Modelling task switching with a skill-based approach

Bachelor's Project Thesis

Renzo Cuadra, s3375404, r.m.cuadra.lazarte@student.rug.nl
Supervisors: C. Hoekstra, PhD & Prof Dr N. A. Taatgen

Abstract: This study explored the feasibility of modelling task switching with a skill-based approach. A model of two switching tasks was built on the cognitive architecture PRIMs: the number-letter task (NL) and the plus-minus task (PM), both of which require alternating between two subtasks. The model followed a skill-based approach whereby procedural knowledge could be maximally reused across tasks. Model feasibility was assessed through two tests: Test 1 evaluated the fit to human response times (RT) in switch and repeat trials; Test 2 analysed skill transfer between tasks. In Test 1, the model replicated human RT and showed switch cost: RT was higher in switch trials than in repeat trials. In Test 2, the model showed a significant decrease in switch cost for both tasks when trained on the opposite task, in line with human performance. Furthermore, there was a greater reduction in switch cost when trained on switch trials than when trained on repeat trials. Since the model showed an adequate fit to human RT and switch cost, as well as near transfer across tasks, modelling task switching with a skill-based approach is considered to be feasible.

1 Introduction

Applying old solutions to new problems is one of the primary tools of cognition to function competently and efficiently in an ever-changing environment. When people face a task they have never encountered before, they begin searching in the pool of available knowledge for cues, experiences and—hopefully—ready-made solutions that may aid in solving the novel problem. These elements are then replicated or recombined and ultimately (re)applied to the new task at hand, in a process known as skill transfer (Carraher and Schliemann, 2002).

In spite of its pivotal role, skill transfer is not often implemented in models of human cognition. In fact, in most cognitive architectures, models are initialised as a blank slate, as if a person came to perform a task with no knowledge or skills other than those needed to initiate the task itself (Taatgen, 2014). This is far from approaching the wealth of experience that a real human subject brings into any task, no matter how novel.

Models of task switching, the executive function responsible for directing attention from one task to another (Miyake, Friedman, Emerson, Witzki, Howerter, and Wager, 2000), typically follow this blank slate approach. Though these models have generally done well at replicating human performance in switching tasks, they cannot demonstrate how a person might be reusing previously learned skills to perform more efficiently. Therefore, the present study examines the feasibility of using skill transfer as the guiding principle to build a cognitive model of task switching. The proposed model follows a skill-based approach, possessing a common set of skills that are reused across different tasks.

A review of skill transfer theory, task switching and advances in cognitive architectures is provided in the following subsections, ending with an overview of the skill-based modelling approach.

1.1 Skill transfer

Skill transfer is the ability to use prior knowledge and experience to solve novel problems (Nokes-Malach and Mestre, 2013). The experience of transfer is hardly foreign to any human being, as we realise from a very young age that recalling past learning is helpful—and even necessary—for acquiring new skills. As we go through the process of learning the four basic arithmetic operations, we might find ourselves drawing from our knowledge of counting numbers to learn addition and subtraction, then using that knowledge of addition to understand multiplication, and ultimately applying the skill of multiplication and subtraction to perform division.

The idea of transfer being so habitual, it has been widely studied in scientific literature for over a century. As early as in 1901, Thorndike and Woodworth proposed the identical elements theory, presenting transfer as the direct application of declarative (factual) or procedural knowledge to novel tasks, which must necessarily share some common features with tasks previously encountered. More contemporary research has found ample evidence of this *near transfer* in multiple domains, such as working memory tasks (Minear, Brasher, Guerrero, Brasher, Moore, and Sukeena, 2016), solving syllogisms (Speelman and Kirsner, 1997) and tasks that involve the same executive control function (Miyake et al., 2000; Karbach and Kray, 2009).

