to the skill-based approach

1.2.1. Working Memory

PRIMs' WM buffer is relatively flexible in the sense that it can deal with any type of information that can be defined in the workspace, however it has limited flexibility when it comes to reusing the procedural knowledge (i.e., the operators) required to interact with this system. Like all other buffers, the WM buffer has numbered slots. It works similar to all the other buffers in the architecture because information can be moved in and out of the slots by a PRIM. For example, information can be moved into WM with a PRIM such as *V2 -> WM1* and out of WM with a PRIM such as *WM2 -> AC2*. The limitation to reuse introduced by this design is that the numbered slots imply that every slot in WM is a different location in WM. Although this might be true on a lower level of analysis (two different items are not stored in the exact same 'location' in WM, whether that is a different group of neurons or in a different firing pattern), on the higher level of analysis on which PRIMs usually operates it is not helpful and severely limits skill reuse. For most models it is only necessary to indicate that something should be held in WM, the exact location is of secondary importance. Because location cannot function as the indication for identity, it is also required to store the identity of an item alongside the value. Finally, these system characteristics should be achieved in a cognitively plausible way.

Aspects of working memory in PRIMs, and also in ACT-R, have always been modelled either by using WM buffer (or imaginal buffer in ACT-R), or by relying on declarative memory. For example, the working memory strategies discussed earlier in this thesis in the context of attentional blink all rely on temporary storage in declarative memory. The modification to PRIMs' working memory system we proposed was to add the functionality of short-term bindings, which is an extension of the idea to use declarative memory as part of working memory. A binding is a short-term connection between a symbol in the architecture (e.g., *current-count*) and a value (e.g., *four*). This binding is represented by a chunk in declarative memory (DM) such as depicted below.

```
ISA binding
SLOT1 current-goal-chunk523
SLOT2 current-count
SLOT3 four
```

The binding makes a connection between an identity and a value because they are stored in the same chunk. Additionally, this is done in a cognitively plausible way because PRIMs' DM is based on the DM in ACT-R which in turn is based on a long line of memory research (Anderson, 1983, 2007; Anderson, Bothell, Lebiere, & Matessa, 1998). Although a binding is represented by a chunk and behaves the same in DM, it cannot be interacted with in the exact same manner. It is possible to search for the value of an identity (e.g., what value is associated with *current-count*?), but not the other way around (e.g., which identities are associated with the value three?). Instead bindings are created, updated and retrieved by means of a PRIM. Bindings are created by assigning a value to a symbol in the same way as information is moved into the WM buffer (*V1 -> *current-count)* and retrieved in the same way as information is moved out of the WM buffer (**current-count -> AC2)*. Updating an existing binding is not fundamentally different from creating a new one, it is accomplished with the same PRIM and it results in the creation of a new chunk without the previous chunk being deleted. Bindings are a compromise between explicit retrievals (which always requires two operators, and can only retrieve one item at a time) and storing intermediate results in buffers, which is immediate but inflexible. Retrieval of bindings is part of execution of an operator, but still has the properties of a regular retrieval in the sense that it costs time, and may fail. In addition, creating a binding incurs the costs normally associated with creating a chunk in the working memory buffer (200 ms).

1.2.2. Skill selection

The second limitation is the way in which control is handed over from one skill to another. Similar to the situation with WM, skill selection is not assumed to be an exceptional cognitive process and is therefore assumed to be accomplished with a PRIM as any other cognitive process. For example, selecting the skill to consolidate an item is done with the PRIM *consolidate -> G1*. There is no pre-specified way in which skill selection occurs, although the design of the architecture makes it likely that the next skill is selected as the final step of accomplishing the previous skill. This system is quite flexible because changing which skill should be set as the next skill can be changed without fundamentally changing the operator through instantiation (Taatgen, 2013). However, when tested to the limit it does become a limitation because a skill does not always have a clear-cut 'end-goal' (which decides the moment in which the next skill is placed in G1), and it can have multiple next skills within a single task and especially between multiple tasks (this leads to a large number of operators only aimed at switching to the next skill) or there perhaps might not even be a next skill.

The solution we proposed to alleviate the issues associated with skill selection does not entail an entirely new mechanism but rather a new method how the next skill should be selected. We proposed that skill selection should be accomplished by a separate skill which is executed in between two skills. Instead of selecting the next skill, the previous skill places the 'select-next-skill' skill in G1. This skill then, based on the current context, determines which skill should be accomplished next by placing that skill in the goal buffer. This skill is created specifically for one task and consists of several operators that place the next skill in G. Depicted below is an example of such a select-next-skill skill. This skill is taken from the category switch task. The operators in this skill are fairly straightforward, all they do is place the next skill in G1 depending on the current context. The first operator places the *shift* skill in G1 (i.e., it selects the *shift* skill as the next skill) when the current context matches the context that requires the model to switch to the *shift* skill. The second and third operator are similar, however they accomplish a switch to the *retrieve-characteristics* skill and the *select-next-skill-category-switch* (which does nothing, resulting in the model waiting for the next stimulus) respectively. Because the operators are always made specifically for every task, it is not necessary to use bindings.

```
define skill select-next-skill-category-switch {

      operator go-to-shift {
      V1 <> G2
      V2 = nil
      ==>
      shift -> G1
      }

      operator go-to-retrieve {
      V1 <> nil
      V2 <> nil
      RT1 = nil
      ==>
      retrieve-characteristics -> G1
      }

      operator wait-for-next {
      V1 = G2
      V2 = nil
      ==>
      select-next-skill-category-switch -> G1
      }

}
```

This is not a fundamentally different approach to the current goal selection mechanism because it is still done by a PRIM. However, the context in which a goal should be selected becomes much more flexible (because it only competes with other goals instead of all operators in a skill and can be set specifically for every task). Furthermore, the goal selection operators do not compete directly with the ‘action’ operators of a skill which makes model behaviour much more straightforward, especially in multi-task situations. Finally, this approach can capture the early learning effects when people start performing a simple new task. The operators in this skill will gradually undergo production compilation representing the increased performance speed when the precise steps for accomplishing a task become clear.

The new mechanism for goal selection allows skills to be picked purely based on the contents of the buffers (i.e., the state of the global workspace) by means of the creation of new operators. PRIMs currently does not include a mechanism with which these operators can be created by the model itself. However, since operators are merely chunks in declarative memory, this mechanism should be well within its capabilities.

1.2.3. Condition checking

The final limitation to PRIMs concerns condition checking. Condition checking is the manner in which a PRIMs model determines which operator to select at what moment. Every operator consists of a condition-side and an action-side. The condition-side contains the context in which an operator should be executed represented as conditions (statements about states of buffers which can be true or false). The action-side consists of the PRIMs which should be executed when these conditions are met. The operator selection process in PRIMs occurs in the following way. First, the most active operator is retrieved and its conditions are tested. If any of the conditions are false, this operator is rejected and the next-most active operator is retrieved and its conditions tested. This process is repeated until an operator is found with only true conditions. Although conditions can contain instantiable variables which adds some level of flexibility, they are still not flexible enough to work in a multi-task model. In order for a model to work as intended, conditions often need to be specified quite precisely. The conditions required for an operator to work in one task usually do not translate fully to a different task which means that an operator cannot be reused.

In order to alleviate the problems associated with this issue we proposed that the use of conditions should be minimal and that it should be combined with an additional mechanism. This additional mechanism is spreading activation. Spreading activation is a large aspect of PRIMs and ACT-R and it plays an important role in the memory systems of both architectures. In PRIMs, operators are simply seen as chunks in declarative memory and are therefore also affected by spreading activation. Additionally, PRIMs and ACT-R assume that chunks also receive activation from all other modules in the workspace (e.g., when the number three is on the screen, this might spread activation to a chunk containing the number three). These two assumptions combined make it possible that certain connections between operators and the context could be learned and that they might play a large role in operator selection and could replace some conditions. Take for example the operator below.

```
operator report-value-WM1 {
      V1 = report
      WM1 <> nil
      ==>
      press -> AC1
      WM1 -> AC2
}
```

This operator gives the response (stored in WM1) at the end of a trial by performing an action (e.g., pressing a key on a keyboard). In this example, the second condition will always be useful for this operator because WM1 should not be empty when the model reports what is stored here. However, the first condition is task-specific, it requires V1 (the first slot of the visual buffer) to contain *report*. Although giving a response will often be accompanied by a request to give this response, it does not necessarily need to be the case. Therefore, a better way of defining this operator would be to remove the first condition and replacing the second condition with a positive association between this operator and the presence of *report* in V1. In our proposed method that can be done in the following way.

```
set-sji("report-value-WM1","input","slot1","report",1)
```

This function creates a positive association between the operator called 'report-value-WM1' and the chunk 'report' when it is present in slot1 of the input buffer (which is V1) with a value of 1. This means that whenever 'report' is in V1 that this operator will get an additional activation of 1, making it more likely to be retrieved.

This approach to operator selection allows for much improved flexibility in model behaviour. One improvement is that allows for disjunctions: 'report' in V1 can activate the operator, but also 'write' in V1 or any other buffer. More complex combinations are also possible, for example that two out of three conditions need to be satisfied. Minimizing the use of conditions is very valuable from a skill-reuse standpoint since it greatly reduces the chance that a condition prevents an operator from matching a new context. Additionally, allowing for spreading activation between the buffers in the workspace and operators leads to controlled but highly flexible model behaviour. Furthermore, these associations are very simple and could result from a simple reinforcement learning process. Such a process is a good candidate for how the Basal Ganglia learns and determines action selection (Breiter, Aharon, Kahneman, Dale, & Shizgal, 2001; Cisek & Kalaska, 2010; Redgrave, Prescott, & Gurney, 1999) and has successfully been used in recent modelling efforts to create context-action mappings (Stewart, Bekolay, & Eliasmith, 2012; Taatgen, 2020). PRIMs also possesses a mechanism capable of learning such connections, although for the current modelling project we set these by hand.

1.3. Executive functions

In order to test the modifications we proposed, we set as a goal to model the nine basic tasks from Miyake et al. (2000) using the skill-based approach. In addition to providing a solid test case for the skill-based approach, the final models can also shed light on two aspects that are not yet fully clear in the executive functions literature: (1) what are the active mechanisms of executive function and (2) are executive functions learned procedural knowledge or do they rely on some level of innate ability?

Executive functions (EF), also referred to as executive or cognitive control, refer to a group of processes thought to underlie conscious and active cognitive performance: tasks in which you have to actively take control and cannot rely on a 'automatic' pilot. For example, solving complex equations, playing chess, or writing a concise introduction to a complex and broad topic. Performing a task that relies on executive control is often effortful (van der Wel & van Steenbergen, 2018) and high performance is difficult to sustain for a long time (Thomson, Besner, & Smilek, 2015). People who excel at applying EFs often also excel in other aspects of their life: academic success (Borella, Carretti, & Pelegrina, 2010; Gathercole, Pickering, Knight, & Stegmann, 2004), job performance (Bailey, 2007) and even physical health (Miller, Barnes, & Beaver, 2011; Riggs, Spruijt-Metz, Sakuma, Chou, & Pentz, 2010) are all related to high executive function. In contrast, poor executive function has been often shown to play a big role in psychological disorders such as addiction (Baler & Volkow, 2006), depression (Tavares et al., 2007) and schizophrenia (Barch, 2005). Over the years, a certain level of consensus has started to arise that there are three core EFs (Huizinga, Dolan, & Van der Molen, 2006; Lehto, Juujärvi, Kooistra, & Pulkkinen, 2003). These are (1) the ability to switch between tasks and mental sets, (2) the ability to monitor and update working memory, and (3) the ability to inhibit responses and reflexes. Different researchers use different labels for these EFs, ranging from more specific to broader. Early research on EFs (Duncan, Johnson, Swales, & Freer, 1997; Logan, 1985; Miyake et al., 2000; Rabbitt, 1997; Smith & Jonides, 1999) looked at the three core EFs as fundamental building blocks of controlled cognition and defined the three EFs with these three narrow labels: shifting, updating, and inhibition. These labels, however, are limited and later research has argued towards broadening the scope of these three core EFs and the labels associated with them. For example, Diamond (2013) suggests that better terms would be: cognitive flexibility instead of shifting, working memory instead of updating and inhibitory control instead of inhibition. Although the broader terms are valuable we will mainly use the labels as defined by Miyake et al. (2000) because we are mostly interested in EFs as fundamental building blocks and in order to keep unity in terminology between our paper and the original.

The first core EF, shifting or cognitive flexibility, refers to the ability to switch between performing different tasks, operations, or ‘mental sets’ (Monsell, 2003). This is often understood to involve disengaging from an irrelevant task or mode and subsequently shifting engagement to a more relevant task or mode. Although shifting is often conceived as switching between tasks, it is also very common to require shifting within a task. For example, to safely get to your destination while driving a car it is crucial to be able to switch between attending the traffic situation and navigating to your destination.

Updating refers to the ability to maintain a correct representation of the current situation in working memory. This requires the maintenance of currently important information as well as the replacing of no longer relevant information. A crucial element of the Updating function as identified by Miyake et al. (2000) is that it uniquely refers to active working memory functioning and does not include passive storage of information. For example, again in driving, it would include updating the current speed after seeing a speed sign but it would not include changing the speed based on subtler road-design features (e.g., trees alongside the road aimed at slowing drivers down).

The final core EF, inhibition, refers to the ability to actively prevent dominant or automatic responses or actions from occurring when needed. Inhibition is a broad term which is also used in many other areas of research, for example in neuroscience to refer inhibitory (in contrast to excitatory) effects of one neuronal population on another. However, inhibition in this sense is only used to refer to the active and deliberate prevention of responses, habits or reflexes. Additionally, the use of the term inhibition is only conceptually and does not mean to imply a mechanism. It does not mean to suggest that inhibition is achieved by suppressing (i.e., lowering the activation) of the automatic response, it is also plausible that the suppression is achieved by boosting the alternative (Kimberg & Farah, 1993) or, perhaps, through a combination of both processes. For a final driving example, inhibition of the distracting effect that a conversation with someone on the passenger seat might have, could be accomplished by suppressing attention to the passenger or by boosting attention to the driving task.

As was already mentioned in the previous paragraph, EFs are usually defined in terms of their effects and usefulness in achieving active and controlled cognition. However, a description of cognitive mechanisms responsible for the EFs is often lacking. This is a rather limited view which adds to the big variation in conceptualization and used methods employed by researchers in different fields or even within a field (Jurado & Rosselli, 2007; Miyake & Friedman, 2012). Additionally, it results in a limited understanding of how to improve executive functioning (Morrison & Chein, 2011;

Shipstead, Redick, & Engle, 2012) and why deficits in EFs can lead to impaired cognitive performance and psychological disorders (Brown, 2006; Duijkers, Vissers, & Egger, 2016; Pennington, 1997). To our knowledge, our model is the first attempt at modelling all three core EFs in a single cognitive model (consisting of numerous smaller models) and attempting to arrive at one unitary mechanism for each of the three EFs. Because we only performed simple model fits we do not claim that the mechanisms we implemented for the three EFs are complete. However, our model can be very valuable in generating more precise hypotheses about these mechanisms, especially because we used the skill-based approach.

The skill-based approach is ideally suited for specifying the general mechanism underlying each of the three EFs because it is based on the same basic idea as the study of executive functions: a few basic abilities are responsible for a wide range of behaviour. These few basic abilities are the general skills in a skill-based approach model and the core EFs in the theory of executive function. This allows for a model to be created in such a way that one of the main difficulties in the study of executive function, task impurity, does not become an issue. Task impurity refers to the fact that EFs cannot be measured directly and always have to be embedded in a task (Miyake & Friedman, 2012). This makes it impossible to get a clean idea of an individuals' executive performance in a single task because the variance attributable to executive functioning will be hidden within task-related variance (e.g., differences in shifting ability in a task-switching experiment will be clouded by the specific proficiency in this task). EFs can only be measured by combining performance on multiple different tasks, allowing for the task-specific variance to be ignored and the EF to be extracted. This is exactly what the skill-based approach allows a modeler to do; it allows a modeler to create models of multiple tasks while keeping the generalizable aspect constant (i.e., the basic skill or the EF).

The exploration of the cognitive mechanisms underlying the three core EFs, also allows us to investigate to what extent executive function is an innate ability and to what extent it is a learned behaviour. There have been many studies investigating whether executive function can be improved. Many of these studies have shown that executive function can indeed be improved (Bergman Nutley et al., 2011; Diamond & Lee, 2011; Karbach & Kray, 2009; Klingberg, 2010). Working memory (i.e., updating) in particular seems to be sensitive to training (Gray et al., 2012; Kamijo et al., 2011). However, concerns have been raised about the ecological validity of these studies and the effects of this training often do not show much transfer (Blair, 2017; Morrison & Chein, 2011; Shipstead et al., 2012) indicating that these training programs only have a limited effect on the core EF. Overall, EF training seems to have an effect on performance, however the extent of this improvement and the reasons why are not clear. Our model can provide an answer to these questions, because these seemingly paradoxical results are not so surprising when they are considered through the lens of the skill-based approach.

One of the crucial assumptions of the skill-based approach is that every task is accomplished by selecting the appropriate (previously learned) skills. The tasks used to study each of the three core EFs are very similar and likely rely on the same basic skill (e.g., the Updating tasks all use the 'updating' skill and the Shifting tasks all use the 'shifting' skill). We consider these basic skills the procedural aspect of an EF. However, this procedural aspect is not capable of influencing behaviour by itself since all it can do is move information around the central workspace. In order to achieve what the procedural knowledge sets out to do, it requires the tools to accomplish it. In the context of the PRIMs architecture, these tools are the architectural mechanisms that the procedural knowledge acts upon. We consider these architectural mechanisms the automatic aspect of an EF. The procedural aspect of an EF can be improved either through exploration of the most effective operators or by training the operators through production compilation; however, PRIMs does not cover to what extent architectural mechanisms can be improved through training. The difference in trainability of these two aspects explains why training on one EF task can only occasionally improve performance on a secondary task. Although both tasks measuring the same EF rely on the same architectural mechanism, training does not result in improved performance because the architectural mechanisms (the automatic aspect of an EF) cannot be improved. This means that training is only possible when both tasks require the exact same basic skill. If they do not rely on the same basic skill (this can also be the case for tasks using the same EF), then training will not transfer across the tasks. In the perspective of the skill-based approach, training effects are caused by training the basic skills required to do an EF task and not the underlying automatic mechanisms. This perspective also has implications for how individual differences in EFs might arise. Individual EF differences are often exclusively considered to be caused by stable fundamental differences in cognitive functioning (the automatic aspect in the skill-based approach perspective) (Carlson, Moses, & Claxton, 2004; Friedman & Miyake, 2017; Miyake & Friedman, 2012). However, learned basic skills might be an additional important underlying source of these individual differences.

1.4. Current study

The current study has two main goals: (1) test the proposed modifications aimed at improving skill reusability and (2) test the perspective of the skill-based approach on the nature of EF. The first goal will be accomplished by creating eight models of simple executive function tasks. These tasks are very similar to the tasks used by Miyake et al. (2000). Being able to successfully model these tasks using the skill-based approach suggests that our modifications have been successful and that creating models using the skill-based approach is possible. The model created this way will be an extensive model of executive functions which can suggest possible cognitive mechanisms. Additionally, the skill-based approach predicts that differences in how well the required basic skills are learned can be a large contributor to individual differences on EF tasks. This hypothesis will be tested by running the models with different levels of training on the basic skills and comparing the individual differences present in the models because of these differences to human data.

2. Method

2.1. Data collection procedure

The data was collected as part of a larger study on the correlations between the basic EF tasks and higher-level tasks similar to the design by Miyake et al. (2000). Participants completed the nine basic tasks in one first session and the higher-level tasks in a later second session. Unfortunately, this larger study was never published and certain details about the data collection procedure are therefore lost. This is not ideal; however, the information we do have is sufficient for our current purposes because we only perform simple model fits which do not rely on intricate manipulations or small details.

The data for the larger study was collected in a lab in the Psychology department of the University of Groningen in two sessions with several participants participating simultaneously. Participants were seated in separated cubicles and did not interact during the session. For our current purposes we will only look at the nine basic tasks included in the first session. The experiments were run and programmed using the experimental software E-prime (Psychology Software Tools, 2012). All experiments were conducted on a computer with the Windows operating system and participants provided responses by pressing keys on a keyboard.

In total 73 participants took part in the experiment. These participants were students of the University of Groningen who received partial course credit for participating. Participants signed an informed consent form before taking part in the study and ethical permission was acquired from the ethics committee psychology of the University of Groningen. Seven participants only completed six or fewer tasks and were therefore removed from the dataset, leaving 66 participants in the final dataset on which we performed the analyses.

2.2. The tasks

The final collection of tasks we modelled included eight tasks: three shifting tasks, three updating tasks, and two inhibition tasks. The original task battery also included a third inhibition task, a stop-signal task, however there was a problem during data collection which prevented the accuracy from being logged correctly during no-go trials. Because of this, we decided to exclude the stop-signal task from further analysis and only included the eight tasks which did not experience any problems. Four of the eight tasks in our battery were also used by Miyake et al. (2000: the keep track task, the letter memory task, the antisaccade task, and the Stroop task. Some of them were slightly modified to better fit the data collection setting (e.g., pen and paper responses were changed to keyboard responses). Additionally, it included four tasks that were not included in Miyake et al. (2000): the category-switch task, the colour-shape task, the colour-letter task, and the spatial two-back task.