Moreover, studies have shown the occurrence of *far transfer* between tasks that share few or no identical elements. Pennington, Nicolich, and Rahm (1995) found reliable transfer between computer programming tasks involving completely different skills. Bassok and Holyoak (1989) reported that students are able to solve physics problems more effectively when they first practice algebra problems. And studies by Miyake et al. (2000) and Karbach and Kray (2009) show that training one executive control function leads to improved performance on tasks requiring other functions.

Research has also found evidence against transfer, suggesting that the current theoretical account is incomplete. Several studies have exposed systematic failures in people's ability to apply previous learning to new situations, leading researchers to call for a revision of transfer theory (Day and Goldstone, 2012; Nokes-Malach and Mestre, 2013). Carraher and Schliemann (2002) challenged the assumption that transfer involves the passive transport of learning from one situation to another. By investigating how students learn to relate operations on numbers to operations on physical quantities, they found that knowledge is actively reshaped and accommodated depending on the learner's context, rather than being merely replicated. Thus, they coined the term *transfer dilemma*, implying that denying transfer is denying that new learning relies on former learning, but endorsing it as it is currently understood implies subscribing to questionable ideas—in fact, Carraher and Schliemann (2002) propose eliminating the concept of *transfer* altogether.

1.2 Task switching

Miyake et al. (2000) investigated the relationship between the executive control functions of *switching* (the ability to change focus from one task to another in a dynamic context), *updating* (the ability to maintain the most currently relevant information in working memory) and *inhibition* (the ability to block automatic yet inadequate responses). Performance in tasks that involved one function was strongly correlated with performance in other tasks requiring that and other functions, suggesting skill transfer.

Among these three executive functions, task switching is one that has elicited its own stream of research. Karbach and Kray (2009) observed that training in task switching produces near transfer to other similar switching tasks, as well as far transfer to other executive control tasks such as the Stroop task, working memory tests and fluid intelligence tests.

A key measure of performance in task switching is *switch cost*: the difference between the average response time in switching trials and non-switching trials (Purić and Pavlović, 2012). Jersild (1927), founder of the current switching paradigm, proposed switch cost as a measure of the additional effort required to reorient attention to the right task, representing the duration of the control process that reconfigures task sets. This reconfiguration is what causes response time to be higher when the subject must switch between tasks, compared to when they repeat the same task.

Numerous cognitive models of task switching have been built, aiming to replicate typical human response times and switch cost. These include not only computational models (Sohn and Anderson, 2001; Altmann and Gray, 2008; Chuderski, 2017), but also mathematical models (Meiran 2000; Logan and Bundesen 2003; Meiran, Kessler, and Adi-Japha 2008, as cited in Grange and Houghton 2014) and connectionist (neural network) models (Gilbert and Shallice 2002; Brown, Reynolds, and Braver 2007, as cited in Grange and Houghton 2014).

While these modelling efforts have generated valuable insight into task switching, they have been primarily centred around switch cost, with little attention devoted to skill transfer. An opportunity thus arises to bridge this gap, applying recent developments in cognitive architectures to build a model that is able to capture the full scope of task switching dynamics, with a skill-based approach.

1.3 Cognitive architectures

Currently, the predominant platform for building computational models of skill acquisition is ACT-R (Taatgen and Lee, 2003; Anderson, Taatgen, and Byrne, 2005; Salvucci and Taatgen, 2008; Taatgen, Huss, Dickison, and Anderson, 2008). ACT-R (Adaptive Control of Thought-Rational) is a symbolic cognitive architecture where declarative knowledge is stored in *chunks*, whereas procedural knowledge is expressed in *production rules*—if-then statements that perform some actions with chunks under specified conditions (Anderson, 2007). Salvucci (2013), for example, built an ACT-R model that was able to learn new skills by receiving instructions and converting them to production rules, obtaining successful performance in a task that the model had never encountered before.