The following three tasks were used as shifting tasks:

Category-switch. In the category-switch task, a word was presented on the screen. The participant was asked to answer one of two questions about the presented stimulus, either (1) “is the stimulus alive?” or (2) “is the stimulus bigger than a football?”. What question was asked was determined randomly for each trial. In order to successfully perform this task, a participant was required to quickly switch between the two possible task sets. Additionally, 150 ms before the stimulus was presented, the to-be asked question was indicated by presenting the category on the screen (“alive” for the first question or “size” for the second question). The cue and the stimulus remained on the screen until a response was given. The response was given by pressing ‘f’ on the keyboard for yes and ‘j’ for no. The participants did one practice block of 20 trials and two experimental blocks of 54 trials each for a total of 108 experimental trials. Participants’ performance on this task was measured by calculating the difference between the average reaction time (RT) on trials on which a switch was necessary and trials on which no switch was necessary (i.e., switch cost).

Color-letter. The colour-letter task is highly comparable to the number-letter task in Miyake et al. (2000) and in Rogers & Monsell (1995). Participants were required to switch between determining whether a letter was a vowel or a consonant or between whether the letter was blue or red. The letter was presented in one of four quadrants on the screen. The trial type was indicated by the position of the letter. When the letter was in the two top quadrants, the participant should indicate whether the letter was a vowel or consonant. When the letter was in the bottom two quadrants, the participant was to indicate the colour. The position of the letter was different on every trial and rotated clockwise over the quadrants, always starting in the lop left quadrant. A letter was on the screen for 5 seconds or until the participant gave a response. This response was given by pressing ‘f’ to indicate vowel or red and ‘j’ to indicate consonant or blue. The participants did one practice block of 24 trials (6 full rotations) and two experimental blocks of 56 trials each (14 full rotations) for a total of 112 experimental trials. The performance measure taken from this task was the difference between average RT in trials on which the letter is in the first or third quadrant (switch trials) and trials on which the letter was in the second or fourth quadrant (no-switch).

Color-shape. The colour-shape task was very similar to the other two tasks in that participants had to switch between two ways of responding to the stimulus. In this task, the stimulus on the screen was a figure (triangle or square) in a certain colour (blue or red). The participants had to indicate the identity of the shape or the colour of the shape. The aspect to attend was determined randomly on every trial. The type of trial was indicated 150 ms before the shape was presented by either presenting ‘k’ (which stands for ‘kleur’ in Dutch – ‘colour’ in English) or ‘v’ (which stands for ‘vorm’ in Dutch – ‘shape’ in English). The informative cue made it easier to understand which aspect to attend. The cue and the stimulus stayed on-screen until a response was given. This response was given by pressing a key on the keyboard, the ‘f’ to indicate either red or triangle and the ‘j’ to indicate blue or circle. The participants did one practice block of 12 trials and two experimental blocks of 54 trials each for a total of 108 experimental trials. Similar to the category-switch task, the performance measure was the difference in RT between switch trials and no-switch trials.

The following three tasks were used as updating tasks:

Keep-track. The basic design of the keep-track task we used was very similar to the design used by Miyake et al. (2000) which was originally adapted from Yntema (1963). In the keep-track task, participants were required to keep track of the most recent item of several categories, doing this successfully required participants to keep the most recent item of the categories in mind while frequently updating which item was the most recent one. The difference between the task our participants did and the one used by Miyake et al. (2000) was the number of categories that had to be updated, four in our task compared to either two or three in the original task. A trial started with the four target categories at the bottom of the screen. Then, fifteen words were successively presented in a random order for 1500 ms each while the target categories remained visible on the screen. Distractor words not belonging to any of the target categories were also among the presented words. At the end of a trial, the participant was asked to indicate the final item of every category by typing in the answer. Prior to the start of the task, the participants were made familiar with the items and their respective category. The total stimulus set consisted of five categories with six items each. The participants did one practice block with three trials and one experimental block with fifteen experimental trials. The performance measure taken from this task was the average accuracy with which the final items were recalled.

Letter-Memory. The letter-memory task was also very similar to the task in the original study which was adapted from Morris & Jones (1990). In this task, participants were required to remember the final three letters in a series. This task also required continuous updating of the to-be recalled items.

The series was presented serially for 2000 ms each and no letters in the series could repeat. No instructions were given about how to perform the task. This differs from the original study which instructed the participants to continuously rehearse the final letters out loud. The length of the presented series varied randomly between 5, 7, 9, or 11 letters to ensure that the participants would be engaged from the start of the series. The participants indicated their response at the end of a trial by typing in the letters in the order in which they were seen. A reply was only counted as correct if the letter as well as the location was provided correctly (e.g., the final letter of the series would need to be the final letter of the responses). The participants did one practice block of three trials and one experimental block of 18 trials. The performance measure taken from this task was the average accuracy in providing the final three letters (and their location) of the series.

Spatial two-back. The final updating task was not included in the original study. The spatial two-back task (Ellis, Silberstein, & Nathan, 2006) is a specific version of the more standard n-back task (Jaeggi, Buschkuehl, Perrig, & Meier, 2010; Redick & Lindsey, 2013) in which the participant is required to remember spatial locations instead of letters or other more declarative items. A trial of this task started with ten white squares arranged in columns alternating between three and two rows (see Figure 1a). During a trial, the squares would turn black in a semi-random order one at a time (see Figure 1b). A square would be black for 500 ms after which it went back to white and all squares would be white for 1000 ms until the next square would turn black. The sequence was determined semi-randomly in order to ensure that half of the trials would be a two-back trial. To do the task successfully, participants had to remember the two previous locations and compare the current location to the two-back location (i.e., the location before the previous). Participants were required to give a response on every trial. They indicated that the current location was the same as the two-back location by pressing ‘j’ on the keyboard and indicated that it was not the same by pressing ‘f’. Every block consisted of 24 trials (i.e., 24 locations). Participants did one practice block and four experimental blocks for a total of 96 experimental trials. The performance measure taken from this study was the proportion of correct responses (either correctly detecting a two-back or correctly detecting that the trial was not a two-back).

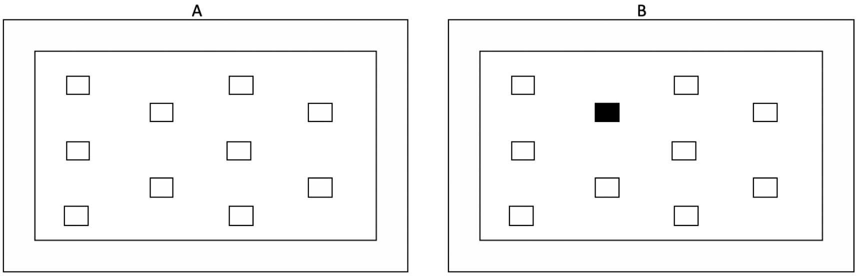

Figure 1. Illustration of the spatial two-back task. Figure 1a shows the configuration of the ten squares in the task. Figure 1b shows an example of how the current location was indicated by 'lighting up' one of the squares.

The following two tasks were used as inhibition tasks:

Antisaccade. The main goal of the antisaccade task is to inhibit the reflexive saccadic response to a new visual stimulus. The antisaccade task used here and in Miyake et al. (2000) was adapted from the task described in Roberts, Hager, & Heron (1994). A trial of the antisaccade task started with a fixation cross lasting between 1500 and 3500 ms. A small cue was then presented on one side of the screen for 225 ms, followed by the larger target which was on screen for 150 ms on the opposite side of the screen. The cue was a small black square. The target was an arrow within a small square that could point left, up, or right. The task for the participant was to indicate the direction of the arrow. Because the target was on screen for such a short amount of time directly following the onset of the cue, the reflexive saccade to the cue had to be inhibited in order to correctly identify the target. The responses were given by pressing the arrow key on the keyboard corresponding to the arrow direction of the target. The participants performed one block of 22 practice trials and one block of 90 experimental trials. Participants' performance was measured by calculating the average accuracy with which the target was identified.

Stroop. The Stroop task (Stroop, 1935) is a well-known task in cognitive psychology in which the automatic process of reading a word has to be inhibited. The Stroop task analysed here was a slightly adapted version from the task described in Miyake et al. (2000). This was done to make it suitable for a setting in which vocal responses could not be provided. The task for the participant was to indicate the colour of a presented word. This version of the task included three types of trials: (1) congruent trials in which a colour word was presented in the same colour as the word (e.g., the word 'blue' in blue), (2) conflict trials in which a colour word was presented in a different colour (e.g., the word 'blue' in red), and (3) neutral trials in which the presented word was not a colour word in a random colour (e.g., the word 'hat' in green). Every trial type was presented 60 times. A block consisted of a mix of the different trial types which were determined randomly by drawing from the possible trial types without replacement. On every trial, the target word was presented in the middle of the screen flanked by two response options for 2500 ms or until a response was given. The response options were the correct answer and a randomly determined other colour (this could also be the conflicting colour word in the incongruent trials). The response was indicated by pressing the 'f' key for the left response option and the 'j' key for the right response option. Participants did one practice block of 18 trials and four experimental blocks of 45 trials each for a total of 180 experimental trials. Participants' performance on this task was measured by calculating the difference in reaction time between the conflict and congruent trials.

3. Models and Model Results

3.1. General structure of the model

The tasks described above show the typical "unity and diversity" often found in executive function tasks. There are a lot of similarities between the tasks aimed at measuring the same EF and much lower (apparent) overlap between tasks measuring different EFs. We attempted to achieve this unity in our model by assuming that a core EF is accomplished by a basic skill and reusing this basic skill in all tasks using that core EF. Therefore, we assumed that there would be three basic skills underlying the performance on these eight tasks. A basic "shifting" skill, a basic "updating" skill and a basic "inhibition" skill. Additionally, the tasks share further non-EF related similarities because all tasks require the participants to provide a response and three tasks require participants to read a word and encode it into WM. Therefore, two more basic skills underlie the models: a "respond" skill and a "read" skill. In total, five basic skills were used in the modelling of these eight tasks: "shift", "update", "inhibit", "read", "respond" (see Table 2 for the overlap). Finally, all models required a certain amount of specific procedural knowledge to be able to perform the modelled tasks, which needed to be created separately for every model. See the appendix for more details about the model building process and how the changes we proposed to the skill-based approach and the cognitive architecture facilitated this process.

Table 2. Skill overlap between the modelled tasks. All tasks measuring the same EF share their respective basic skill, all tasks share the respond skill, and three tasks share the read skill.

Task/skill	*Shift*	*Update*	*Inhibit*	*Read*	*Respond*
Category-Switch	X				X
Color-Letter	X				X
Color-Shape	X				X
Keep-Track		X		X	X
Letter-Memory		X		X	X
Spatial two-back		X			X
Antisaccade			X		X
Stroop			X	X	X

3.2. Resulting models and model fits

3.2.1. Basic Models

The above described modelling structure resulted in eight models built around three basic EF skills and two additional general skills. We will now describe the basic functioning of the EF skills and with which architectural mechanisms they interact. Additionally, we will describe how these skills were integrated in the eight tasks and how well they fit the data. The basic skills vary in their procedural complexity: the updating skill is the most procedurally complex (i.e., it has the most operators) while the two other skills are procedurally less complex since they rely more on architectural mechanisms. Although the basic skills we created are sufficient to model all eight tasks included in this study, this does not guarantee that they are also sufficient for all tasks requiring these EFs.

3.2.2. Shifting Models

The shifting task we modelled all required the model to shift between two tasks. The procedural knowledge required for the shifting skill is fairly straightforward as can be seen below. This skill only consists of two operators which perform the two basic operations required for a successful shift. The first operator retrieves a new task set (the task set associated with either ‘task 1’ or ‘task 2') and the second operator puts this task set in the right place so that it can influence further processing. In essence, shifting is accomplished by setting a binding (**characteristic*) to a certain value depending on the retrieved task set and because this binding is embedded throughout the model it will influence the model to perform either ‘task 1’ or ‘task 2’. This binding influences processing by serving as input for memory retrievals and in condition checking.

```
define skill shift {

      operator retrieve-task-set {
      RT1 = nil
      ==>
      task-set -> RT1
      *task -> RT2
      V1 -> RT3
      }

      operator set-characteristic {
      RT1 <> nil
      ==>
      RT3 -> G2
      RT4 -> *characteristic
      *next-skill-shift -> G1
      }
}
```

The category-switch model (see Figure 2a) consists of two basic skills (*shift* and *respond*) and two skills specific for this model (*select-next-skill-category-switch* and *retrieve characteristic*). These skills were combined in the following way. The model starts a trial with the *select-next-skill-category-switch* skill. This skill determines whether the current trial is a switch trial or not. If necessary, when the shift is completed, the model returns to the *select-next-skill-category-switch* skill and waits for the presentation of the word. After presentation of the word, the *retrieve characteristics* skill determines the answer of whether the presented word is alive or bigger than a football (depending on the trial) by performing a memory retrieval. Finally, the *respond* skill carries out the final response.

The colour-letter model (Figure 2b) also consists of two basic skills (*shift* and *respond*) and two skills specific to this model (*select-next-skill-colour-letter* and *attend-colour-or-type*). The colour-letter model is combined in the following way. The colour-letter task starts with a blank screen for 150 ms, during this the model starts with the *select-next-skill-colour-letter* skill which determines whether the current trial is a switch trial or not. If they are not the same, the model will go to the *shift* skill. After completing the shift, the model will return to the *select-next-skill-colour-letter* and waits for the presentation of the letter. If they are the same, the model simply waits for the letter presentation. After the letter is presented, the model will go to the *attend-colour-or-type* skill which either attends the letter itself or the colour in which the letter is presented. Finally, after either the letter-type or colour is determined, the response skill gives the appropriate response.

The colour-shape model (Figure 2c) also consists of two basic skills (*shift* and *respond*) and two skills specific to this model (*select-next-skill-colour-shape* and *attend-colour-or-shape*). These skills are combined for the colour-shape task in the following way. The model starts in the *select-next-skill-colour-shape* skill which determines whether a trial is a switch trial or not. When the shift is completed, the model will return to the *select-next-skill-colour-shape* and waits for the shape to be presented. If the previous task and cue are the same, the model does not need to switch and simply waits for the shape to be presented. After presentation of the shape, the *attend-colour-or-shape* skill determines the colour or the identity of the shape. Finally, the *respond* skill gives the final response. Please see the appendix for a more detailed description of the three shifting models.

3.2.3. Shifting model fit

We assessed the fit of the models by comparing the performance of the model to the performance of the participants on the inhibition tasks in the larger study on EF. We did this by means of a visual inspection (see Figure 3) and by a simple statistical comparison of the models and the data. This was done

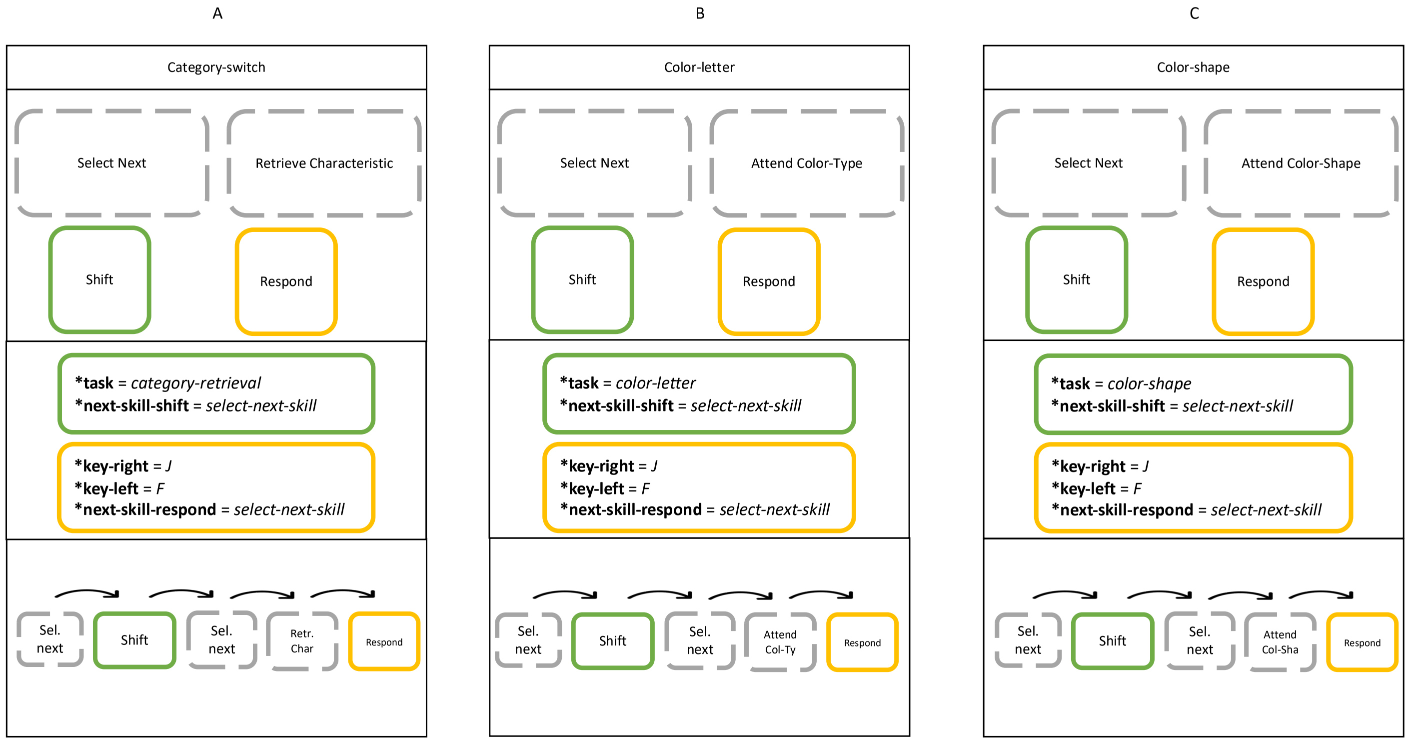

Figure 2. *Overview of the shifting models*. This overview shows the skills used in shifting models in the top panels, how they were instantiated (i.e., which bindings were added to the model) in the middle panel, and how they were arranged for the task on a switch trial in the bottom panels. (A) depicts the category-switch model, (B) the colour-letter model and (C) the colour-shape model.

by creating linear regression models of both the data produced by the model as well as the data produced by the human participants and comparing the intercepts and slopes of these regression models. These regression models were created with the statistical software ‘R’ (R Core Team, 2017) and the package ‘lme4’ and ‘lmerTest’ (Bates, Mächler, Bolker, & Walker, 2015; Kuznetsova, Brockhoff, & Christensen, 2017). The models are very simple and do not include any random slopes or intercepts. For the shift tasks, we analysed both the accuracy and the reaction time data. This means that for every task we ran four separate linear models, two on the human data (accuracy and RT) and two on the model data (accuracy and RT).

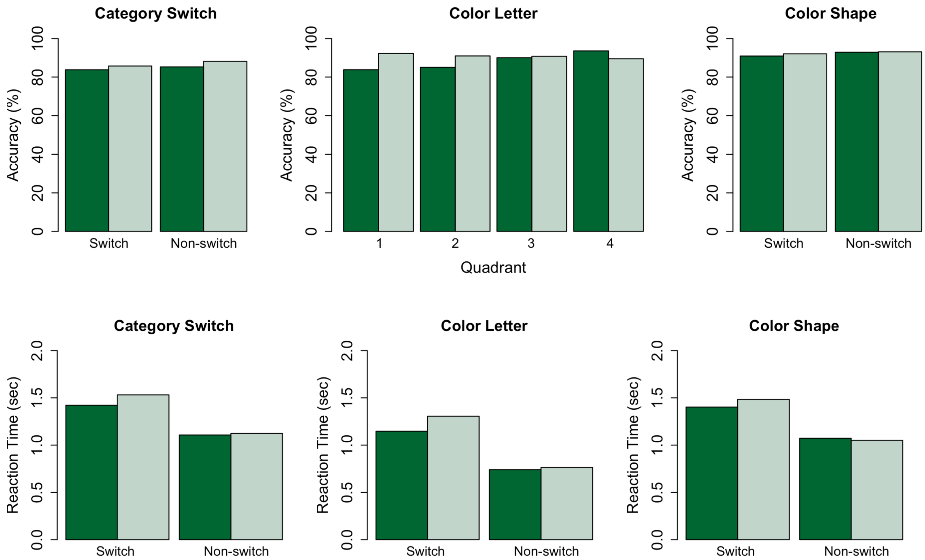

Figure 3. *Shifting model fit.* The model fits of the three models created for the shifting tasks. The human data in the dark green, the model in the light green. Figures show the average accuracy and average reaction times in switch and non-switch trials for the category-switch task and the colour-shape task. For the colour-letter task, the figure shows the average accuracy and RT in the four quadrants that the letter could be presented in: quadrant 1 and 3 are switch trials.

Table 3. Linear model estimates for the model and the human data. The estimates produced by the linear regression models for the three Shifting tasks. Significant slopes at the p < .05 level are indicated with an *.

	Model				Human data			
	Estimate	SE	t value	p value	Estimate	SE	t value	p value
Category switch								
Accuracy								
(Intercept)	88.2	0.6	147.8	> .001*	85.1	0.9	92.9	> .001*
Switch	-2.8	0.6	-3.3	.001*	-1.1	1.3	-0.9	.376
Reaction times								
(Intercept)	1123	3.5	314.8	> .001*	1115	18.4	60.5	> .001*
Switch	409.7	5	81.2	> .001*	295.7	26.1	11.4	> .001*
Color letter								
Accuracy								
(Intercept)	89.5	0.5	164.8	> .001*	93.0	2.3	41.3	> .001*
Switch	1.3	0.6	2.0	.04*	-2.4	2.6	-0.9	.353
Letter	1.5	0.6	2.4	.02*	-7.8	2.6	-3.0	.003*
Reaction times								
(Intercept)	676.6	4.6	146.8	> .001*	664.7	26.7	24.9	> .001*
Switch	542.2	5.3	103.1	> .001*	406.6	30.8	9.8	> .001*
Letter	173.7	5.3	32.6	> .001*	145.7	30.8	4.7	> .001*
Color shape								
Accuracy								
(Intercept)	93.1	0.3	323.5	> .001*	92.9	1.6	57.8	> .001*
Switch	-1.0	0.4	-2.5	.01*	-2.1	2.3	-0.9	.359
Reaction times								
(Intercept)	1050.8	3.5	298.3	> .001*	1063.1	38.5	27.6	> .001*
Switch	432.8	5.0	86.9	> .001*	335.6	54.4	6.1	> .001*

We assessed the fit of the shifting models by looking at the average accuracy on switch and non-switch trials and the reaction times of these two types of trials. The visual inspection of the fits suggests that both the accuracy and reaction times of the models are highly comparable to the accuracy and reaction times of the human participants. Both the models and participants perform the shifting tasks with around 80% to 90% accuracy and both show the crucial increased reaction times for switch trials. The models, however, seem to consistently have slightly higher switch costs than the human participants in all three models. However, overall the models and the human participants show large similarities, both in terms of absolute performance (similar accuracies and RTs) as well as in the crucial effect of longer RTs and slightly lower accuracy in switch trials. These conclusions are supported by the linear regression models (see Table 3).

3.2.4. Updating Models

The updating skill is the most complex skill and consists of six operators as can be seen below. These operators accomplish the basic updating step in the three updating tasks. This basic updating step can have two general characteristics, it can either (1) update a single value or (2) shift multiple values. Updating a single value is a fairly basic operator since it only consists of creating one new binding. Shifting (not to be confused with the shifting EF) is more complex and represents the updating necessary in for example the letter-memory task: adding a new value to the existing list, removing the oldest value, and shifting the other values over one position (resulting in two or three new bindings). Note that this definition of updating does not include removal of the old binding but merely creating a new more recent one. Updating single values is the most basic function of the updating skill and needed to be done in all three updating tasks, while the two shifting operators were only used in one task each.

```
define goal update {

      operator update-first {
      WM1 <> nil
      ==>
      *current-target -> *first-item
      nil -> WM1
      *next-skill-update -> G1
      }

      operator update-second {
      WM1 <> nil
      ==>
      *current-target -> *second-item
```

```
        nil -> WM1
        *next-skill-update -> G1
        }

        operator update-third {
        WM1 <> nil
        ==>
        *current-target -> *third-item
        nil -> WM1
        *next-skill-update -> G1
        }

        operator update-fourth {
        WM1 <> nil
        ==>
        *current-target -> *fourth-item
        nil -> WM1
        *next-skill-update -> G1
        }

        operator shift-values-three {
        WM1 <> nil
        ==>
        *second-item -> *first-item
        *third-item -> *second-item
        *current-target -> *third-item
        *next-skill-update -> G1
        }

        operator shift-values-two {
        WM1 <> nil
        ==>
        *second-item -> *first-item
        *current-target -> *second-item
        *next-skill-update -> G1
        }

}
```

The keep-track model (see Figure 4a) consisted of three basic skills (*read*, *update*, and *respond*) in addition to one keep-track specific skill (*category search*). These skills were combined in the following way. At the start of a trial, the first item was read by the read skill. Subsequently, the *category search* skill determined to which category this item belonged through means of a memory retrieval. Afterwards, the *update* skill updated the binding associated with this category (e.g., the **third-item)*. This process repeated until the end of the trial. At the end of a trial, the final responses were given by the *respond* skill which was done by retrieving the value associated with each of the bindings and typing them on a keyboard. In order to successfully perform the keep-track task, the first four operators of the *update* skill were needed.

The letter-memory model (Figure 4b) also consisted of three basic skills (*read*, *update*, and *respond*) in addition to one letter-memory specific skill (*count*). These skills were combined for the letter-memory task in the following way. At the start of a trial, the *read* skill read the first letter. After this, the *count* skill placed a value in WM1 depending on how many stimuli had already been presented (e.g., a '1' for the first letter and a 'more-than-three' for the fourth letter and up). Subsequently, the *update* skill performed the correct update based on the value stored in WM1, it fired one of the *update-x* operators if this value was lower than three and it fired the *shift-values-three* if this value was 'more-than-three'. This process repeated until the final letter was presented. Finally, the *respond* skill took care of the responses at the end of the trial in the same way as in the keep-track model.

The spatial two-back model (Figure 4c) consisted of only two basic skills (*update* and *respond*) and one two-back-specific skill (*compare*). These skills were combined for the spatial two-back task in the following way. When a square was presented, the *compare* skill compared the location of the current square to the location of the two-back square. After this, the *respond* skill gave the response associated with the outcome of the *compare* skill (pressing 'j' for yes and 'f' for no). Finally, the *update* skill performed the update, either with the *update-first* or *update-second* operator for the first two locations of the trial or with the *shift-values-two* operator for the remainder of the trial. This process repeated until the end of the trial.

3.2.5. Update model fit

Similar to the shifting models, we assessed the fit of the updating models by comparing the model performance to the performance of the participants on the same updating tasks in the larger study on EF. This was also done by a visual inspection supplemented with a simple statistical analysis. This analysis included creating a linear regression model with one dependent variable and one continuous independent variable per task.

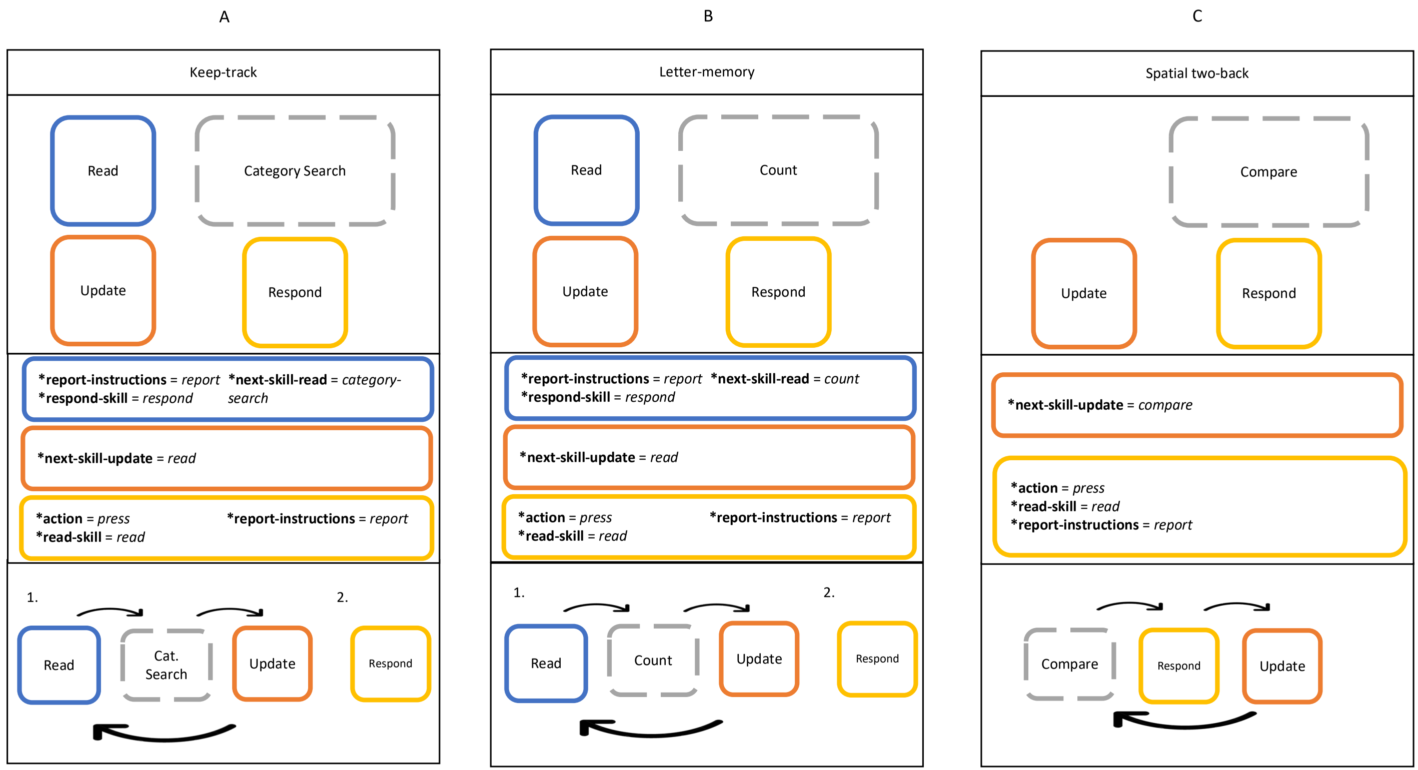

Figure 4. *Overview of the updating models.* (A) represents the keep-track model, (B) represents the letter-memory model, and (C) represents the spatial two-back model. The top panels show the skills used, the middle panels show how they were instantiated (i.e., which bindings were added to the model), and the bottom panels show the order in which they are carried out.

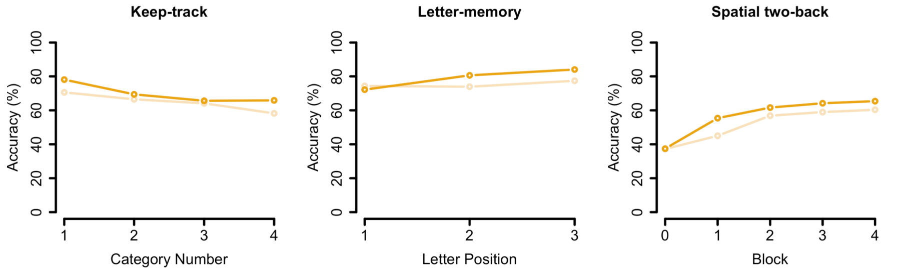

Figure 5. *Update model fit.* Figure shows the model fits in the updating tasks with human data in dark orange and model data in light orange. Shown for the keep-track task is average accuracy for the four category numbers, for the letter-memory task is average accuracy for the three letter positions, for the spatial two-back performance over the course of the experiment (divided in blocks with '0' being the practice block).

This process revealed that the models performed the three updating tasks very similarly to the human participants (Figure 5). All three tasks included a different manipulation; however, our models showed the same performance patterns as the human participants in all three tasks. The models showed a significant slope in the same direction as the participants in the keep-track and the spatial two-back over the respective independent variables tasks as well as showing a non-significant positive slope in the letter-memory task as a function of letter position (as can be seen in Table 4). Although it is a very rudimentary model fitting procedure, the models perform similarly to the human participants in the crucial aspects of these three tasks lending credibility to the validity of the three updating models.

3.2.6. Inhibition Models

The procedural knowledge included in the skill responsible for inhibition is less extensive than the update skill. It only includes two operators; the first operator retrieves the correct 'inhibition instructions' based on the current task the model is performing and the second operator activates the retrieved 'inhibition instructions' by placing it in WM1. These 'inhibition instructions' have spreading activation relationships (S_{ji}s) with certain operators in the model. Therefore, in our models, inhibition is accomplished by placing something in WM1 which spreads activation to operators. In both tasks we modelled, the instructions in WM1 could spread both positive and negative activation. This will likely be necessary in many tasks modelled in a production-based architecture because the model can never be idle (i.e., there is a production firing at all times). This means that it is just as important to boost the alternative path as it is to inhibit the undesired path.

Table 4. Regression coefficients of the Updating tasks. The estimates produced by the linear regression models for the three Updating tasks. Significant slopes at the p < .05 level are indicated with an *.

	Model				*Human data*			
	Estimate	SE	t value	p value	Estimate	SE	t value	p value
Keep track								
Accuracy								
(Intercept)	74.7	0.9	79.2	> .001*	83.1	2.4	34.3	> .001*
Category number	-3.9	0.3	-11.5	.001*	-4.1	0.9	-4.6	> .001*
Letter memory								
Accuracy								
(Intercept)	72.2	1.6	44.3	> .001*	67.1	3.1	21.6	.03*
Letter position	1.5	0.8	2.0	.06	5.9	1.4	4.1	.15
Spatial two-back								
Accuracy								
(Intercept)	40.3	0.7	55.0	> .001*	39.3	2.9	13.6	> .001*
Block	5.6	0.3	18.9	> .001*	7.1	1.2	6.0	> .001*

We chose to model inhibition using retrieval of an 'instruction' from DM because it can provide a simpler learning mechanism and it provides a straightforward way in which previously learned information can be brought to the current task. It provides a simpler learning mechanism because the only thing that needs to be learned is an association between this 'instruction' (e.g., inhibit-saccade) and the operators that need to be inhibited or boosted. Secondly, all these associations can easily be brought into the current task by simply retrieving this 'instruction' and placing it in WM1. Although it would be interesting to investigate how these associations are learned, for the current models we merely provided them by hand.

```
define skill prepare-to-inhibit {

       operator retrieve-preparation {
       G1 = *prepare-goal
       RT1 = nil
       RT2 = nil
       ==>
       *fact-type -> RT1
       *task -> RT2
       }

       operator activate-instructions {
       G1 = *prepare-goal
       RT1 <> nil
       RT2 <> nil
       ==>
       RT3 -> WM1
       *next-skill-prepare -> G1
       }

}
```

The antisaccade model consisted of two basic skills (*prepare-to-inhibit* and *respond*) and two specific antisaccade skills (*select-next-skill-antisaccade* and *determine target*). These skills were combined for the antisaccade task in the following way (Figure 6a). A trial of the antisaccade task started with a fixation cross, during this time the *prepare-to-inhibit* skill prepared for the upcoming stimuli by placing *inhibit-saccade* in WM1. After the *prepare-to-inhibit* skill, the model switched to the *determine target* skill. This skill waited for the arrival of the distractor and the target. Finally, the *respond* skill gave the final response by pressing the appropriate arrow key on the keyboard (left, up, or right).

The Stroop model consisted of three basic skills (*read*, *prepare-to-inhibit*, and *respond)* and three Stroop specific skills (*select-next-skill-stroop*, *determine colour*, and *choose hand*). These skills were combined for the Stroop task in the following way (Figure 6b). A trial of the Stroop also started with a fixation cross, during this time the *prepare-to-inhibit* skill prepared for the upcoming trial by placing *inhibit-reading* in WM1. After this, the model waited for the word to be presented. Depending on the activation of the operators the model could either select the *read* skill or the *determine colour* skill. Selecting the *read* skill could lead to a slowed reaction and selecting the *determine colour* skill would be the optimal choice. After detecting the colour of the word, the *choose hand* skill determines the correct hand to respond with. Finally, the *respond* skill would press the key associated with the correct hand.

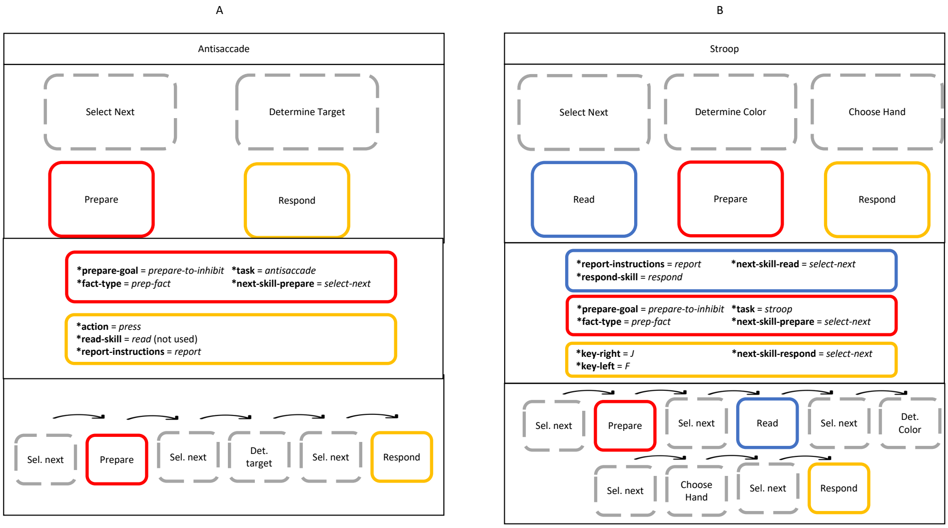

Figure 6. *Overview of the inhibition models*. The top panels show the skills involved in the two models, how they were instantiated (i.e., which bindings were added to the model) in the middle panel, and the bottom panel shows the way in which they were arranged for these models. (A) depicts the antisaccade model. (B) depicts the Stroop model. The bottom panel of (B) shows how the model works on a conflict trial in which it did not inhibit the prepotent read response.

3.2.7. Inhibition model fit

The antisaccade task did not include any manipulations that needed to be modelled and there were no important aspects in the reaction time data since all participants needed to do was press a key on the keyboard after the target was presented. Therefore, we only performed the model fit procedure on the average accuracy of responding the arrow direction. We did this with a very simple linear model that only included an intercept. The Stroop task included a more complex manipulation in the three different trial types. To analyse this data, we performed linear regression models on the accuracy and reaction time data with one three-level independent variable of Trial Type with

Conflict as reference level. Neither of these models included random slopes or intercepts.

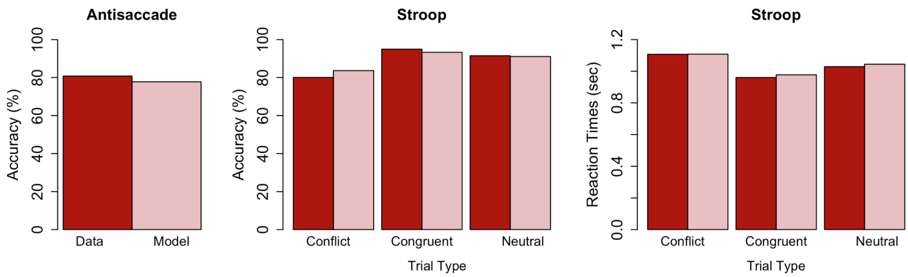

Figure 7. *Inhibition model fit.* Figure shows the average accuracy for the antisaccade and Stroop tasks and the average reaction time for the Stroop task for the human data and the model. The human data is in dark red with the model data in light red. Depicted for the Stroop tasks is the average accuracy and RT in the three different trial types (conflict, congruent, and neutral).

The analysis of the antisaccade task showed that the model and the human participants performed were very similar in their accuracy of reporting the direction of the arrow (see Figure 7). Additionally, the participants and the model showed the same hierarchy in trial difficulty in the Stroop task. For both the model and the participants the conflict trials were the most difficult (as indicated by the slowest RTs and lowest accuracies), followed by the neutral trials and, finally, the congruent trials were the easiest for both the model and the participants. These results suggest that the inhibition skill we created is cognitively plausible. The coefficients of the linear models can be found in Table 5.

3.3. Can the models explain individual differences?

To answer our second research question whether differences in basic EF skill proficiency can explain the individual differences present in these tasks, we ran the models with different levels of training on the EF skills (for more details see the ‘model training’ section in the appendix). In total, we ran 400 simulated participants with varying levels of training on the EF skills. We split these participants in two halves for every task, one half with the 50% best performing ‘participants’ (high performers) on this task and the other half with the 50% worst ‘participants’ (low performers) on this task.

Subsequently, we compared the average performance of these two halves with the average performance of the human participants who were split in the same manner (see Figure 8).

Table 5. Linear model estimates for the model and the human data. The estimates produced by the linear regression models for the three Shifting tasks. Significant slopes at the p < .05 level are indicated with an *.