Building new skills from existing knowledge requires two components: *reuse*, by which knowledge is retrieved in novel contexts, and *integration*, by which components of various skills are combined to form a new skill (Salvucci, 2013). ACT-R incorporates reuse by retrieving existing chunks based on activation, and allows for integration in a process called *production compilation*, by which two production rules that fire together are specialised and combined (Taatgen and Lee, 2003).

Nonetheless, ACT-R falls short of demonstrating skill transfer, facing a particularly stubborn issue with far transfer (Taatgen, 2013). Production rules can only manipulate specific types of chunks, which often limits their applicability beyond the particular task at hand. While chunks can be easily reused across tasks, production rules typically need to be rewritten to be applied in more than one task (Taatgen, 2014). This makes it difficult to shift away from the blank slate approach, meaning that ACT-R models tend to be designed with no knowledge or skills other than those needed for one immediate task. The specificity in production rules also impedes integration, decreasing their capacity to be combined into single production rules to increase efficiency. Therefore, skill reuse and integration may not be sufficiently robust in ACT-R to allow for building models with a skill-based approach.

Taatgen (2013) hence proposes the *primitive information processing elements* (PRIMs) theory, whereby production rules can be broken down further, to the point where they are separated into basic information processing units that are more easily interchangeable between tasks. In this paradigm, cognitive skills are based on only two basic processes: comparing pieces of information, or moving them across the workspace.

The PRIMs cognitive architecture developed by Taatgen (2013) has if-then statements called *operators*—analogous to production rules in ACT-R—which specify some conditions (comparing information) and actions (moving information). This procedural knowledge is encoded in declarative memory, together with facts—unlike ACT-R, which designates a separate procedural module. Operators are grouped into sets to construct a skill, and a skill can contain variables that are instantiated with different elements depending on the task, so it can be reused in multiple situations. Production compilation integrates PRIMs, operators and skills in order to allow for maximal reuse and integration, allowing to compose complex representations from simple elements.

Various skill-based models have been constructed on PRIMs with favourable results in replicating human behaviour and known cognitive phenomena. For example, Hoekstra, Martens, and Taatgen (2020) built a model of the attentional blink effect from skills extracted from models of visual attention and working memory, showing the versatility of the architecture for skill reuse across various tasks. Furthermore, Taatgen has built sucessful models of various tasks, such as arithmetic problems, task switching, Stroop and text editing (2013; 2014).

PRIMs thus presents an attractive alternative for building cognitive models with a skill-based approach, where preexisting skills are directly applied in many different tasks.

1.4 Overview of this study

In order to investigate **whether it is feasible to model task switching with a skill-based approach**, a model of two switching tasks was built on the PRIMs cognitive architecture. The tasks chosen were the number-letter task and the plus-minus task, both of which involve alternating between two subtasks. In the number-letter task (Rogers and Monsell, 1995; Sohn and Anderson, 2001), the participant is shown a stream of digit-letter pairs and is asked to alternate between indicating whether the digit is even or odd, or whether the letter is a consonant or a vowel. In the plus-minus task (Jersild, 1927; Spector and Biederman, 1976; Rubinstein, Meyer, and Evans, 2001), the participant is given a list of two-digit numbers and is requested to alternate between adding or subtracting three from those numbers.

The model was designed with a skill-based approach in mind, maximising skill reuse and integration as a priority. A common set of skills was implemented in order to complete the two tasks successfully, most of which were directly used in both. In addition to switching between subtasks within each task, the model was able to freely switch between both tasks (number-letter and plus-minus) throughout trials. Rather than starting as a blank slate every time it had to switch, the model had sufficient preexisting knowledge to solve all four subtasks across the two tasks.

The feasibility of the model was evaluated based on the fit to human performance in the plus-minus task and the number-letter task, as measured by response times and switch cost, and the ability to show positive skill transfer between tasks.

Demonstrating the feasibility of modelling cognitive tasks with a skill-based approach may contribute to building better, more realistic models of cognition which closely approach the way skills are formed, strengthened and transferred in the brain. This is a step forward from current standard models, which need to be programmed and trained anew every time a new task has to be performed. In the long term, skill-based cognitive models may bring us closer to developing more comprehensive intelligent systems with a wide ranging skill set.