	Model				*Human data*			
	Estimate	SE	t value	p value	Estimate	SE	t value	p value
Antisaccade								
Accuracy								
(Intercept)	77.7	0.3	254.3	> .001*	79.2	1.7	46.5	> .001*
Stroop								
Accuracy								
(Intercept)	83.7	0.3	253.2	> .001*	77.6	1.7	44.6	> .001*
Congruent	9.7	0.5	20.7	> .001*	17.2	2.5	7.0	> .001*
Neutral	7.4	1.5	15.9	> .001*	12.9	2.5	5.2	> .001*
Reaction times								
(Intercept)	1108.2	6.2	251.2	> .001*	1108.9	29.4	53.5	> .001*
Congruent	-31.1	6.2	-21.0	> .001*	-140.5	29.4	-4.8	> .001*
Neutral	-63.9	6.2	-10.2	> .001*	-71.4	29.4	-2.4	.016*

Overall, this analysis shows high similarity in how the high and low performing groups in the simulated participants and the human participants perform. In almost all tasks, the high performing models perform at the same level as the high performing participants and the low performers perform similarly worse in the models and the human data. The only exceptions are that the models do not capture the low performers in the two-back task and the Stroop task. More details about the analysis can be found in the 'individual differences model fit' section of the appendix. These results show that manipulating the level of training of the basic EF skills (representing different levels of proficiency) leads to the same differences in performance that are found in human data. However, this analysis only considered the overall performance of two groups and not the performance on the individual level. To analyse how well our models fit the data on the individual level, we additionally calculated the correlations between the tasks for the model and the data. More specifically, the correlation between the average performance of the (simulated) participants on all tasks. The results of this can be seen in Table 6 (for the data) and Table 7 (for the model).

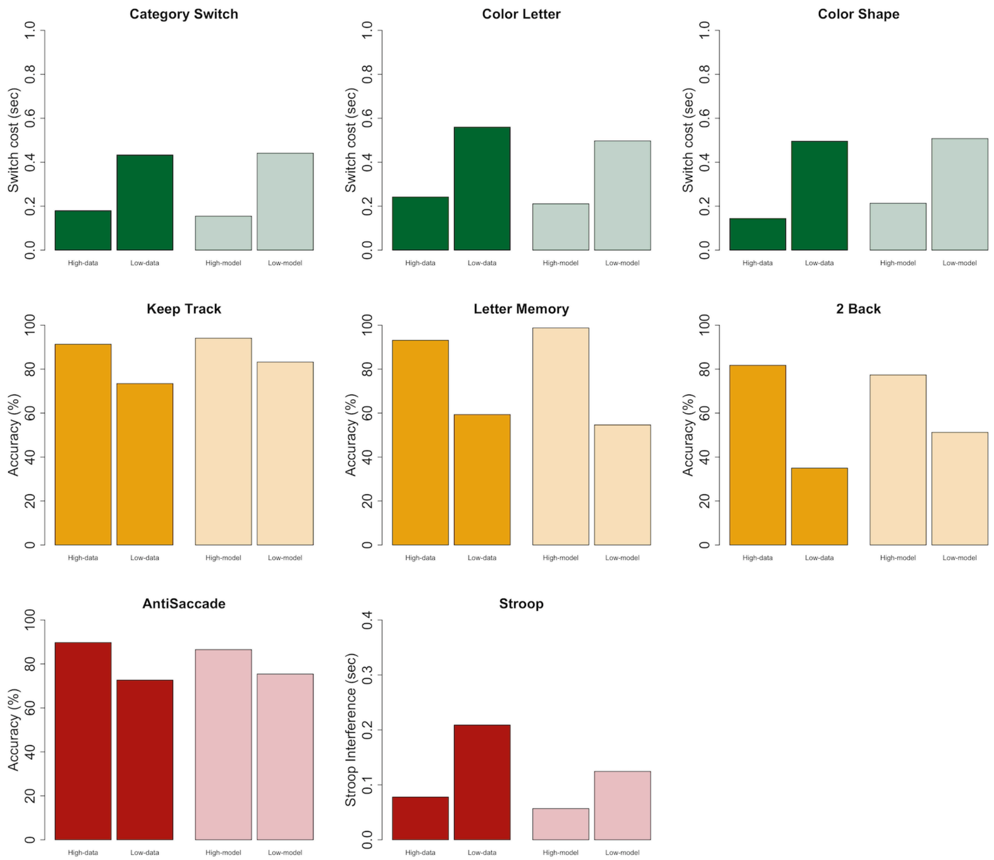

Figure 8. Comparison of the high and low halves. Figure showing the top 50% performers ('high') and the bottom 50% performers ('low') in the model and the data. The data is displayed in the dark colours and the model in the light colours.

The correlations between the tasks in the human data are very low with no correlations that could be considered strong (i.e., over .4). There were four significant correlations, the correlation between category-switch and colour-shape (r (64) = .25, p = .04), between colour-letter and colour-shape (r (64) = .37, p = .001), between letter-memory and the spatial two-back task (*r* (64) = .29, *p* = .018). The final correlation is between tasks measuring two different EFs, namely colour-shape and Stroop (*r* (64) = .32, *p* = .008). The final correlation was surprising and might be due to the fact that both tasks heavily rely on colour processing.

Table 6. Pearson correlation coefficients for the eight tasks in the human data. Shifting tasks are green, updating tasks are yellow and inhibition tasks are orange. Significant correlations where p < .05 are indicated with an *.

Task	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1. Category-switch	1							
2. Color-letter	.14	1						
3. Color-shape	.25*	.37*	1					
4. Keep-track	.20	.11	.17	1				
5. Letter-memory	-.07	-.05	.09	.13	1			
6. Spatial two-back	-.09	-.06	.02	-.21	.29*	1		
7. Antisaccade	-.05	-.01	-.16	.10	.01	-.11	1	
8. Stroop	.24	.17	.32*	.04	-.07	.04	.12	1

Table 7. Pearson correlation coefficients for the eight tasks in the model data. Shifting tasks are green, updating tasks are yellow and inhibition tasks are orange. Significant correlations where p < .05 are indicated with an *.

Task	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1. Category-switch	1							
2. Color-letter	.91*	1						
3. Color-shape	.91*	.90*	1					
4. Keep-track	-.02	-.02	-.01	1				
5. Letter-memory	.01	.02	.02	.70*	1			
6. Spatial two-back	.01	.03	.03	.73*	.96*	1		
7. Antisaccade	.04	.06	.05	-.09	-.02	-.02	1	
8. Stroop	.14*	.13*	.13*	-.08	-.03	.02	.55*	1

The correlations calculated based on our model data are much stronger. The model data contains ten significant correlations. The three shifting tasks are highly correlated (all rs > .9 and ps < .001), the three updating tasks are highly correlated (rs > .7 and ps < .001), and finally, the inhibition tasks are also strongly correlated (r (398) = .55, p < .001) The only significant correlations between tasks outside an EF are the correlations between the Stroop task and the three shifting tasks: category-switch (r (398) = .14, p = .005), colour-letter (r (398) = .13, p = .012) and colour-shape (r (398) = .13, p = .012).

These results provide two crucial insights. Firstly, it provides additional evidence for the validity of the individual models. The average performances on the tasks associated with the same EF are highly correlated suggesting that the skills we created are central to these tasks. Secondly, it shows that simulated low EF proficiency strongly influences performance on the tasks associated with that EF and, crucially, this impaired performance is limited to the tasks associated with that EF. Additionally, the correlations found in the human data follow largely the same pattern with high(er) correlations in the tasks associated with the same EF. However, the absolute values of these correlations are very different, the tasks belonging to the same EF are all correlated in the model (compared to only some of them in the data) and the magnitude of these correlations is much larger in the model. The very strong correlations between the tasks are caused by the fact that a well-trained skill (e.g., shifting) is very predictive of performing well on all tasks involving that skill (e.g., category-switch) because the models do not include any additional source of 'noise' in the current model besides the default sources in PRIMs. This could be differences in strategy (e.g., using a very active strategy on the letter-memory but a very passive one on the two-back), effects of fatigue or motivation, or any other difference present in noisy human data.

In conclusion, both the analysis comparing high and low performers as well as the correlation analysis suggest that differences in EF skill proficiency can lead to large differences in performance (as shown by our models in both analyses), however in human data this effect is accompanied by several other factors influencing performance (especially shown by the lower correlations in the human data).

4. Discussion

4.1. The Skill-Based Approach

The skill-based approach is a promising method to increase generalizability in cognitive modelling. Although the cognitive architecture PRIMs is designed with the aim of enabling skill reuse, there are still limitations that prevented the creation of fully reusable skills (Hoekstra et al., 2022a). In the previous chapter, we identified these limitations and proposed initial modifications to the PRIMs architecture. In this chapter, we fledged out the proposals from the previous chapter and tested whether these modifications really improved the process of using the skill-based approach. This was done by re-attempting the modelling of the EF tasks described in Miyake et al. (2000) with the modifications included in PRIMs. Additionally, the models created in such a way could provide interesting insights in the underlying mechanisms of the modelled EFs and shed light on the question whether EFs are a learned or innate ability.

We identified three limitations to the implementation of the skill-based approach in PRIMs: (1) inflexible WM, (2) rigid goal selection and, (3) all-or-nothing condition checking. We introduced temporal bindings to PRIMs with the aim of improving the flexibility of WM. Secondly, we introduced the convention of using a specific skill for goal selection to improve the flexibility of goal selection. Finally, we introduced the use of spreading activation as an alternative to condition checking to ease the inflexibility associated with condition checking.

The three modifications greatly improved the effectiveness of the skill-based approach. This was largely accomplished by the increase in the level of skill reusability made possible by these modifications. Although reuse may seem like a dichotomous variable (i.e., a skill is reusable or not), there are different levels of reuse. The skills we created for the attentional blink (AB) models (Hoekstra et al., 2020; Hoekstra, Martens, & Taatgen, 2022b) possessed the basic level of reusability because they were used in more than one task. However, their reusability was limited since they cannot be reused in a different context outside of the AB without considerable changes to the procedural knowledge of this skill (i.e., the operators). The limited reusability in the AB models was largely caused by the necessity to include subtle task specific knowledge as part of the procedural knowledge of the skill (Hoekstra et al., 2022a). The modifications we made to the architecture and the modelling approach tempered this necessity and allowed the creation of functional and effective general skills without having task specific knowledge fundamentally entrenched in the reusable skill.

The use of spreading activation instead of conditions was the most important modification we made. Conditions were the prime culprits in importing task specific knowledge into a reusable skill. The need to specify the context in which an operator should fire as a fundamental part of an operator severely limits reusability because a different task almost guarantees a different context. Using flexible spreading activation relationships between operators and context maintains the high level of controlled model behaviour realized by conditions and combines it with the flexibility that comes with these relationships not being a fundamental part of the operator. Additionally, the mechanism with which these relationships can be learned are in line with current research on how the Basal Ganglia learn and accomplish action selection (Redgrave et al., 1999; Stewart et al., 2012; Stocco, Lebiere, & Anderson, 2010).

The added functionality of temporary bindings to the PRIMs architecture was an additional major benefit to skill reusability. Skills can now be designed in such a way that they can store and retrieve information from short-term memory without the need to specify where to store the item or from where to the retrieve the item. This prevents reuse limitations caused by differences in working memory slot availability between tasks. This especially impacted the reusability of the update skill. Additionally, bindings allow for a plausible and principled way in which task specific information can be integrated into the general procedural knowledge. These bindings also provide a more plausible mechanism with which these task specific pieces of information can be learned. It is plausible to assume that people create a collection of new bindings when reading (or listening to) task instructions. Furthermore, this mechanism provides interesting hypotheses about when and how the learning of a new task might be easier or harder. For instance, a task will be more difficult to learn if it involves the creation of many new bindings or when some bindings are only rarely used (and therefore may have decayed).

The new way of selecting which goal to select next also greatly improves skill reusability. Because of this method, skills do not need to include several operators that are only responsible for switching to the next goal and only need to include one PRIM or operator that switches to the select-next-skill skill when the goal is completed. The necessity of this modification largely depends on the complexity of the task in which the skill needs to be integrated. It is mainly useful for particular skills that have a variable endpoint or need to switch to several different goals. However, it also plays an important conceptual role in that it provides a more plausible learning mechanism. Employing a separate skill for goal selection models the gradual learning of the order of skills in a new task more accurately than a method that uses operators that are fundamental parts of the skill (and therefore are already learned).

The modifications we proposed and tested also introduced new challenges and questions. Firstly, it is not clear how the different methods to perform operator selection relate to one another. It is not clear whether the method we proposed here (using spreading activation relations) should completely replace conditions or whether they coexist. Furthermore, if they do coexist, it is not clear how to exactly combine the two methods or how the balance might develop over time (e.g., are S_{ji}s slowly replaced by conditions or the other way around, or not at all?). Additionally, the extent to which spreading activation relationships (S_{ji}s) are reused is not clear: are they stable or does their influence on behaviour decrease over time?

Secondly, the temporary bindings introduced a new but potentially strong influence on model behaviour. The bindings follow the same principles of the other chunks in DM (e.g., decay, noise, or retrieval failure rates). This can have an occasional big impact when, for example, a crucial binding fails to be retrieved at an important moment. In addition to that, it also has a consistent subtle impact on the reaction times of a model over the course of an experiment. Because the bindings slowly decay over time, reaction times on later trials will be slower than reaction times on earlier trials if this effect is not counteracted. This raises the questions whether bindings should be treated the same as other chunks in DM. Following the idea of rational analysis (Anderson & Schooler, 1991; Chater & Oaksford, 1999; Schooler & Anderson, 1997), it is likely that the development of the activation of a binding depends on the importance of this binding in the task. This is currently not taken into account. Alternatively, it could also be possible that frequently used bindings become part of the procedural knowledge of a particular skill when it is very highly learned. This skill would be optimized and very quick for one specific task but lose its generalizable nature. Mechanisms that could accomplish this would be very useful for further development of the skill-based approach and cognitive architectures in general.

Finally, the biggest downside of our method of selecting the next goal is that it might, occasionally, be too slow. After performing the same task for a certain amount of time, goal selection might be achieved automatically which makes the goal selection skill redundant. This high level of learning can be for a large part captured by production compilation (over time the select-next operators can be completed in one execution cycle), however, this still might be too slow in situations with many different small goals. This slower selection of goals becomes especially apparent in highly learned tasks. However, this downside is not completely a limitation of this method of goal selection but perhaps more a limitation of general skills. Similar to the problem discussed with the temporal bindings: when people become highly skilled in a certain task they may not use the general skill anymore but they might use a specific (not generalizable) skill instead.

To conclude, the modifications to the architecture and the modelling approach greatly improved the process of creating models with the skill-based approach. This is largely due to the big improvement to skill reusability made possible by these changes which resulted in the creation of skills that did not include task specific knowledge that prevented reuse in other contexts.

4.2. Executive functions

As mentioned before, the successful modifications resulted in a model of EF that can provide interesting insights in the mechanism underlying the EFs and shed light on the question whether EFs are part of the cognitive architecture or consist of a set of learned general skills. The now feasible skill-based approach is in a unique position to provide these insights because, in a sense, it functions as the cognitive modelling equivalent of the latent factor analysis. Because of task impurity, performance on several tasks need to be combined in order to extract the EF. Without going into the statistical details, this is accomplished by factor analysis by combining the performance on individual tasks and building the EFs from the interplay between the tasks. For example, a participants' Updating performance is determined by combining performance on all updating tasks and determining the common factor in these tasks (i.e., the Updating EF). The skill-based approach allows for a very comparable process. By constructing a single skill that will be used in all tasks requiring this skill, the nature of this skill and therefore the common element between these tasks, will become gradually apparent. For instance, creating the 'updating' skill for only one updating task will make the skill very specific for this single task. However, applying this same skill to a second and a third updating task will include common aspects that were not clear from the first task and exclude aspects that only apply to one (or two) of the tasks, resulting in only the common updating aspect between all tasks.

4.2.1. Mechanisms of executive functions

Much is known about the importance of executive functioning in daily life. A strong ability to perform the three core EFs, shifting, updating, and inhibition has been found to predict academic success (Gathercole, Pickering, Knight, & Stegmann, 2004), job performance (Bailey, 2007) and even physical health (Riggs et al., 2010). Additionally, low executive functioning has been shown to play in role cognitive impairments and psychological disorders such as addiction (Baler & Volkow, 2006), depression (Tavares et al., 2007) and schizophrenia (Barch, 2005). However, an explanation of what underlies this big impact of executive functioning is often lacking. This is because the focus of most research on EFs is on uncovering the role EFs play in general cognition, but not on the cognitive mechanisms underlying these EFs. This narrow view leads to a limited understanding of why low executive functioning can lead to impaired performance or psychological disorders (Duijkers et al., 2016) or how to improve it (Shipstead et al., 2012). The model (consisting of eight smaller models) we created of all three core EFs using a unitary mechanism for each EF can suggest the basic mechanisms underlying the core EFs.

As was mentioned in the introduction, according to the skill-based approach a core EF consists of two parts. The first part is the procedural part, this includes the operators and skills required to accomplish the function of the core EF. The second part is the automatic part, this includes the architectural mechanisms that the 'active' part acts upon but cannot be directly controlled by the person (or model). For example, for the updating skill, the procedural part are the operators that change the values associated with the variables and the automatic part are the bindings (DM chunks) that are created by those operators. The automatic part represents the DM characteristics (e.g., decay or noise, aspects that are outside of conscious control). These aspects depend on each other but influence the update success independently. For example, someone might have very efficient updating operators but also a very high decay rate. This person would perform quite poorly on an updating task; however, this is only due to the poorly functioning automatic aspect of Updating.

Using the skill-based approach, capable of extracting the processing steps all three shifting tasks have in common, we concluded that a successful shift contains two basic cognitive processing steps (at least in the tasks we modelled). Firstly, a new task set needs to be retrieved from declarative memory which, secondly, needs to be stored in such a way that it can be retrieved quickly. This might seem like a redundant way of processing since it includes two memory retrievals while it could also be accomplished with a single retrieval. However, the storage of the task set in an easier to access place is crucial to support fast and accurate processing (similar reasons why a computer needs working memory). These general processing steps were captured in our model by the shifting skill which contained two operators. The first operator of this skill retrieves a task set from declarative memory and the second operator updates a certain binding based on the retrieved task set. The binding that is updated by the skill is a crucial binding which either influences which operator subsequently fires or how subsequent operators will function. These two processing steps are the procedural part of the shifting ability. The automatic part of the shifting EF depends on the effectiveness of short-term memory, both in the ability to remember multiple task sets as in the ability to maintain the current relevant task set. We did not manipulate the automatic part of Shifting in our model (all models had the same DM parameters), however in people, differences in the automatic part of shifting might be a large contributor to individual variation. To conclude, the cognitive mechanism underlying Shifting consists of procedural knowledge that retrieves task sets from declarative memory and stores them in an easy to access location. Additionally, these easily accessible task sets require a healthy short-term memory system to survive until they are retrieved.

The basic cognitive processing steps we identified as underlying the three updating tasks are as follows. Firstly, identifying the update-target. For example, when updating a count to three, three is the update-target. Secondly, selecting which already existing variable should be updated. Thirdly, changing the value associated with the to-be-updated variable from the previous value to the update-target. For example, in the keep-track task, the first step is identifying that the newly presented item should be remembered (if it belongs to one of the target categories), the second step is determining which variable to update (e.g., the third category if the item belongs to the third category) and, finally, actually updating the value associated with this category to the new value (e.g., the third category is now iron). This process is the same for updating that requires shifting but it is repeated as many times as necessary depending on the number of items that need to be shifted. For example, on the fifth trial of a letter-memory task, the newly presented letter should go to the spot currently associated with the previous newest letter and the previous newest letter should go to the spot previously associated with the previous second-oldest letter and the previous second-oldest letter is now the new oldest letter. This is simply a repetition of identifying the update-target, identifying the to-be updated variable and performing the update three times. These three processing steps are the basic procedural part of Updating. The automatic part of updating, similar to the automatic part of Shifting relies heavily on the effectiveness of short-term memory since the most recent version of every variable-value pair needs to remain more active than any of the previous versions. The success of this mainly depends on the noise levels in DM, high noise makes it more likely that an older version of a variable-value pair will be more active by chance than the newest version. Interestingly, high levels of decay might actually improve performance on an updating task since older versions might dip below the retrieval threshold sooner and noise (assuming it does not scale with decay) impacts the activation levels less strongly. Because updating often is the main goal of updating tasks, people might adopt active strategies that reduce the reliance on short-term memory. These strategies could include rehearsal or visualization of the to-be remembered items. Adoption of such strategies might have a strong impact on between-subject variation or even between-task variation (within one subject). Because of the importance of strategy, the basic procedural part of Updating that is measured in EF tasks might often include strategy effects. However, we did not include this in our model because it would not fundamentally change the cognitive mechanism underlying updating but merely repeated updating (in the case of rehearsal) or updating in a different modality (in the case of visualization, although this might be very different from the updating we considered here). To conclude, the cognitive mechanism underlying Updating consists of procedural knowledge that identifies the update-target, determines what to update and establishes an association between the update-target and the determined to-be updated variable. Additionally, these associations require a healthy short-term memory to be retrieved later and strategy might play a considerable role in compensating for the unpredictability of short-term memory.

The basic processing steps that the two inhibition tasks share are as follows. Firstly, a memory retrieval has to be done to retrieve the correct 'inhibition instructions', these instructions are variables that represent the particular inhibition that is required for the current task (e.g., *inhibit-reading* for the Stroop task). Secondly, these instructions need to be placed in the workspace so they can influence further processing. In our models, the instructions were placed in the first slot of the imaginal buffer, however it is possible to place these instructions anywhere in the workspace. A suitable alternative location for this would be the goal buffer since the instructions play a comparable role as a goal. Interestingly, something representing those 'instructions' could also be placed in the input buffer (i.e., on the screen, a sound, or an object), although this will likely not be considered Inhibition since it does not involve active effort (besides the initial placement). In the end, the core characteristic of Inhibition is that *something* that influences operator selection into the intended direction is actively placed in the workspace. The procedural part of Inhibition is accomplished by the retrieving and placing of 'inhibition instructions' in the workspace. The automatic part depends on declarative memory (to store the 'instructions') and on the effect that the 'instructions' have on subsequent operator selection. The 'instructions' need to be able to overcome the influence of the unintended direction, for example the saccadic reflex to a new stimulus in the antisaccade task or the habit of reading a word in the Stroop task. This requires very fast acting and powerful spreading activation. Furthermore, the stochastic nature of operator selection requires this spreading activation to be considerable larger than the reflexive behaviour in order to compensate for potential noise. An interesting question is whether individual variation in Inhibition ability is caused by differences in ability to set high enough spreading activation relationships (e.g., some people can only produce S_{ji}s of maximum 1 while other can produce S_{ji}s of 3 or more) or if this variation is caused by strategy (i.e., the choice to prepare or not), or, alternatively, a difference in how easy spreading activation relationships can be learned (e.g., some people can learn that they need to inhibit a particular operator after only two unsuccessful trials while others might need twenty). To conclude, the cognitive mechanism for Inhibition requires an active step to retrieve and place 'inhibition instructions' in the workspace. Additionally, these instructions need to have a strong enough influence to overcome the pre-potent behaviour.

4.2.2. Can executive functions be improved?

There has been a lot of research on whether executive functioning can be improved or whether it is a stable characteristic of any individual. Many studies have shown a significant effect of EF training on performance on other EF tasks (Bergman Nutley et al., 2011; Diamond & Lee, 2011; Karbach & Kray, 2009; Klingberg, 2010). Especially, Updating seems to be sensitive to training effects (Gray et al., 2012; Kamijo et al., 2011). However, the ecological validity of these studies has been questioned and they often show little to no transfer beyond the initial training task (Morrison & Chein, 2011; Shipstead et al., 2012). Overall, these studies indicate that EF training has an effect on performance, however it is highly questionable whether these training programs truly improve the core EF (Blair, 2017).

Our model could potentially provide a basic answer to the paradoxical results produced by the research on EF training. Our model assumes that the core EFs (shifting, updating, and inhibition) depend on two related but independent aspects, a procedural part and an automatic part as described above. Furthermore, it assumes that the automatic aspects are stable within and between participants (i.e., people do not differ in their module efficiency and this efficiency is the same for every task). Therefore, it predicts that all the individual differences can be ascribed to differences in the procedural part of an EF. We tested this prediction by running several models with differing levels of procedural EF ability to check whether such models would display comparable individual differences (between the different models) as is found between human participants.

The models, only differing in how well the basic EF skills function, showed very comparable performance to the human participants in our data set. The top 50% highest performing models performed very similarly to the top 50% of human performers and the bottom 50% of models and humans also showed highly comparable performance patterns. This suggests that the individual variance present in the data can be explained by merely varying the mastery levels of the procedural knowledge required for the eight EF tasks. This result supports the idea that the procedural part of the core EFs can explain the entire range of individual differences present in these tasks. This is a very different view of executive functioning than the one that is currently common in the EF literature. Instead of conceptualizing an EF as a hard to pin-down abstract construct, it views it as tangible and concrete blocks of procedural knowledge.

Furthermore, our model explains the inconsistent results produced by the studies investigating the effects of EF training. Our model suggests that success of training depends on two things: (1) the tasks need to rely on the same procedural knowledge and (2) the participants have relatively underdeveloped basic EF skills. Firstly, the tasks need to rely on the same procedural knowledge because training is only effective when the procedural knowledge required for the training task is the same or highly similar to the test task. Secondly, the skill level of the individual participant is crucial because the training only works for participants that have relatively underdeveloped basic EF skills. Only these participants still have room to improve because the basic skills can receive additional training. If a participant has fully developed basic EF skills (i.e., they are fully compilated), training is not possible even though the tasks might have a lot in common. Training beyond the compilation of the basic skills would require the learning of new skills that only apply to the training task, but not beyond that.

However, our model is not complete. The data and the model showed a very similar pattern of correlations between the tasks with many significant correlations between tasks measuring the same EF and few correlations between tasks belonging to different EFs. However, the correlations found in the model were much higher compared to the correlations present in the human data. This suggests that other sources of individual variation also strongly influence the individual differences found in the EF tasks. This includes differences in strategy, motivation or fatigue.

To conclude, differences in how well a basic EF skill is learned can provide an explanation for the individual differences found in the EF tasks. Furthermore, the differentiation between the procedural part of an EF and the automatic part of an EF is very helpful in explaining why EF training is sometimes effective and sometimes not effective. This is because only the procedural part of an EF can be trained (if there is still room for training), but not the automatic part.

Appendix

Detailed description of model building procedure

The basic practical process of building the models was as follows. We started by fully building one model from start to finish (e.g., the letter-memory model) and perform basic model fitting (compare accuracy and RT to the human data). Subsequently, we built the next model from the same core EF (e.g., the keep-track model) and started by creating the specific skills required to accomplish this task. Then, we added the already built basic skill to the model. This often required some small changes to the basic skill(s) in order to successfully integrate into the new model. These changes were usually due to the original skill being too specific for the first task which made it incomplete for the second task or prevented it from functioning properly in the second task. An example of the first reason is that the keep-track model needed to update four items while the letter-memory model only updated three which required adding a new operator. An example of the second reason is that the updating operators were only used at the start of the task in the letter-memory model while they were used throughout the whole task in the keep-track model, thus requiring the conditions to be modified in order to reflect this. After the basic skills were integrated into the new model, we performed a basic model fit procedure again on both tasks. This was done on both tasks because the changes made to the basic skill could have resulted in changed model behaviour on the first task. During model building we kept all model parameters equal in order to get a clear idea of how the basic skills functioned. Because of this, during model running almost all parameters could be kept equal as well. Only the latency factor was different in the Stroop model models because the bindings were slowing the model down too strongly, otherwise all parameters were kept constant.

The changes to the architecture we proposed were instrumental in this process. These were: temporary bindings, a specific goal selection skill, and limited condition checking combined with spreading activation. The specific goal selection skill is the smallest modification and is already explained with a specific example in the introduction, therefore we will only discuss the specific use of the bindings and the condition checking here.

The temporary bindings were used to store information that was relevant at certain points in the task but did not need to be present in the workspace at all times. Specifically, they were used for two important roles in the models: (1) storage of task-specific information and (2) storage of short-term relevant information. Both of these roles can be illustrated by looking at the first operator of the final update skill depicted below.

The first role is crucial for adding task-specific information to a general skill. Infusing the task-specific information into a general skill is one of the biggest challenges when creating reusable skills. The bindings allowed for a flexible but concrete way of doing this because the task-specific information did not need to be included in the procedural knowledge of a skill. Instead, the task-specific information was provided by the bindings. This role can be seen in the final PRIM of the operators of the update skill depicted below. These PRIMs place the next goal in the goal buffer which is stored in the binding **next-skill-update*. Because this is stored in a binding it can be different in every task and, potentially, within a task (which is how it was used in the shift skill). The second role was especially fundamental to the updating tasks since it allowed for the flexible storage of the to-be remembered items in these three tasks. The bindings allowed for storage of short-term relevant information of to-be remembered values combined with their 'identity'. Because of this, the model did not need to know exactly where an item was stored but only which item (i.e., the identity) to retrieve. This role can be seen in the first PRIM of the operators, these PRIMs update the value of the binding **first-item* based on the value stored in the binding **current-target*. In addition to a more flexible use of WM across task, this method also frees up the imaginal buffer to only be used for immediately relevant information (such as providing the information on which this operator is selected) and it prevents the difficult calibration process when this operator is reused in a different context. For example, if this operator would place the current-target in WM2 instead of a binding, it would require WM2 to be available.

```
define goal update {

      operator update-first {
      WM1 <> nil
      ==>
      *current-target -> *first-item
```

```
        nil -> WM1
        *next-skill-update -> G1
        }

        operator update-second {
        WM1 <> nil
        ==>
        *current-target -> *second-item
        nil -> WM1
        *next-skill-update -> G1
        }

        operator update-third {
        WM1 <> nil
        ==>
        *current-target -> *third-item
        nil -> WM1
        *next-skill-update -> G1
        }

        operator update-fourth {
        WM1 <> nil
        ==>
        *current-target -> *fourth-item
        nil -> WM1
        *next-skill-update -> G1
        }

        operator shift-values-three {
        WM1 <> nil
        ==>
        *second-item -> *first-item
        *third-item -> *second-item
        *current-target -> *third-item
        *next-skill-update -> G1
        }

        operator shift-values-two {
        WM1 <> nil
        ==>
        *second-item -> *first-item
        *current-target -> *second-item
        *next-skill-update -> G1
        }

}
```

Secondly, the modifications proposed to condition checking were used to allow for more flexible use of the operators in a skill. Instead of specifying specific conditions for every operator, the conditions were kept to a minimum and task-specific conditions were replaced by setting spreading activation relationships (*S_{ji}'s*). This can be seen by looking at the conditions of the operators in the final update skill depicted above. The only condition used in all operators is checking whether WM1 is not empty. In this case, the task-specific information about which operator to select was indirectly provided by WM1 because we set different *S_{ji}'s* between contents of WM1 and the operators. As a side note, this is only an illustration of how conditions can be replaced by setting *S_{ji}'s,* it does not always have to be done by using the imaginal buffer. For example, in the keep-track task, which updating operator to select was based on which category should be updated. This was done by setting the following *S_{ji}'s*.

```
set-sji("update-first","imaginal","slot1","one",3)
set-sji("update-second","imaginal","slot1","two",3)
set-sji("update-third","imaginal","slot1","three",3)
set-sji("update-fourth","imaginal","slot1","four",3)
```

Running these functions set positive *S_{ji}'s* of three between a certain value in WM1 and an operator (e.g., when *one* is in WM1, spread three activation to the operator *update-first,* or when *two* is in WM1, spread three activation to the operator *update-second)*. A skill specific for the keep-track task assigns a number to a category based on the position of this category on the screen (the most left receives *one* and the most right receives *four)*. Throughout all models, task-specific conditions were replaced, when necessary, by setting spreading activation relationships in this manner. In this example, using spreading activation instead of conditions was useful because it removed the need to add the condition that a certain value should be present in WM1. This condition would be useful for this particular task; however, it would not be (necessarily) useful in other tasks.

Detailed descriptions of models

Shifting models

The category-switch model (see Figure 4a) consists of two basic skills (*shift* and *respond*) and two skills specific for this model (*select-next-skill-category-switch* and *retrieve characteristic*). These skills were combined in the following way. The model starts a trial with the *select-next-skill-category-switch* skill. This skill determines whether the current trial is a switch trial or not by comparing the current cue ('size' or 'alive') to the task that the model did on the previous trial (also either 'size' or 'alive'). If they are not the same the model is doing a switch-trial, which means that it will change to the *shift* skill which will change the value associated with **characteristic* to either 'size' or 'alive' depending on the trial. When the shift is completed, the model returns to the *select-next-skill-category-switch* skill and waits for the presentation of the word. If they are the same, the model does not need to shift and it simply waits until the word is presented. After presentation of the word, the *retrieve characteristics* skill determines the answer of whether the presented word is alive or bigger than a football (depending on the trial) by performing a memory retrieval based on the value in **characteristic*. Finally, the *respond* skill carries out the final response which is either pressing 'j' for yes or 'f' for no. In order to integrate the reusable *shift* and *respond* skill into the model the bindings were instantiated as follows. For the *shift* skill, **task* was instantiated as *category-retrieval* and **next-skill-shift* was instantiated as *select-next-skill-category-switch*. For the *respond* skill, **key-right* was instantiated as *J, *key-left* as *F,* and **next-skill-respond* as *select-next-skill-category-switch*.

The colour-letter model (Figure 4b) also consists of two basic skills (*shift* and *respond*) and two skills specific to this model (*select-next-skill-colour-letter* and *attend-colour-or-type*). Although the *select-next-skill-colour-letter* has the same function as the *select-next-skill-category-switch*, it is nevertheless considered as a distinct skill because the operators are not the same. The colour-letter model is combined in the following way. The colour-letter task starts with a blank screen for 150 ms, during this the model starts with the *select-next-skill-colour-letter* skill which determines whether the current trial is a switch trial or not. It does this by comparing the location of where the next letter will be to where the previous letter was. If they are not the same, the model will go to the *shift* skill. This skill will change the value associated with **characteristic* to either 'letter-type' or 'colour' depending on the trial. After completing the shift, the model will return to the *select-next-skill-colour-letter* and waits for the presentation of the letter. If they are the same, the model simply waits for the letter presentation. After the letter is presented, the model will go to the *attend-colour-or-type* skill which either attends the letter itself or the colour in which the letter is presented. The letter-type (vowel or consonant) is determined through a memory retrieval (which takes two operators), whereas the colour is determined simply by attending the colour (which only takes one operator). Finally, after either the letter-type or colour is determined, the response skill gives the appropriate response ('f' for vowel or red or 'j' for consonant or blue). In order to integrate the reusable *shift* and *respond* skill into the model the bindings were instantiated as follows. For the *shift* skill, **task* was instantiated as *colour-letter* and **next-skill-shift* was instantiated as *select-next-skill-colour-letter*. For the *respond* skill, **key-right* was instantiated as *J, *key-left* as *F,* and **next-skill-respond* as *select-next-skill-colour-letter*.

The colour-shape model (Figure 4c) also consists of two basic skills (*shift* and *respond*) and two skills specific to this model (*select-next-skill-colour-shape* and *attend-colour-or-shape*). These skills are combined for the colour-shape task in the following way. The model starts in the *select-next-skill-colour-shape* skill which determines whether a trial is a switch trial or not by comparing the new cue to the task done on the previous trial. If they are not the same, the model goes to the *shift* skill which changes the value associated with **characteristic* to either 'colour' or 'shape' depending on the trial. When the shift is completed, the model will return to the *select-next-skill-colour-shape* and waits for the shape to be presented. If the previous task and cue are the same, the model does not need to switch and simply waits for the shape to be presented. After presentation of the shape, the *attend-colour-or-shape* skill determines the colour or the identity of the shape. This can be done by simply attending the stimulus without the need of a memory retrieval. Finally, the *respond* skill gives the final response (either 'f' for red or triangle or 'j' for blue or square). In order to integrate the reusable *shift* and *respond* skill into the model the bindings were instantiated as follows. For the *shift* skill, **task* was instantiated as *colour-shape* and **next-skill-shift* was instantiated as *select-next-skill-colour-shape*. For the *respond* skill, **key-*

right was instantiated as *J, *key-left* as *F,* and **next-skill-respond* as *select-next-skill-colour-shape*.

Updating models

The keep-track model (see Figure 5a) consisted of three basic skills (*read*, *update*, and *respond*) in addition to one keep-track specific skill (*category search*). These skills were combined in the following way. At the start of a trial, the first item was read by the read skill. Subsequently, the *category search* skill determined to which category this item belonged through means of a memory retrieval. Afterwards, the *update* skill updated the binding associated with this category (e.g., the **third-item)*. This process repeated until the end of the trial. At the end of a trial, the final responses were given by the *respond* skill which was done by retrieving the value associated with each of the bindings and typing them on a keyboard. In order to successfully perform the keep-track task, the first four operators of the *update* skill were needed. In order to integrate the reusable *read, update* and *respond* skill into the model the bindings were instantiated as follows. For the *read* skill, **report-instructions* was instantiated as *report, *respond-skill* as *respond* and **next-skill-read* as *category-search*. For the *update* skill, **next-skill-update* was instantiated as *read*. For the *respond* skill, **action* was instantiated as *press, *read-skill* as *read,* and **report-instructions* as *report*. Different bindings needed to be created for the updating tasks because different operators from the *respond* skill were used. The previously mentioned bindings were also created; however, they did not influence model behaviour.

The letter-memory model (Figure 5b) also consisted of three basic skills (*read*, *update*, and *respond*) in addition to one letter-memory specific skill (*count*). These skills were combined for the letter-memory task in the following way. At the start of a trial, the *read* skill read the first letter. After this, the *count* skill placed a value in WM1 depending on how many stimuli had already been presented (e.g., a ‘1’ for the first letter and a ‘more-than-three’ for the fourth letter and up). Subsequently, the *update* skill performed the correct update based on the value stored in WM1, it fired one of the *update-x* operators if this value was lower than three and it fired the *shift-values-three* if this value was ‘more-than-three’. This process repeated until the final letter was presented. Finally, the *respond* skill took care of the responses at the end of the trial in the same way as in the keep-track model. In order to successfully perform the letter-memory task, the model required three single-value update operators and the *shift-value-three* operator. In order to integrate the reusable *read, update* and *respond* skill into the model the bindings were instantiated as follows. For the *read* skill, **report-instructions* was instantiated as *report, *respond-skill* as *respond* and **next-skill-read* as *count*. For the *update* skill, **next-skill-update* was instantiated as *read*. For the *respond* skill, **action* was instantiated as *press, *read-skill* as *read,* and **report-instructions* as *report.*

The spatial two-back model (Figure 5c) consisted of only two basic skills (*update* and *respond*) and one two-back-specific skill (*compare*). These skills were combined for the spatial two-back task in the following way. When a square was presented, the *compare* skill compared the location of the current square to the location of the two-back square. After this, the *respond* skill gave the response associated with the outcome of the *compare* skill (pressing 'j' for yes and 'f' for no). Finally, the *update* skill performed the update, either with the *update-first* or *update-second* operator for the first two locations of the trial or with the *shift-values-two* operator for the remainder of the trial. This process repeated until the end of the trial. In order to successfully perform the two-back task, the model required the first two single-value update operators and the *shift-values-two* operator. In order to integrate the reusable *update* and *respond* skill into the model the bindings were instantiated as follows. For the *update* skill, **next-skill-update* was instantiated as *read*. For the *respond* skill, **action* was instantiated as *press, *read-skill* as *read,* and **report-instructions* as *report.*

Inhibition models

The antisaccade model consisted of two basic skills (*prepare-to-inhibit* and *respond*) and two specific antisaccade skills (*select-next-skill-antisaccade* and *determine target*). These skills were combined for the antisaccade task in the following way. A trial of the antisaccade task started with a fixation cross, during this time the *prepare-to-inhibit* skill prepared for the upcoming stimuli by placing *inhibit-saccade* in WM1 which spread activation to associated operators in such a way that it inhibited the reflexive response of looking at a new stimulus (also depicted above). After the *prepare-to-inhibit* skill, the model returned to the *select-next-skill-*

antisaccade skill. This skill selected the next goal which was to determine the direction of the upcoming arrow and switched to the *determine target* skill. This skill waited for the arrival of the distractor and the target. On a successful antisaccade trial this skill ignored the first stimulus (the distractor) and only attended the target. On an unsuccessful trial, this skill did attend to the first stimulus and was unable to respond fast enough to detect the direction of the target arrow. After attending either the distractor or the target, the model returned to the *select-next-skill-antisaccade* skill which then moved on to the *respond* skill. Finally, the *respond* skill gave the final response by pressing the appropriate arrow key on the keyboard (left, up, or right). In order to integrate the reusable *prepare-to-inhibit* and *respond* skill into the model the bindings were instantiated as follows. For the *prepare-to-inhibit* skill, **prepare-goal* was instantiated as *prepare-to-inhibit, *fact-type* as *prep-fact*, **task* as *antisaccade*, and **next-skill-prepare* as *select-next-skill-antisaccade*. For the *respond* skill, **action* was instantiated as *press, *read-skill* as *read,* and **report-instructions* as *report*. The binding **read-skill* is still defined in this model because the operator using it would not be able to fire if it did not exist. However, this does not influence model behaviour because this operator fires at the end of a trial.

The Stroop model consisted of three basic skills (*read*, *prepare-to-inhibit*, and *respond)* and three Stroop specific skills (*select-next-skill-stroop*, *determine colour*, and *choose hand*). These skills were combined for the Stroop task in the following way. A trial of the Stroop also started with a fixation cross, during this time the *prepare-to-inhibit* skill prepared for the upcoming trial by placing *inhibit-reading* in WM1 which spread activation to operators in such a way that it inhibited the prepotent response of reading a word. After this, the model returned to the *select-next-skill-stroop* skill which waited for the word to be presented and selected the next skill. Depending on the activation of the operators in this skill, this could be either of two skills: the *read* skill or the *determine colour* skill. If the model selected the *read* skill, it would read the word which would take a certain amount of time before returning to the *select-next-skill-stroop*. Then, if it is a conflict trial (e.g., the word blue printed in red) it would need to select the *determine colour* skill (it could also skip this step which would result in an incorrect response). If the model had selected the *determine colour* skill it would go straight to this part or, alternatively, if it was a congruent trial it could skip the *determine colour* skill because it already had the correct answer. The *determine colour* skill, then, attended the colour in which the word was printed and placed it in the binding **current-target* so that it could be remembered for later use and the model returned to *select-next-skill-stroop*. This skill would select the *choose hand* skill which determines the correct hand to respond with by looking at the response options that flanked the presented word. Then, the model would return to *select-next-skill-stroop* one more time which selected the *respond* skill. Finally, the *respond* skill would press the key associated with the correct hand ('f' for left and 'j' for right). In order to integrate the reusable *read, prepare-to-inhibit* and *respond* skill into the model the bindings were instantiated as follows. For the *read* skill, **report-instructions* was instantiated as *report, *respond-skill* as *respond* and **next-skill-read* as *select-next-skill-stroop*. For the *prepare-to-inhibit* skill, **prepare-goal* was instantiated as *prepare-to-inhibit, *fact-type* as *prep-fact*, **task* as *antisaccade*, and **next-skill-prepare* as *select-next-skill-stroop*. For the *respond* skill, **key-right* was instantiated as *J, *key-left* as *F,* and **next-skill-respond* as *select-next-skill-stroop*.