2 Method

The PRIMs model of task switching was designed to perform the number-letter (NL) task (Sohn and Anderson, 2001) and the plus-minus (PM) task (Spector and Biederman, 1976; Rubinstein et al., 2001). In the NL task, a digit-letter pair is shown and the subtasks are to indicate (a) whether the digit is even or odd (*number subtask*), or (b) whether the letter is a consonant or a vowel (*letter subtask*). In the PM task, a two-digit number is shown and the subtasks are to report the result of either (a) adding three to that number (*plus subtask*) or (b) subtracting three from that number (*minus subtask*).

The model follows the experimental paradigm used by Sohn and Anderson (2001) on human participants. In this setup, each trial comprises of two subtasks, which can either be the same (such as plus-plus) or different (such as letter-number)——see Table 2.1 offers a list of all subtask combinations. Trials where both subtasks are equal are called *repeat trials*, whereas trials where subtasks are different are called *switch trials*. Trials are organised in blocks comprising of either repeat or switch trials exclusively. Participants are told, at the start of every block, whether they will have to switch or repeat (Sohn and Anderson (2001) refer to this as the foreknowledge condition), which means they can prepare in advance for subtask 2. Both the NL and PM tasks are implemented in this way in the model, even though Spector and Biederman (1976) and Rubinstein et al. (2001) used a different experimental approach in their studies with the PM task.

A trial began by showing a fixation dot on the

Table 2.1: Possible subtask combinations in one trial

Trial	Subtask 1	Subtask 2
NL repeat	number letter	number letter
NL switch	number letter	letter number
PM repeat	plus minus	plus minus
PM switch	plus minus	minus plus

screen for 1 second. Then, the stimulus for subtask 1 was presented. For the NL task, the stimulus was a pair of one digit and one letter, presented in random order (digit first or letter first). Letters were taken from the set [A,E,I,U,G,K,M,R] and digits were taken from the set [2,3,4,5,6,7,8,9]. For the PM task, the stimulus was a two-digit number between 10 and 99 (inclusive). The stimuli were accompanied by a cue indicating which subtask should be performed. In the NL task, the cue was the colour in which the number-letter pair was shown: red for the number subtask and green for the letter subtask. In the PM task, the subtask was indicated by a plus or minus sign. In the model, these cues were presented as a string ('red', 'green' in the NL task; 'plus-sign', 'minus-sign' in the PM task).

The stimulus was shown until an answer was provided. In the case of the NL task, the left key had to be pressed if the digit was even or the letter was a consonant, and the right key had to be pressed if the digit was odd or the letter was a vowel (depending on whether the number or letter subtask was being performed). In the PM task, the result of the addition or subtraction operation had to be provided. After a response was given, a blank screen was shown for 0.6 seconds. Then, the stimulus for subtask 2 was shown until an answer was provided. The trial ended when the response for subtask 2 was given. A trial was considered correct when the appropriate responses for both subtasks had been provided.

Two tests were performed in order to assess the feasibility of the PRIMs model of task-switching. In Test 1, the model's ability to replicate human performance was evaluated, in terms of RT and switch cost for the NL and PM task individually. The model's mean RT and switch cost were compared to those obtained in studies with human participants: Sohn and Anderson (2001) for the NL task, and Spector and Biederman (1976) and Rubinstein et al. (2001) for the PM task. These two studies were considered for the PM task because they report very similar switch costs, even though they obtained different response times.

In Test 2, skill transfer between the NL and PM tasks was analysed. Following research by Karbach and Kray (2009), a decrease in switch cost is expected to occur as a result of training on the opposite task. In the model, this general reduction can be attributed to continuously reusing a set of common skills to perform the NL and PM tasks during training. Karbach and Kray (2009) also found that the decrease in switch cost is greater when training on switch trials than when training on repeat trials. The model was implemented with two different skills for switching and for repeating subtasks within a trial. Repetitive use of the switching skill while training on switch trials is expected to generate this further reduction in switch cost, compared with training on repeat trials.