Model training

Besides testing the effectiveness of the proposed changes to the architecture, a second goal of this study was to investigate how differences in basic-skill proficiency can lead to individual differences in performance on the EF tasks. To achieve this, we ran models that differed with respect to the amount of training they received, which resulted in differences in the amount of production compilation that the basic skills underwent.

Production compilation is a fundamental aspect of PRIMs and it explains why people speed up over the course of performing a task (Taatgen & Lee, 2003). Production compilation allows combinations of PRIMs that are part of the same operator and therefore often executed together to combine into one. When the process of production compilation is completed every operator of a model can be fully executed in one production cycle regardless of how many PRIMs are in this operator. In short, production compilation happens automatically every time a model runs and therefore the more often a model has run the quicker it is in performing its actions.

Because of the importance of production compilation, the usual model running procedure of PRIMs models consists of a training phase and an experimental phase (similar to how psychological experiments include a training block before the experimental block). Typically, the training phase is used to allow the model to complete production compilation, however for our current goal we ran two different versions of the training phase for every basic skill. One resulting in full production compilation (high proficiency) and one resulting in incomplete production compilation (low proficiency). The high proficiency version included as many practice trials necessary to achieve full compilation (e.g., 50), while the low proficiency version include only half that number (e.g., 25).

Subsequently, we tested the effect of the different levels of basic-skill proficiency on model performance. The model runs were structured in such a way that every run simulated a single participant. A participant did all eight tasks with the same level of basic-skill proficiency; however, the different skills could have different proficiencies. Each combination of basic-skill proficiencies was included fifty times, resulting in a total of 400 simulated participants. This means that every proficiency level was run 200 times total evenly combined with both levels of the two other skills (e.g., the combination low shift, high update, high inhibition was run fifty times, the combination low shift, low update, high inhibition was run fifty times et cetera until all combinations were included). After every 'participant' the models were reset. The tasks in the experimental phase were always done in the same order, however this did not cause any issues because production compilation was turned off during the experimental phase.

Individual differences model fit

We determined the fits of all models the same way as the individual model fits by creating a separate linear regression model for the model data and the human data and comparing the resulting model estimates. For this analysis we ran a simple linear model with one dependent variable (the measure plotted in Figure 10) and one categorical independent variable Group (i.e., high or low performers) with high performers as the reference level. There were no random slopes or intercepts in the models. Table 5 shows the estimates produced by these models. The high Group was the reference level; therefore, the average performance of this group is shown as the intercept and the average performance of the low Group is shown as the coefficient (called Low). Because the low Group is a coefficient what is shown in the table is the difference between the low and high Group (i.e., in order to arrive at the average performance of the low group, the coefficients should be added to the intercept).

The fit of the shifting tasks was determined by looking at the average switch costs of the high and low performers. These were calculated by taking the average reaction time on a switch trial and subtracting the average reaction time on a non-switch trial. Visual inspection suggests that the fits of the shifting models is good. The high and low halves of the model perform very similarly to the high and low halves of the human data. The low performers in all tasks in both the model and the participants have higher switch costs than the high performers and the magnitude of this effect seems very comparable. This is corroborated by the linear models. In the category-switch task, the high performers had similar low average switch costs in the model and the data with 154 ms and 180 ms respectively. Additionally, the low performers had significantly higher switch costs in both the model and the data with a very similar increase of 287 ms and 254 ms respectively. The same was true for the colour-letter task, with the high performers having very similar low average switch costs in the model and the data with 211 ms and 242 ms respectively. Additionally, the low performers had significantly higher switch costs in the model and the data with a very similar increase of 286 ms and 318 ms respectively. Finally, the colour-shape task showed a similar pattern. The high performers have low average switch costs in the model and the data with 213 ms and 144 ms respectively. Additionally, the low performers had significantly higher average switch costs in both the model and the data with an increase of 294 ms and 352 ms respectively.

The fit of the updating tasks was determined by looking at the average accuracy of the high and low performers. Visual inspection suggests that the model fits are good. The models and the participants performed at very similar levels in the high and low groups in all tasks and the difference between high and low was comparable in the model and the data. This was supported by the linear models. For the keep-track task, the high performers reached a similar high average accuracy in the model and the data of 94% and 91% respectively. Additionally, the low performing group achieved a significantly lower level of performance in both the model and the data with a decrease of 11 percent point and 18 percent point respectively. A similar pattern was visible for the letter-memory task. The high performers reached a high level of accuracy in both the model and the data with 99% and 93% respectively. Additionally, the low performers scored considerably and significantly lower in the model and the data with a decrease of 44 percent point and 34 percent point respectively. Finally, the spatial two-back task showed the same pattern. The high performers performed very well in both the model and the data with an average accuracy of 77% and 82% respectively. Additionally, the low performers performed significantly worse in the model and the data with a decrease in performance of 26 percent point and 47 percent point. The model did not capture the magnitude of the Group effect very well though. This is because the model always made a guess on every trial, while the low-performing participants may not have given a response at all when they lost track of the stimuli. Because of this, the model could not perform under chance level (which was 50%) whereas the participants did.

The fit of the inhibition tasks was determined by looking at the average accuracy for the antisaccade task and at the average stroop interference for the Stroop task. The stroop interference was calculated per participant by taking the average response time on conflict trials and subtracting the average response time on congruent trials. The visual inspection suggests a good fit for the models. The high and low performers have similar average performance and follow the same pattern in the antisaccade task. The interference costs in the Stroop task are also very similar, although the model seems to slightly underestimate the stroop interference in the low performers. This was supported by the linear models. For the antisaccade task, the high performers reached a similarly high level of accuracy in both the model and the data with 87% and 90% respectively. Additionally, the low performers performed significantly worse in the model and the data with a decrease of 11 percent point and 17 percent point respectively. For the Stroop task, the high performers experienced similarly low levels of stroop interference in the model and the data with 57 ms and 78 ms respectively. Additionally, the low performers experienced significantly higher levels of stroop interference in both the model and the data, although the average interference of 124 ms in the model was lower than the average interference of 199 ms in the data.

Table A1. Regression estimates for the high and low performers in all tasks. The intercept shows the average performance of the high performers. The coefficient indicated with Low shows the adjustment to the intercept for the low performers.

	Model				*Human data*			
	Estimate	SE	t value	p value	Estimate	SE	t value	p value
Shifting								
Category-switch								
(Intercept)	154.4	7.1	21.7	> .001*	179.9	16.1	11.2	> .001*
Low	286.7	10.1	28.5	> .001*	253.5	22.6	11.2	> .001*
Color-letter								
(Intercept)	210.8	9.1	23.3	> .001*	241.8	28.3	8.5	> .001*
Low	286.1	12.8	22.3	> .001*	318.3	39.8	9	> .001*
Color-shape								
(Intercept)	213.3	7.2	29.7	> .001*	143.9	26.1	5.5	> .001*
Low	294.3	10.1	29	> .001*	351.8	36.6	9.6	> .001*
Updating								
Keep-track								
(Intercept)	94.1	0.4	263.6	> .001*	91.3	1.2	79.2	> .001*
Low	-10.9	0.5	-22.3	> .001*	-17.9	1.6	-11.4	> .001*
Letter-Memory								
(Intercept)	98.8	0.4	269.8	> .001*	93.1	4.2	22	> .001*
Low	-44.1	0.5	-85.3	> .001*	-33.8	5.4	-6.2	> .001*
Spatial Two-back								
(Intercept)	77.4	0.08	975	> .001*	81.7	2.7	29.8	> .001*
Low	-26.1	0.1	-233	> .001*	-46.8	3.9	-12	> .001*
Inhibition								
Antisaccade								
(Intercept)	86.5	0.3	291	> .001*	89.7	2	44.5	> .001*
Low	-11.1	0.4	-26.7	> .001*	-17	2.8	-6.1	> .001*
Stroop								
(Intercept)	56.8	1.8	31	> .001*	77.7	7.9	9.7	> .001*
Low	67.6	2.6	26.1	> .001*	131.2	11.3	11.6	> .001*

6
General Discussion

In this dissertation we attempted to develop a modelling approach that mirrors the human approach to novel (simple) task learning. When people are confronted with a new task, they do not figure out from scratch how to accomplish this task but instead rely on previously learned knowledge. This is the approach we intended to translate to a practical and useable modelling approach. Instead of creating a brand-new model for every new task as is currently standard practice in cognitive modelling, the modelling approach we set out to develop would allow modelers to reuse blocks of procedural knowledge (i.e., skills) that were created for other models and use them for the new model. The approach we developed to this end is called the skill-based approach.

In this discussion, we will discuss the final resulting modelling approach and how it can contribute to cognitive science, and the general challenges in translating the approach to other cognitive architectures.

1. Main conclusions of the dissertation

The work done in chapter 2 in proposing, developing and testing the initial version of the skill-based approach pointed to three main conclusions. Firstly, we successfully created a model of the attentional blink from skills that were taken from other models. This shows the basic feasibility of the skill-based approach and supports the idea that performing a simple task (such as the attentional blink task) is done by simply selecting the appropriate skills. Secondly, it suggested that this selection does not always select the most optimal skill for the task. Although the task can be completed in most situations (all Lags except for Lag 2 and 3) with the separate consolidation skill, it does not work for Lag 2 and Lag 3. Finally, chapter 2 showed the basic feasibility of the skill-based approach in producing plausible and novel models.

Chapter 3 further developed and tested the attentional blink model proposed in chapter 2. This chapter provided two main conclusions. Firstly, a large cluster of participants did not experience the attentional blink in our second experiment, this suggests that the attentional blink is not as fundamental to human performance as is often assumed. Secondly, it showed that models created using the skill-based approach can provide interesting new insights in an intensively studied experimental paradigm since our model suggests that the attentional blink is a simple consequence of the normal functioning of the cognitive system (i.e., it is the result of a general consolidation skill without any extra assumptions).

Chapter 4 discussed three limitations to the initial design of the skill-based approach and the PRIMs architecture. The limitations and its solutions showed that, even in a cognitive architecture such as PRIMs, facilitating skill reuse has not received enough attention. Additionally, implementing changes to support skill reuse raises many interesting questions about how cognitive architectures should be designed, after all, skill reuse is done by people and therefore cognitive architectures should be able to support it too.

Chapter 5 implemented the changes proposed in chapter 4 and applied the approach to the paradigm of executive functioning. This chapter shows that the changes we made to PRIMs and the skill-based approach strongly improved the feasibility of the skill-based approach in a more involved experimental paradigm. Additionally, it provided a new perspective on executive functioning. This supports the idea that using skill reuse as a basis for a modelling project can lead to interesting new insights.

2. The resulting modelling approach

The skill-based approach has gone through several stages of development during the different chapters of this dissertation. From a basic idea that every task consists of multiple smaller (reusable) pieces to the final conception of a skill being any collection of operators that has become associated with a certain goal and the inclusion of specific modelling conventions. We will now describe the main result of this dissertation, the final (at least at the end of this dissertation) version of the skill-based approach.

The skill-based approach centres around defining the basic skills for the to-be modelled task and assembling a model from these basic skills. Basic skills are the skills that a modeler can assume to already be learned when the task is performed for the first time (e.g., when the participant enters the lab). This means that stage two learning as described in the line of research started by Fitts (1964) is completed to a certain extent (the operators are associated with a certain purpose/goal) and the operators have undergone a basic level of compilation.

The first step of creating a model with the skill-based approach is identifying the basic processing steps required to complete the to-be modelled task and identifying which basic skills capable of accomplishing these steps a participant might already possess when entering the lab. This step represents the idea that the new task is (mostly) a composition of previously learned skills which only need to be applied to the new task. This can be done by looking at relevant literature and previously built models.

The second step is creating and validating the identified basic skills. This is done by creating other models that include the required basic skill that participants have done before or, alternatively, by creating models of tasks (which participants did not necessarily do before) but use the same basic skill. For example, in our initial model of the attentional blink, we created a model of a simple working memory task in order to build the consolidation skill which is also used for performing the attentional blink. We did not assume that participants had done this simple working memory task before but because this task is so similar to basic tasks people do every day (e.g., remembering a phone number), it still served as a helpful proxy and a good source to create the basic skill. The skill created in this way is subsequently validated by fitting the data produced by this model to real human data.

The final step is adapting the basic skills to fit the context of the current task. This final step depends on the cognitive architecture (or other modelling method) employed by the modeler. In general, this step should not fundamentally change the procedural knowledge of the skill but merely integrate the basic skill in the specific context of the task. For example, using a flexible and principled way in which symbols can be changed in a symbolic architecture or a way in which the operators can be flexibly linked to the new context (i.e., specifying which operator should fire at what moment).

Following the above described steps will result in a model that is more constrained, cognitively plausible and whose findings can be more easily translated to other models. Models are more constrained because it limits a modeler in how the operators or production-rules can be created. By embedding the productions used for the final model into a different model, the productions cannot be made too specifically for the final task or be modified slightly in order to better fit the data. Additionally, models will be more plausible because they will be created in a way that mirrors human cognition and the skills used in the model will already be supported by human data (i.e., in the other task used to create the basic skill) before being used in the final task improving the chance that the skill is performing in a realistic way. Finally, it will be easier to translate the findings of a model to other models and, crucially, it will be easier to integrate findings from other models into a new model. By using the skill-based approach, the process of creating a cognitive model consists for a large part out of reusing already existing skills. These skills might be created by other researchers from other fields studying different experimental paradigms and using these skills in a new model would automatically integrate these findings into the new model. This will be a very explicit gradual building of knowledge wherein any new modelling project will integrate all available relevant previous knowledge.

3. Skill-based approach in other cognitive architectures

In this dissertation we focused on implementing the skill-based approach in the cognitive architecture PRIMs. However, the general principles of the approach are not limited to this architecture alone, they can be implemented in other architectures as well. Although we did not explicitly investigate which obstacles will be encountered when implementing the skill-based approach in specific other cognitive architectures, the general issues that are present in PRIMs will most likely generalize to other architectures.

The fundamental challenge in creating a model from reusable parts is the balance between being general enough to work in multiple contexts while still remaining specific enough to model the data in one specific task. Almost all cognitive architectures or other modelling methods are primarily designed to support creating models for one specific task. Therefore, the basic structure of these architectures and methods will be strongly leaning towards being too specific and lose the ability to create fully generalizable models. We identified three crucial general areas which might lead to reuse limitations when using an architecture designed for single-task models. More details on these issues and how we solved them in PRIMs can be found in Chapters 4 and 5.

The first area concerns how to store short-term relevant information (i.e., working memory). Storing short-term information is crucial for a model to function properly. This information has to be quickly and reliably available and therefore needs to be stored in a very predictable location. The model needs to both know the location of every relevant item (e.g., where is the phone number stored) in order to quickly retrieve it as well as know what is stored in every location (e.g., a phone number is stored here) in order to quickly update the correct information. This is not a large issue for models that only have to model a single task, because the same type of information can always be stored in the same location: location and identity are equal. However, when a model needs to be designed to work in multiple contexts, this equality of identity and location does not always hold. For the one model it might be logical to store a phone number in slot 1, but a different model might already use slot 1 for something else. Therefore, location and identity cannot be assumed to be equal for reusable skills and a different method of indicating what is where and where is what is required. This method has to be flexible without becoming cognitively implausible.

The second area concerns how to accomplish goal selection. Goals are commonly assumed to be crucial for organizing behaviour and cognition and they play a central role in many cognitive architectures such as ACT-R (Anderson et al., 2004), SOAR (Laird, 2012; Laird, Newell, & Rosenbloom, 1987) and PRIMs (Taatgen, 2013). These architectures assume that every task is divided into a certain number of goals and that higher-level goals (e.g., completing an entire task) is accomplished by performing (sub-)goals in the correct order. For example, in the attentional blink task, the first goal is to identify the targets from distractors, the second goal is to consolidate the encountered target into memory and the final goal is to give a response. Goal selection for a single-task model is very different from goal selection for models based on reusable skills. Single-task models often have a very clear order in which the goals have to be accomplished and, therefore, relationships between goals are often described in terms of other goals (e.g., perform ‘consolidation’ after ‘identify’), however this is not the case for a model based on reusable skills. Designing a way of accomplishing goal selection that works for reusable skills cannot rely on knowledge about what the previous goal was. Instead it has to be able to be flexibly defined based on context cues (Altmann & Trafton, 2002).

The third area concerns how to specify the context in which an action should be taken. Most cognitive architectures provide a wide range of possible actions that can be executed by a model. It is, therefore, not a small task for a modeler to make sure that the model executes the correct action at the right time. This is usually done by providing a context in which it should be executed for every action. Context usually refers to the situation in the workspace (i.e., what information is in the buffer slots). For single-task models it is usually possible to come up with the correct context for every task, because the number of actions is limited and the number of possible contexts is not too large. Additionally, every action is usually only required to be selected in one or a few contexts. When constructing multi-task models, it becomes clear quite quickly that it is very difficult to specify the context in which an action should be taken, because there are many possible actions, the same action has to be chosen in sometimes very different contexts, and the actions encounter many different contexts in which they should not be chosen. This difficulty is mainly caused by two characteristics of the way in which most cognitive architectures require action-context combinations to be specified. The first characteristic is that the context in which an action is supposed to be chosen is fundamentally linked to the action itself and unchangeable (e.g., in ACT-R the right-hand side of a production-rule cannot exist without its left-hand side). Because of this assumption, every action needs to be separately defined for every situation that it should be chosen by the model. The second characteristic is that the context requirements for an action to be chosen have to be fully met. If a single requirement is not met, an action will not be picked. This makes the action-context combination very inflexible. In order for actions to be picked flexibly, the action selection mechanism has to be different. The context in which an action can be chosen should not be fundamentally linked to the action itself (or it should be very limited at least) and the mechanism with which an action is selected should not be based on an all-or-nothing decision but, instead, be more gradual. A final issue with the current action selection mechanism is that it does not reflect current neurological evidence. Action selection is thought to be the result of a relatively simple reward learning process in the Basal Ganglia (Breiter, Aharon, Kahneman, Dale, & Shizgal, 2001; Cisek & Kalaska, 2010; Redgrave, Prescott, & Gurney, 1999). Such a process is unlikely to end up with an explicit summary of the correct context for every action.

To summarize, most architectures are designed for single-task models, therefore their designs possess certain characteristics that make creating multi-task models very difficult. These limitations include short-term memory, goal selection and action selection. However, this is not an extensive list of all issues and not all architectures experience the same issues to the same extent. One commonality among these issues is that the model relies on input from the modeler about certain task-specific characteristics: it relies on the modeler indicating what goes where in short-term memory (and maintaining this implicit organization throughout the model), it relies on the modeler providing the order in which the goals should be accomplished, and it relies on the modeler providing the exact context in which an action should be selected. This task-specific input from the modeler is often completely interwoven with the model which makes it very likely that a model created this way is too task-specific. Additionally, the mechanisms currently in place to inject task-specific information into a model are not designed for multi-task models. They are often not flexible enough, too interwoven with the (in theory) generalizable parts of the model and hard to capture with simple learning mechanisms.

4. Roadblocks to adoption of the skill-based approach

The skill-based approach is a very promising modelling method and could prove to be very valuable to many projects. However, this raises a crucial question for the conclusion of this dissertation. How likely is it that the skill-based approach will be adopted by the wider modelling community and what might the roadblocks towards this path be?

Firstly, the research field is a crucial factor for the usefulness and usability of the skill-based approach. The skill-based approach is most suitable to fields studying higher levels of cognition and experimental paradigms involving tasks that study the interaction of several skills. Very fundamental cognitive research (e.g., studying the basic functioning of working memory) will profit less directly from applying the skill-based approach. However, even in such fundamental research, the idea behind the skill-based approach is still very valuable. Considering that the task being studied is part of the larger picture of cognition and fundamentally accomplished by skills will mediate the tunnel vision that often occurs when a paradigm is thoroughly investigated (e.g., the attentional blink). Furthermore, it would also facilitate the permeation of the knowledge gained by fundamental research into other (possibly more applied) fields of study. A related potential roadblock is that the modelling 'software' used is highly instrumental in the ease with which the skill-based approach can be applied. Paradigms that model cognition on a very basic level (e.g., neural networks) or a very abstract level (e.g., accumulator models) will be less conducive to supporting the skill-based approach. However, for such models it can still prove very valuable to break the modelled task down into composite skills and consider which of these skills are responsible for producing the modelled effects. It will link the lower level behaviours of the neural networks to higher levels of behaviour and cognition and it will remove a large part of the abstraction inherent in the abstract models such as accumulator models.