2.1 Model implementation

The model relies on six skills to perform the given tasks. Four of them, which control task management, are reused across both tasks. Another one is responsible for solving the PM task and a sixth one is dedicated to handling the NL task. Table 2.2 summarises the six skills used in the model, while Figure 2.1 provides illustrates the tasks and their respective skills and operators in a diagram.

In order to perform all necessary processes for both tasks, the model's declarative memory was populated with several types of chunks. Table 2.3 presents a summary of these chunks.

The `set-task` skill allows the model to identify the subtask to be performed by looking at the stimulus. It fires at the start of every trial, when the subtask is unknown. First, the model looks at the fixation dot and waits for the stimulus to appear. When the stimulus is shown, the model reads the cue string, recalls the stimulus-task (S-T) map-

Table 2.2: Overview of skills

Skill	Description
`set-task`	Identify the subtask by looking at the given cue
`number-letter`	Find the correct response for the NL task by retrieving class mappings
`plus-minus`	Find the result of addition or subtraction by direct retrieval or calculation
`answer`	Respond and clear buffers
`repeat`	Keep subtask 1 as a goal for subtask 2
`switch`	Update goal for subtask 2

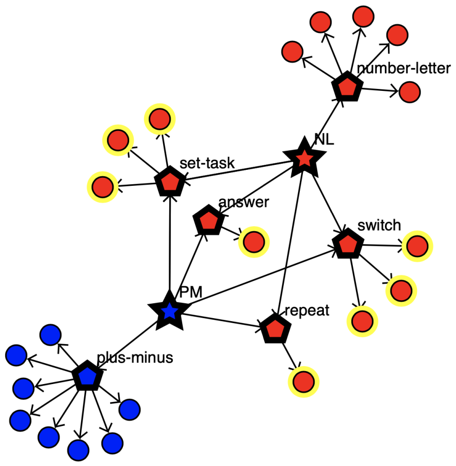

Figure 2.1: Diagram of the operators and skills used in the model. Stars represent tasks, pentagons represent skills and circles represent operators. Operators shared by both tasks are shown with a yellow halo.

ping for that cue, saves the corresponding subtask in working memory and then places it in the goal buffer.

Then, the skill relevant to the goal subtask is executed. Using PRIMs' capacity for instantiation, the `number-letter` and `plus-minus` skills contain variables that are instantiated to specific values, depending on the subtask at hand. That allows the model to reuse these two skills to complete all four subtasks.

The `number-letter` skill finds the correct response for the NL task by recalling class mappings. First, the model decides whether it should focus on the digit or the letter, as indicated by the current goal. Then, it looks at the character on the left of the screen. If it matches the focus object, it is held in working memory and mapped to its class (for instance, 'A-letter-vowel'). Otherwise, the model maps the character on the right. The stimulus-response (S-R) mapping is then retrieved for the encoded character (for example, 'vowel-right') and kept in working memory.

The `plus-minus` skill formulates the result of addition or subtraction for the PM task, which the model can do in two different ways: by retrieving the answer directly from memory or by performing a calculation. This follows research on the cognitive processing of arithmetic problems, where participants commonly report using a mix of direct retrieval and various calculation strategies for one and two-digit addition and subtraction (Campbell and Xue, 2001; Lemaire and Arnaud, 2008). The proportion of direct retrieval and calculation, as well as the exact calculation strategy, can vary greatly depending on factors such as age and cultural background. Therefore, the model was built to perform direct retrieval approximately 50% of the time and calculation the other 50%. When the model initialises, declarative memory is populated with a set of addition and subtraction facts that provide the result of adding and subtracting three from various two-digit numbers. The numbers for these facts are sampled randomly without replacement from the full set of integers from 10 to 99.