A second challenge in the application of the skill-based approach is that even in suitable modelling paradigms and cognitive architectures, using the approach will take more time and effort than not using it. Fully applying the skill-based approach requires a modeler to create additional models next to the model the modeler is interested in. This will always take more time than only creating the model of the task that the modeler intends to model. Additionally, this modeler would require data to fit the basic models which might require running additional experiments or searching the literature for suitable datasets. Furthermore, as we experienced during the model building for this dissertation, using the same skill for multiple models is challenging since changing one aspect of the skill to fit one task will often result in changing model behaviour on the other tasks which might result in a vicious domino effect of changes. Although this extra labour and time cost is considerable, it might only be temporary since over time a kind of library of skills will develop which will eventually greatly speed up the modelling process (although this 'library-effect' will be limited to users of the same architecture). However, for this case, a middle-ground solution might prove to be the most effective. Merely composing a model out of skills will already provide most of the benefits of using the skill-based approach without any of the extra time investment. By determining which skills are used in the modelled task and using the structure in the model will already improve the generalizability and explainability of this model greatly.

To conclude, fully applying the skill-based approach might prove to be too challenging in many situations, however, considering the philosophy behind it that every task is a composition of skills will be very valuable in improving the generalizability and plausibility of models in all fields of cognitive science.

Bibliography

Agresti, A., & Coull, B. A. (1998). Approximate Is Better than "Exact" for Interval Estimation of Binomial Proportions. *The American Statistician*, *52*(2), 119–126. https://doi.org/10.2307/2685469

Akyürek, E. G., Abedian-Amiri, A., & Ostermeier, S. M. (2011). Content-Specific Working Memory Modulation of the Attentional Blink. *PLOS ONE*, *6*(2), e16696. Retrieved from https://doi.org/10.1371/journal.pone.0016696

Akyurek, E. G., Eshuis, S. A. H., Nieuwenstein, M. R., Saija, J. D., Baskent, D., & Hommel, B. (2012). Temporal target integration underlies performance at Lag 1 in the attentional blink. *Journal of Experimental Psychology. Human Perception and Performance*, *38*(6), 1448–1464. https://doi.org/10.1037/a0027610

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: an activation-based model. *Cognitive Science*, *26*(1), 39–83. https://doi.org/https://doi.org/10.1207/s15516709cog2601_2

Anderson, J., & Douglass, S. (2001). Tower of Hanoi: Evidence for the Cost of Goal Retrieval. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *27*, 1331–1346. https://doi.org/10.1037/0278-7393.27.6.1331

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, *89*(4), 369.

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, *22*(3), 261–295.

Anderson, J. R. (2007). How can the human mind occur in the physical universe? *How Can the Human Mind Occur in the Physical Universe?* New York, NY, US: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195324259.001.0001

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

Anderson, J. R., Bothell, D., Fincham, J. M., Anderson, A. R., Poole, B., & Qin, Y. (2011). Brain regions engaged by part- and whole-task

performance in a video game: a model-based test of the decomposition hypothesis. *Journal of Cognitive Neuroscience*, *23*(12), 3983–3997. https://doi.org/10.1162/jocn_a_00033
Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An Integrated Theory of List Memory. *Journal of Memory and Language*, *38*(4), 341–380. https://doi.org/10.1006/jmla.1997.2553
Anderson, J. R., & Schooler, L. J. (1991). Reflections of the Environment in Memory. *Psychological Science*, *2*(6), 396–408. https://doi.org/10.1111/j.1467-9280.1991.tb00174.x
Arend, I., Johnston, S., & Shapiro, K. (2006). Task-irrelevant visual motion and flicker attenuate the attentional blink. *Psychonomic Bulletin & Review*, *13*(4), 600–607. https://doi.org/10.3758/BF03193969
Bailey, C. E. (2007). Cognitive accuracy and intelligent executive function in the brain and in business. *Annals of the New York Academy of Sciences*, *1118*(1), 122–141.
Baler, R. D., & Volkow, N. D. (2006). Drug addiction: the neurobiology of disrupted self-control. *Trends in Molecular Medicine*, *12*(12), 559–566.
Barch, D. M. (2005). The cognitive neuroscience of schizophrenia. *Annual Review of Clinical Psychology(2005)*, *1*(1), 321–353.
Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software; Vol 1, Issue 1 (2015)* . Retrieved from https://www.jstatsoft.org/v067/i01
Beanland, V., & Pammer, K. (2012). Minds on the blink: The relationship between inattentional blindness and attentional blink. *Attention, Perception, & Psychophysics*, *74*(2), 322–330. https://doi.org/10.3758/s13414-011-0241-4
Bergman Nutley, S., Söderqvist, S., Bryde, S., Thorell, L. B., Humphreys, K., & Klingberg, T. (2011). Gains in fluid intelligence after training non-verbal reasoning in 4-year-old children: A controlled, randomized study. *Developmental Science*, *14*(3), 591–601.
Blair, C. (2017). Educating executive function. *Wiley Interdisciplinary Reviews: Cognitive Science*, *8*(1–2), e1403.
Borella, E., Carretti, B., & Pelegrina, S. (2010). The specific role of inhibition in reading comprehension in good and poor comprehenders.

Journal of Learning Disabilities, *43*(6), 541–552.

Borst, J. P., Taatgen, N. A., & van Rijn, H. (2010). The problem state: a cognitive bottleneck in multitasking. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *36*(2), 363–382. https://doi.org/10.1037/a0018106

Bowman, H., & Wyble, B. (2007). The simultaneous type, serial token model of temporal attention and working memory. *Psychological Review*, *114*, 38–70. https://doi.org/10.1037/0033-295X.114.1.38

Breiter, H. C., Aharon, I., Kahneman, D., Dale, A., & Shizgal, P. (2001). Functional Imaging of Neural Responses to Expectancy and Experience of Monetary Gains and Losses. *Neuron*, *30*(2), 619–639. https://doi.org/https://doi.org/10.1016/S0896-6273(01)00303-8

Brown, T. E. (2006). Executive functions and attention deficit hyperactivity disorder: Implications of two conflicting views. *International Journal of Disability, Development and Education*, *53*(1), 35–46.

Carlson, S. M., Moses, L. J., & Claxton, L. J. (2004). Individual differences in executive functioning and theory of mind: An investigation of inhibitory control and planning ability. *Journal of Experimental Child Psychology*, *87*(4), 299–319.

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software; Vol 1, Issue 6 (2014)* . https://doi.org/10.18637/jss.v061.i06

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences*, *3*(2), 57–65. https://doi.org/10.1016/s1364-6613(98)01273-x

Choi, H., Chang, L.-H., Shibata, K., Sasaki, Y., & Watanabe, T. (2012). Resetting capacity limitations revealed by long-lasting elimination of attentional blink through training. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(30), 12242–12247. https://doi.org/10.1073/pnas.1203972109

Chow, S.-C., Wang, H., & Shao, J. (2007). *Sample size calculations in clinical research*. CRC press.

Chun, M. M., & Potter, M. C. (1995). A two-stage model for multiple target detection in rapid serial visual presentation. *Journal of Experimental Psychology: Human Perception and Performance*. US: American

Psychological Association. https://doi.org/10.1037/0096-1523.21.1.109

Cisek, P., & Kalaska, J. F. (2010). Neural mechanisms for interacting with a world full of action choices. *Annual Review of Neuroscience*, *33*, 269–298.

Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: a source activation account. *Cognitive Science*, *25*(3), 315–353. https://doi.org/10.1016/S0364-0213(01)00039-8

Di Lollo, V., Kawahara, J.-I., Ghorashi, S., & Enns, J. (2005). The attentional blink: Resource depletion or temporary loss of control? *Psychological Research*, *69*, 191–200. https://doi.org/10.1007/s00426-004-0173-x

Diamond, A. (2013). Executive functions. *Annual Review of Psychology*, *64*, 135–168. https://doi.org/10.1146/annurev-psych-113011-143750

Diamond, A., & Lee, K. (2011). Interventions shown to aid executive function development in children 4 to 12 years old. *Science (New York, N.Y.)*, *333*(6045), 959–964. https://doi.org/10.1126/science.1204529

Duijkers, J. C. L. M., Vissers, C. T. W. M., & Egger, J. I. M. (2016). Unraveling executive functioning in dual diagnosis. *Frontiers in Psychology*, *7*, 979.

Duncan, J., Johnson, R., Swales, M., & Freer, C. (1997). Frontal lobe deficits after head injury: Unity and diversity of function. *Cognitive Neuropsychology*. United Kingdom: Taylor & Francis. https://doi.org/10.1080/026432997381420

Dux, P. E., & Marois, R. (2009). The attentional blink: A review of data and theory. *Attention, Perception, & Psychophysics*, *71*(8), 1683–1700.

Ellis, K. A., Silberstein, R. B., & Nathan, P. J. (2006). Exploring the temporal dynamics of the spatial working memory n-back task using steady state visual evoked potentials (SSVEP). *NeuroImage*, *31*(4), 1741–1751.

Feldman, J. (2013). The neural binding problem(s). *Cognitive Neurodynamics*, *7*(1), 1–11. https://doi.org/10.1007/s11571-012-9219-8

Ferlazzo, F., Lucido, S., Di Nocera, F., Fagioli, S., & Sdoia, S. (2007).

Switching between goals mediates the attentional blink effect. *Experimental Psychology*, *54*(2), 89–98. https://doi.org/10.1027/1618-3169.54.2.89

Fitts, P. M. (1964). Perceptual-motor skill learning. In *Categories of human learning* (pp. 243–285). Elsevier.

Friedman, N. P., & Miyake, A. (2017). Unity and diversity of executive functions: Individual differences as a window on cognitive structure. *Cortex*, *86*, 186–204.

Gathercole, S. E., Pickering, S. J., Knight, C., & Stegmann, Z. (2004). Working memory skills and educational attainment: Evidence from national curriculum assessments at 7 and 14 years of age. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, *18*(1), 1–16.

Giesbrecht, B., & Di Lollo, V. (1998). Beyond the attentional blink: visual masking by object substitution. *Journal of Experimental Psychology: Human Perception and Performance*, *24*(5), 1454.

Giesbrecht, F. G., & Burns, J. C. (1985). Two-Stage Analysis Based on a Mixed Model: Large-Sample Asymptotic Theory and Small-Sample Simulation Results. *Biometrics*, *41*(2), 477–486. https://doi.org/10.2307/2530872

Gray, S. A., Chaban, P., Martinussen, R., Goldberg, R., Gotlieb, H., Kronitz, R., … Tannock, R. (2012). Effects of a computerized working memory training program on working memory, attention, and academics in adolescents with severe LD and comorbid ADHD: a randomized controlled trial. *Journal of Child Psychology and Psychiatry*. Tannock, R.: Hospital for Sick Children, 555 University Avenue, Toronto, ON, Canada, M5G 1X8, rosemary.tannock@utoronto.ca: Wiley-Blackwell Publishing Ltd. https://doi.org/10.1111/j.1469-7610.2012.02592.x

Greff, K., van Steenkiste, S., & Schmidhuber, J. (2020). On the binding problem in artificial neural networks. *ArXiv Preprint ArXiv:2012.05208*.

Hoekstra, C., Martens, S., & Taatgen, N. A. (2020). A Skill-Based Approach to Modeling the Attentional Blink. *Topics in Cognitive Science*, *12*(3), 1030–1045. https://doi.org/10.1111/tops.12514

Hoekstra, C., Martens, S., & Taatgen, N. A. (2022a). Obstacles to the Skill-

Based Approach: Why is skill reuse so difficult for cognitive architectures? In T. C. Stewart (Ed.), *Proceedings of the 20th International Conference on Cognitive Modeling*.

Hoekstra, C., Martens, S., & Taatgen, N. A. (2022b). Testing the skill-based approach: Consolidation strategy impacts attentional blink performance. *Plos One*, *17*(1), e0262350.

Hrong-Tai Fai, A., & Cornelius, P. L. (1996). Approximate F-tests of multiple degree of freedom hypotheses in generalized least squares analyses of unbalanced split-plot experiments. *Journal of Statistical Computation and Simulation*, *54*(4), 363–378. https://doi.org/10.1080/00949659608811740

Huizinga, M., Dolan, C. V, & Van der Molen, M. W. (2006). Age-related change in executive function: Developmental trends and a latent variable analysis. *Neuropsychologia*, *44*(11), 2017–2036.

Jaeggi, S. M., Buschkuehl, M., Perrig, W. J., & Meier, B. (2010). The concurrent validity of the N-back task as a working memory measure. *Memory*, *18*(4), 394–412.

Jolicoeur, P., & Dell'Acqua, R. (1998). The demonstration of short-term consolidation. *Cognitive Psychology*, *36*(2), 138–202. https://doi.org/10.1006/cogp.1998.0684

Jurado, M. B., & Rosselli, M. (2007). The elusive nature of executive functions: A review of our current understanding. *Neuropsychology Review*, *17*(3), 213–233.

Kamijo, K., Pontifex, M. B., O'Leary, K. C., Scudder, M. R., Wu, C., Castelli, D. M., & Hillman, C. H. (2011). The effects of an afterschool physical activity program on working memory in preadolescent children. *Developmental Science*, *14*(5), 1046–1058.

Karbach, J., & Kray, J. (2009). How useful is executive control training? Age differences in near and far transfer of task-switching training. *Developmental Science*, *12*(6), 978–990.

Kassambara, A. (2017). *Practical guide to principal component methods in R: PCA, M (CA), FAMD, MFA, HCPC, factoextra* (Vol. 2). STHDA.

Kim, J. W., Ritter, F. E., & Koubek, R. J. (2013). An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical Issues in Ergonomics Science*, *14*(1), 22–37. https://doi.org/10.1080/1464536X.2011.573008

Kimberg, D. Y., & Farah, M. J. (1993). A unified account of cognitive impairments following frontal lobe damage: the role of working memory in complex, organized behavior. *Journal of Experimental Psychology: General*, *122*(4), 411.

Klingberg, T. (2010). Training and plasticity of working memory. *Trends in Cognitive Sciences*, *14*(7), 317–324.

Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17–94. https://doi.org/10.1007/s10462-018-9646-y

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software; Vol 1, Issue 13 (2017)* . https://doi.org/10.18637/jss.v082.i13

Lee, F. J., & Anderson, J. R. (2001). Does Learning a Complex Task Have to Be Complex?: A Study in Learning Decomposition. *Cognitive Psychology*, *42*(3), 267–316. https://doi.org/https://doi.org/10.1006/cogp.2000.0747

Lehto, J. E., Juujärvi, P., Kooistra, L., & Pulkkinen, L. (2003). Dimensions of executive functioning: Evidence from children. *British Journal of Developmental Psychology*, *21*(1), 59–80.

Logan, G. D. (1985). Executive control of thought and action. *Acta Psychologica*, *60*(2–3), 193–210.

MacLean, M. H., & Arnell, K. M. (2012). A conceptual and methodological framework for measuring and modulating the attentional blink. *Attention, Perception & Psychophysics*, *74*(6), 1080–1097. https://doi.org/10.3758/s13414-012-0338-4

Martens, S., Munneke, J., Smid, H., & Johnson, A. (2006). Quick minds don’t blink: electrophysiological correlates of individual differences in attentional selection. *Journal of Cognitive Neuroscience*, *18*(9), 1423–1438. https://doi.org/10.1162/jocn.2006.18.9.1423

Martens, S., & Wyble, B. (2010). The attentional blink: past, present, and future of a blind spot in perceptual awareness. *Neuroscience and Biobehavioral Reviews*, *34*(6), 947–957. https://doi.org/10.1016/j.neubiorev.2009.12.005

Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-

source, graphical experiment builder for the social sciences. *Behavior Research Methods*, *44*(2), 314–324.
Miller, H. V., Barnes, J. C., & Beaver, K. M. (2011). Self-control and health outcomes in a nationally representative sample. *American Journal of Health Behavior*, *35*(1), 15–27.
Miyake, A., & Friedman, N. P. (2012). The Nature and Organization of Individual Differences in Executive Functions: Four General Conclusions. *Current Directions in Psychological Science*, *21*(1), 8–14. https://doi.org/10.1177/0963721411429458
Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex "frontal lobe" tasks: A latent variable analysis. *Cognitive Psychology*, *41*(1), 49–100.
Monsell, S. (2003). Task switching. *Trends in Cognitive Sciences*, *7*(3), 134–140.
Morris, N., & Jones, D. M. (1990). Memory updating in working memory: The role of the central executive. *British Journal of Psychology*, *81*(2), 111–121.
Morrison, A. B., & Chein, J. M. (2011). Does working memory training work? The promise and challenges of enhancing cognition by training working memory. *Psychonomic Bulletin & Review*, *18*(1), 46–60.
Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, *64*(5), 482.
Newell, A. (1990). Unified theories of cognition. *Unified Theories of Cognition.* Cambridge, MA, US: Harvard University Press.
Nijboer, M., Borst, J., van Rijn, H., & Taatgen, N. (2016). Contrasting single and multi-component working-memory systems in dual tasking. *Cognitive Psychology*, *86*, 1–26. https://doi.org/https://doi.org/10.1016/j.cogpsych.2016.01.003
Olivers, C. N. L., & Meeter, M. (2008). A boost and bounce theory of temporal attention. *Psychological Review*, *115*(4), 836–863. https://doi.org/10.1037/a0013395
Olivers, C. N. L., & Nieuwenhuis, S. (2005). The beneficial effect of concurrent task-irrelevant mental activity on temporal attention. *Psychological Science*, *16*(4), 265–269. https://doi.org/10.1111/j.0956-7976.2005.01526.x

Olivers, C. N. L., van der Stigchel, S., & Hulleman, J. (2007). Spreading the sparing: against a limited-capacity account of the attentional blink. *Psychological Research*, *71*(2), 126–139. https://doi.org/10.1007/s00426-005-0029-z

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., … Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

Pennington, B. F. (1997). Dimensions of executive functions in normal and abnormal development. *Development of the Prefrontal Cortex: Evolution, Neurobiology, and Behavior.* Baltimore, MD, US: Paul H Brookes Publishing.

Proctor, R. W., & Dutta, A. (1995). Skill acquisition and human performance. *Skill Acquisition and Human Performance.* Thousand Oaks, CA, US: Sage Publications, Inc.

Psychology Software Tools. (2012). E-Prime 2.0. Pittsburgh, PA. https://doi.org/http://www.pstnet.com

R Core Team. (2015). RStudio: Integrated Development for R. Rstudio, Inc. Retrieved from http://www.rstudio.com/

R Core Team. (2017). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

Rabbitt, P. (1997). *Methodology of frontal and executive function*. Psychology Press Hove, UK.

Rasmussen, J. (1987). *Information processing and human-machine interaction. An approach to cognitive engineering*. North-Holland.

Raymond, J. E., Shapiro, K. L., & Arnell, K. M. (1992). Temporary suppression of visual processing in an RSVP task: An attentional blink? *Journal of Experimental Psychology: Human Perception and Performance*. US: American Psychological Association. https://doi.org/10.1037/0096-1523.18.3.849

Redgrave, P., Prescott, T. J., & Gurney, K. (1999). The basal ganglia: a vertebrate solution to the selection problem? *Neuroscience*, *89*(4), 1009–1023. https://doi.org/10.1016/s0306-4522(98)00319-4

Redick, T. S., & Lindsey, D. R. B. (2013). Complex span and n-back measures of working memory: A meta-analysis. *Psychonomic Bulletin & Review*, *20*(6), 1102–1113.

Riggs, N. R., Spruijt-Metz, D., Sakuma, K.-L., Chou, C.-P., & Pentz, M. A. (2010). Executive cognitive function and food intake in children. *Journal of Nutrition Education and Behavior*, *42*(6), 398–403.

Roberts, R. J., Hager, L. D., & Heron, C. (1994). Prefrontal cognitive processes: Working memory and inhibition in the antisaccade task. *Journal of Experimental Psychology: General*, *123*(4), 374.

Rogers, R. D., & Monsell, S. (1995). Costs of a predictible switch between simple cognitive tasks. *Journal of Experimental Psychology: General*. US: American Psychological Association. https://doi.org/10.1037/0096-3445.124.2.207

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, *48*(2), 362–380.

Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, *37*(5), 829–860. https://doi.org/10.1111/cogs.12032

Schooler, L. J., & Anderson, J. R. (1997). The role of process in the rational analysis of memory. *Cognitive Psychology*. Netherlands: Elsevier Science. https://doi.org/10.1006/cogp.1997.0652

Seiffert, A. E., & Di Lollo, V. (1997). Low-level masking in the attentional blink. *Journal of Experimental Psychology: Human Perception and Performance*, *23*(4), 1061.

Shapiro, K L, Raymond, J. E., & Arnell, K. M. (1994). Attention to visual pattern information produces the attentional blink in rapid serial visual presentation. *Journal of Experimental Psychology. Human Perception and Performance*, *20*(2), 357–371. https://doi.org/10.1037//0096-1523.20.2.357

Shapiro, Kimron L, Caldwell, J., & Sorensen, R. E. (1997). Personal names and the attentional blink: A visual “cocktail party” effect. *Journal of Experimental Psychology: Human Perception and Performance*. US: American Psychological Association. https://doi.org/10.1037/0096-1523.23.2.504

Shipstead, Z., Redick, T. S., & Engle, R. W. (2012). Is working memory training effective? *Psychological Bulletin*, *138*(4), 628.

Smith, E. E., & Jonides, J. (1999). Storage and executive processes in the frontal lobes. *Science*, *283*(5408), 1657–1661.

Stearns, B., & Laird, J. (2018). Modeling Instruction Fetch in Procedural Learning. In *Proceedings of the 16th international conference on* *cognitive modeling*.

Stewart, T. C., Bekolay, T., & Eliasmith, C. (2012). Learning to select actions with spiking neurons in the Basal Ganglia. *Frontiers in Neuroscience*, *6*, 2. https://doi.org/10.3389/fnins.2012.00002

Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: a model of the basal ganglia's role in cognitive coordination. *Psychological Review*, *117*(2), 541–574. https://doi.org/10.1037/a0019077

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*. US: Psychological Review Company. https://doi.org/10.1037/h0054651

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*(3), 439–471. https://doi.org/10.1037/a0033138

Taatgen, N. A. (2014). Between architecture and model: Strategies for cognitive control. *Biologically Inspired Cognitive Architectures*, *8*, 132–139. https://doi.org/10.1016/j.bica.2014.03.010

Taatgen, N. A. (2020). A spiking neural architecture that learns tasks. In *Proceedings of ICCM 2019 - 17th International Conference on Cognitive Modeling* (pp. 253–258).

Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology. General*, *137*(3), 548–565. https://doi.org/10.1037/0096-3445.137.3.548

Taatgen, N. A., Juvina, I., Schipper, M., Borst, J., & Martens, S. (2009). Too much control can hurt: A threaded cognition model of the Attentional Blink. *Cognitive Psychology*, *59*, 1–29. https://doi.org/10.1016/j.cogpsych.2008.12.002

Taatgen, N. A., Van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review*, *114*(3), 577.

Tavares, J. V. T., Clark, L., Cannon, D. M., Erickson, K., Drevets, W. C., & Sahakian, B. J. (2007). Distinct profiles of neurocognitive function in unmedicated unipolar depression and bipolar II depression. *Biological Psychiatry*, *62*(8), 917–924.

Thomson, D. R., Besner, D., & Smilek, D. (2015). A Resource-Control

Account of Sustained Attention: Evidence From Mind-Wandering and Vigilance Paradigms. *Perspectives on Psychological Science*, *10*(1), 82–96. https://doi.org/10.1177/1745691614556681

van der Wel, P., & van Steenbergen, H. (2018). Pupil dilation as an index of effort in cognitive control tasks: A review. *Psychonomic Bulletin & Review*, *25*(6), 2005–2015. https://doi.org/10.3758/s13423-018-1432-y

Van Rij, J., van Rijn, H., & Hendriks, P. (2012). How WM load influences pronoun interpretation. *Nele Rußwinkel| Uwe Drewitz| Hedderik van Rijn (Eds.)*, 101.

Visser, T. (2007). Masking T1 Difficulty: Processing Time and the Attentional Blink. *Journal of Experimental Psychology. Human Perception and Performance*, *33*, 285–297. https://doi.org/10.1037/0096-1523.33.2.285

Wierda, S. M., van Rijn, H., Taatgen, N. A., & Martens, S. (2012). Pupil dilation deconvolution reveals the dynamics of attention at high temporal resolution. *Proceedings of the National Academy of Sciences*, *109*(22), 8456 LP – 8460.

Willems, C., Damsma, A., Wierda, S. M., Taatgen, N., & Martens, S. (2015). Training-induced Changes in the Dynamics of Attention as Reflected in Pupil Dilation. *Journal of Cognitive Neuroscience*, *27*(6), 1161–1171. https://doi.org/10.1162/jocn_a_00767

Willems, C., & Martens, S. (2016). Time to see the bigger picture: Individual differences in the attentional blink. *Psychonomic Bulletin & Review*, *23*, 1289–1299. https://doi.org/10.3758/s13423-015-0977-2

Willems, C., Wierda, S. M., van Viegen, E., & Martens, S. (2013). Individual Differences in the Attentional Blink: The Temporal Profile of Blinkers and Non-Blinkers. *PLOS ONE*, *8*(6), e66185. Retrieved from https://doi.org/10.1371/journal.pone.0066185

Wyble, B., Bowman, H., & Nieuwenstein, M. (2009). The attentional blink provides episodic distinctiveness: sparing at a cost. *Journal of Experimental Psychology. Human Perception and Performance*, *35*(3), 787–807. https://doi.org/10.1037/a0013902

Yntema, D. B. (1963). Keeping track of several things at once. *Human Factors*, *5*, 7–17. https://doi.org/10.1177/001872086300500102

Acknowledgements

Ik had mijn dissertatie nooit kunnen voltooien zonder de waardevolle hulp van de mensen om me heen. Ten eerste mijn twee begeleiders, Niels en Sander. Zonder jullie was dit proefschrift er natuurlijk nooit gekomen.

Niels jij was altijd beschikbaar voor vragen en hebt me wegwijs gemaakt in de toen voor mij nog onbekende wereld van cognitief modelleren. Je hebt een uitstekende kijk op het proces van onderzoek doen en ik heb hier ontzettend veel van geleerd. Je hebt altijd een originele kijk op vele onderwerpen en dit leidt vaak tot interessante onderzoeken. Verder waardeer ik heel erg dat je mij vrijliet om zelf na te denken over waar we heen wilden met dit project maar dat je bijstuurde wanneer dit nodig was (het is blijkbaar best lastig om een vier+-jarig onderzoeksproject te doen).

En, Sander, wij werken al een hele tijd samen inmiddels. Het begon voor mij als een master student die geïnteresseerd was in de Attentional Blink en op zoek was naar een begeleider die hier net zo enthousiast over was als ik en het eindige met een proefschrift waarin dit onderwerp weer terugkwam. Ik heb tijdens mijn Master project en mijn promotietraject veel geleerd van jou over schrijven, presenteren en je hebt me in deze tijd altijd veel gesteund en vertrouwen gegeven in een goede afloop.

Additionally, I want to thank all the people of the ‘lunch-crew’: Stefan, Hang, Maarten (soms), Stephen, Lionel, Hermine, Katja, Yoezy, Su, Christina and everyone else who joined occasionally. In het bijzonder, Stefan, jij hebt me ontzettend veel geholpen met het uitvogelen van de soms mysterieuze werkingen van PRIMs. Of course, I should not forget about the two extra members who only joined on Friday to eat kibbeling and talk about the dynamic weather of the Netherlands, Yuri and Stipe. Also, I want to thank all the members of the Cognitive Modeling group for their support and input during the

weekly CogMod meetings and random conversations in the hallways: Jelmer, Jacolien, Marieke, Joshua, Moritz, Catherine, Andrea, Jennifer, Mega (also my office mate), Thomas, Mark, Ivo, Mi, Gabriel, Katja and Fokie.

Verder wil ik ook even de mensen benoemen met wie ik bijna elke dag heb gewerkt in de master studenten ruimte in het Heymans in de tijd vlak voordat ik met mijn PhD begon: Elef, Robbert, Tabitha, Joost, Ko, Marlijn, Jesse, Cecilia, Yavor, Wouter, Alex en de senior member van de groep Mark Span. Het was altijd erg gezellig en ik was geen PhD gaan doen zonder jullie steun! En ook de event-related potentials crew: Cees, Muriel, Martine, Jolijn.

Furthermore, I would like to thank the members of the reading committee Dario Salvucci, Jelmer Borst, and Dick de Waard for taking the time to read my dissertation and provide interesting and colorful comments. Zeker jij, Dick, dat je een dissertatie wilde lezen die zo fundamenteel is en niet direct in jouw interessegebied valt. Ik kan me je strenge feedback op mijn allereerste wetenschappelijke schrijfwerk in mijn tweede jaar als bachelor student nog goed herinneren, dus ik zet me schrap voor je vragen tijdens de verdediging.

En natuurlijk mijn ouders, Marcus en Wietske, mijn zusje, Annerens and also our close friends in Groningen, Edo and Rennik, you've made this time so much more enjoyable for our family. En ook mijn eigen gezinnetje Abby en Minke. Abby you have been a tremendous support during these years. Back in 2018 I could have never imagined that next to finishing my PhD I would have also started a family. En, Minke, mocht je dit een keer teruglezen als je wat ouder bent (op moment van schrijven ben je zes maanden), ik heb ontzettend veel zin om jou de komende jaren te zien opgroeien en in alle leuke dingen die we samen zullen gaan doen.

Nederlandse samenvatting

Mensen zijn ontzettend snel in het leren van (simpele) taken. Dit wijst erop dat we wanneer we geconfronteerd worden met een nieuwe taak, we niet van nul hoeven te beginnen met het uitvogelen van hoe de taak werkt maar dat we al bestaande kennis kunnen inzetten in deze nieuwe situatie. Dit inzicht is breed geaccepteerd onder cognitieve wetenschappers, maar wordt echter zeer beperkt toegepast tijdens het bouwen van cognitieve modellen. Bijna alle modellen worden gecreëerd voor een enkele taak zonder dat er aandacht voor is dat de onderdelen van dit model waarschijnlijk ook gebruikt worden in andere situaties. Dit kan ervoor zorgen dat een model te specifiek is en dus weinig bijdraagt aan de algemene kennis over cognitie. Om dit te voorkomen hebben wij een modelleermethode ontwikkeld die modelbouwers in staat stelt om modellen te bouwen op de manier zoals mensen het leren van nieuwe taken aanvliegen, namelijk met het hergebruiken van al bestaande vaardigheden. Volgens deze methode is het leren van een nieuwe taak slechts het juist herkennen van welke (al geleerde) vaardigheden nuttig zijn voor de nieuwe taak. De methode heet de skill-based methode omdat het ervan uitgaat dat elke taak bestaat uit (meerdere) basisvaardigheden (skills) die ingezet kunnen worden in meerdere situaties.

In hoofdstuk twee hebben we dit idee voorgesteld en hebben we de eerste versie van de skill-based methode ontwikkeld. In deze versie waren de stappen om de skill-based methode uit te voeren eerst het ontleden van de verwerkingsstappen die nodig waren voor het uitvoeren van een taak en daarna het bouwen van modellen van andere taken die deze zelfde verwerkingsstappen nodig hadden. Op deze manier konden we een model bouwen van de taak waar we geïnteresseerd in waren, zonder dat we specifiek voor deze taak iets nieuws hoefden te bouwen. Dit brachten we in de praktijk door middel van het creëren van een model van de attentional blink. De attentional blink is een cognitief fenomeen dat gevonden wordt in een taak waarin de proefpersonen twee ‘targets’ moeten ontdekken in een reeks van afleiders die zeer snel worden gepresenteerd (vaak met een snelheid van 10 items per seconde). De attentional blink is dan dat de tweede ‘target’ vaak wordt gemist als deze wordt gepresenteerd tussen 200 en 500 milliseconde na de eerste. Interessant genoeg wordt de tweede ‘target’ veel vaker gezien als deze direct na de eerste wordt gepresenteerd. De eerste stap van de skill-based methode is het bepalen van de onderliggende basisvaardigheden van een taak. Voor een attentional blink taak zijn dit: ‘target’ detectie, geheugen consolidatie, het ophalen van opgeslagen items uit het geheugen, en het geven van het antwoord. Vervolgens maakten we deze vaardigheden door het creëren van drie verschillende modellen, een simpel model dat ‘targets’ kan vinden, een model van een eenvoudige werkgeheugentaak en een model van een complexe werkgeheugentaak. Hiernaast had dit een model een strategiecomponent in de zin van dat het model de attentional blink taak met twee verschillende strategieën kon uitvoeren. De eerste strategie was het apart onthouden van de twee targets, dit hield in dat het model de twee ‘targets’ opsloeg in twee verschillende ‘chunks’. De tweede strategie was een ‘chunking’ strategie, dit hield in dat het model de twee ‘targets’ in het geheugen opsloeg als één ‘chunk’. Chunk betekent brok en is de wetenschappelijke term voor een object (‘herinnering’) in het declaratief geheugen; bijvoorbeeld de cijfers 1, 9, 8, 8 kunnen apart onthouden worden in vier verschillende ‘chunks’, maar ze kunnen ook worden opgeslagen als een geheel (in één chunk dus) als 1988. Het interessante aan deze twee strategieën is dat het model voorspelde dat er geen attentional blink zou optreden als iemand de tweede ‘chunking’ strategie zou toepassen. Dit is een verassende voorspelling omdat wordt aangenomen dat de attentional blink een fundamentele beperking van het cognitieve systeem is. Dit hoofdstuk wijst op twee rode draden die door de gehele dissertatie lopen. Ten eerste suggereert het dat een simpele taak (zoals de attentional blink) gedaan wordt door het simpelweg ophalen van basisvaardigheden en dat deze basisvaardigheden soms tot imperfecte prestaties kunnen leiden. Ten tweede toont het de fundamentele haalbaarheid van de skill-based methode en laat het zien dat deze methode tot plausibele en interessante nieuwe modellen kan komen.

In hoofdstuk drie testten we de voorspelling van het model dat we in het vorige hoofdstuk hebben gebouwd dat het gebruiken van een 'chunking' strategie voorkomt dat de attentional blink optreedt. Dit testten we ten eerste door het repliceren van een experiment gedaan door Ferlazzo et al. (2007). Dit experiment liet zien dat proefpersonen die de attentional blink uitvoerden met de instructies dat de twee 'targets' (letters in dit geval) een lettergreep vormden geen attentional blink vertoonden. Dit komt overeen met onze voorspelling dat het onthouden van de twee targets als één geheel (in dit geval als een lettergreep) ervoor zorgt dat de attentional blink vermeden kan worden. Echter repliceerden we deze resultaten niet. Dit kan komen door taalverschillen tussen de twee studies, onze proefpersonen hadden overwegend Nederlands als moedertaal terwijl de proefpersonen in de originele studie vooral Italiaans als moedertaal hadden. Omdat we nog steeds ons model wilden testen hebben we een tweede experiment uitgevoerd waarin we expliciet de strategie wilden manipuleren in combinatie met 'targets' die makkelijker als één geheeld onthouden konden worden. Dit leidde tot een merendeels succesvolle test van de voorspelling van ons model. Via k-means clustering vonden we drie clusters van proefpersonen met verschillende prestatie patronen. We vonden een groot cluster van proefpersonen die geen attentional blink ervaarden tijdens het experiment. Dit laat zien dat het gebruiken van 'targets' die makkelijk als één geheel onthouden kunnen worden leidt tot een grote verbetering van de prestatie in een attentional blink taak. Echter, deze prestatieverbetering kwam niet alleen voor in de groep die de instructies had gekregen om de 'targets' als één geheel te onthouden dus we kunnen niet met volledige zekerheid vaststellen dat deze prestatieverbetering voortkomt uit het toepassen van de 'chunking' strategie. Dit hoofdstuk draagt twee belangrijke inzichten toe aan de uiteindelijke conclusie van deze dissertatie. Ten eerste laat het zien dat de attentional blink niet een fundamentele beperking is omdat het niet optrad in een grote groep proefpersonen in het tweede experiment. Ten tweede laat het zien dat modellen die gebouwd zijn met de skill-based approach interessante nieuwe inzichten kunnen leveren, ook in een paradigma dat al vele jaren heel intensief is bestudeerd. In dit geval is dit inzicht dat de attentional blink voortkomt uit het normale functioneren van het cognitieve systeem zonder dat er extra aannames voor nodig zijn.

In hoofdstuk vier verlegden we de focus weer naar de ontwikkeling van de modelleer methode. Nadat bleek dat de methode in principe haalbaar is door het creëren van de modellen van de 'attentional blink', pasten we de methode toe op een grotere set van taken. Deze taken zijn de negen basistaken van executieve functies zoals beschreven in Miyake et al. (2000). Naast dat het een goede test voor de methode is om negen taken te modelleren zijn de vaardigheden waar deze taken van gebruik maken cruciaal voor veel van de dagelijkse bezigheden van mensen en worden ze dus veel gebruikt in vele verschillende situaties waardoor ze, in theorie, ideaal geschikt zijn om de methode te testen. Echter, tijdens het creëren van deze modellen liepen we tegen beperkingen in onze methode en in de cognitieve architectuur die we gebruikten. Deze beperkingen zijn veroorzaakt omdat bij het ontwerp van de cognitieve architectuur niet genoeg aandacht is geweest voor het creëren van modellen voor meerdere taken. Dit is een fundamentele uitdaging in het veld van cognitief modelleren. Het doel van dit veld is om een volledig computermodel van cognitie te bouwen met alle eigenschappen van menselijke cognitie. Echter, er is een groot fundamenteel verschil tussen de manier waarop mensen werken en de manier waarop computers werken dat van belang is voor deze uitdaging (er zijn er natuurlijk heel veel). De meeste mensen kunnen uitstekend omgaan met onzekerheid en onnauwkeurigheid, terwijl computers volledig stuklopen tenzij er een heel duidelijk vooropgezet plan is dat gevolgd kan worden. Dit zorgt ervoor dat het heel lastig is voor modelbouwers om een goede balans te vinden tussen flexibiliteit en voorspelbaarheid. Aan de ene kant is het van belang dat een gebouwd model flexibel is en om kan gaan met verscheidene situaties, maar aan de andere kant is het ook van belang dat een model voorspelbaar gedrag vertoont. Wat dit hoofdstuk concludeert is dat het cruciaal is om een duidelijk onderscheid te hebben tussen taak-specifieke (kennis die alleen toepasbaar is voor één situatie) en algemene kennis (kennis die toepasbaar is in meerdere situaties) om een goede balans te krijgen tussen flexibiliteit en voorspelbaarheid. Er is zeer specifieke kennis nodig om een model zich voorspelbaar te laten gedragen, maar aan de andere kant is er ook genoeg algemene kennis nodig om dit op een flexibele manier toe te kunnen passen. Dit hoofdstuk laat zien dat er te weinig aandacht is voor het hergebruik van vaardigheden zelfs in een cognitieve architectuur (PRIMs) die specifiek ontwikkeld is om dit te faciliteren. Ten tweede toont het aan dat er nog veel onbeantwoorde vragen zijn over hoe hergebruik precies kan worden geïmplementeerd in cognitieve architecturen.

In hoofdstuk vijf gaan we dieper in op de problemen en de potentiele oplossingen die we in het vorige hoofdstuk aanstipten. We presenteren oplossingen voor het loskoppelen van taak-specifieke en algemene kennis en testten of dit ervoor zorgde dat het maken van modellen volgens de skill-based methode makkelijker zou worden. Het resultaat van dit hoofdstuk is dat de oplossingen die we presenteerden inderdaad ervoor zorgden dat de skill-based methode beter verliep. Dit wijst erop dat het loskoppelen van taak-specifieke kennis van algemene kennis een cruciale stap is die cognitieve architecturen moeten nemen om herbruikbare vaardigheden te kunnen ondersteunen. Het tweede doel van hoofdstuk vijf was het creëren van een uitgebreid model van executieve functies. Dit model, gebouwd dus met de skill-based methode, verschafte een interessant nieuw inzicht in de studie van executieve functies. Executieve functies verwijzen naar een groep van cognitieve processen die bewust en actief functioneren mogelijk maken. Dit is nodig om taken te kunnen doen waarbij je doelgericht bezig moet zijn en niet kunt vertrouwen op de ‘automatische piloot’. Executieve functies zijn zeer belangrijk en onderzoek heeft aangetoond dat ze cruciaal zijn in bijna alle aspecten van het dagelijks functioneren. Beter executief functioneren is gelinkt aan betere prestaties op school en werk, en slecht functioneren is gelinkt aan een grotere kans op psychische problemen zoals depressies of verslavingen. Omdat deze functies zo belangrijk zijn is er veel aandacht om deze te verbeteren. Denk bijvoorbeeld aan de vele ‘braintrain’ spellen en apps. Het idee van deze interventies is dat het trainen op deze executieve functies de algemene vaardigheid verbetert en dat er na de training beter gepresteerd kan worden in alle gebieden waar deze executieve functie nuttig is. Dit beschouwt de hersenen als een soort spier die getraind kan worden, het trainen van de spier zorgt ervoor dat het beter functioneert in alle gevallen waar deze spier nodig is. Volgens ons model is dit echter niet het geval. Het trainen van een vaardigheid is alleen nuttig als deze exact hetzelfde is als de vaardigheid in het dagelijks leven. Dit is vaak niet het geval omdat de trainingstaak meestal gebaseerd is op het gebruik van een bepaalde strategie die niet makkelijk te vertalen is naar de echte situatie. Hiernaast wordt de daadwerkelijke algemene executieve vaardigheid al zo vaak uitgevoerd in het normale functioneren en is dus goed getraind dat er sowieso al weinig trainingswinst te behalen is. Vaak worden individuele verschillen niet bepaald door verschillen in algemeen executief functioneren maar juist in het handig gebruik maken van strategieën. Dit hoofdstuk laat zien dat het loskoppelen van taak-specifieke en algemene kennis ervoor zorgt dat het hergebruiken van vaardigheden makkelijker wordt. Daarnaast toont het nogmaals aan dat het gebruiken van de skill-based methode kan leiden tot interessante nieuwe inzichten.