The direct retrieval and calculation processes rely on a different set of operators within the `plus-minus` skill. The operators responsible for the retrieval strategy always fire first, attempting to retrieve the answer to the operation at hand from declarative memory. This follows the assumption that a human participant would try to recall the result of a simple operation before doing any calculation, as a time saving strategy. If the answer is found, it is saved in working memory. If not, a chain of operators fires that carries out the calculation process. The procedure involves the following steps: decomposing the two-digit number into units and tens, adding or subtracting three from the units, checking whether there is a remainder that should be carried over to increase or decrease the tens, and finally adding the resulting units and tens to compose the answer. This result is then saved in working memory.

The `answer` skill provides the response and clears the workspace, both for the NL and PM task. It fires when the model has saved the final result for a subtask in working memory.

After providing an answer, the screen goes blank and the model must transition to subtask 2. Two skills can be used to handle this transition: `repeat` or `prepare`, depending on the type of block.

The `repeat` skill ensures that subtask 1 is again set as the goal for subtask 2 in repeat trials. The model keeps recalls subtask 1 from working memory, puts it in the goal buffer and waits for the next stimulus to appear on the screen. It then applies the same instantiated skill used for solving subtask 1 in

Table 2.3: Chunks in declarative memory

Chunk type	Description
Number-letter task	
S-T mappings	Pairs of subtask cues (red/green) and their corresponding subtasks
Subtask mappings	Pairs of opposing subtasks (N-L/L-N)
Class mappings	Characters with corresponding classes
S-R mappings	Character classes with corresponding keys
Plus-minus task	
S-T mappings	Pairs of subtask cues (+/-) and their corresponding subtasks
Subtask mappings	Pairs of opposing subtasks (+-/-+)
Addition & subtraction facts	Results for direct retrieval
Decompositions	Two-digit numbers broken down into units and tens
Remainders	Results from carrying over remainders from units to tens
Compositions	Units and tens added into two-digit numbers

subtask 2.

The `switch` skill, on the other hand, updates the goal for subtask 2. This goal update requires the model to take additional steps, which results in switch cost. Two different strategies can be used to transition to the new subtask: the model may simply wait for the second stimulus to be shown and then encode the new subtask cue or, in a more proactive strategy, it may update its goal for subtask 2 by recalling the opposite of subtask 1.

Whether the model applies the proactive or reactive strategy depends on the activation of two operators within the `switch` skill: `forget` and `prepare`. If the blank screen is shown and the `forget` operator is more highly activated, the model clears the goal buffer and waits for the second stimulus to appear. It then fires the `set-task` skill to encode the cue for subtask 2 as a new goal, repeating the procedure used in subtask 1. This reactive strategy simulates a situation where the subject does not know what the next subtask will be and therefore decides to wait in order to ensure it performs the correct one.

If, on the other hand, the `prepare` operator has a higher activation, the proactive strategy is enforced. The model performs a memory retrieval to identify the opposite of whatever subtask was carried out first, which it places in the goal buffer. Note that the `prepare` operator can only fire while the screen is blank, so if it takes longer than the inter-stimulus interval, the preparation fails and the model is forced to use the reactive strategy instead. Figure 2.2 provide a schematic depiction of this implementation.

This dual approach, based on a competition between a reactive and a proactive strategy, has been applied in previous research. Sohn and Anderson's (2001) ACT-R model, built on this principle, was able to replicate human response time (RT) and switch cost with various inter-stimulus intervals. Also using this approach, Taatgen (2013) successfully modelled skill transfer between switching tasks, whereby training on switch trials led to a greater decrease in switch cost in posterior test trials, compared to training on repeat trials. In the model built for this study, this effect is expected to derive from reusing the `switch` skill while training on switch trials, which increases activation of the skill's operators and therefore reduces firing latency. In contrast, reusing the `repeat` skill while

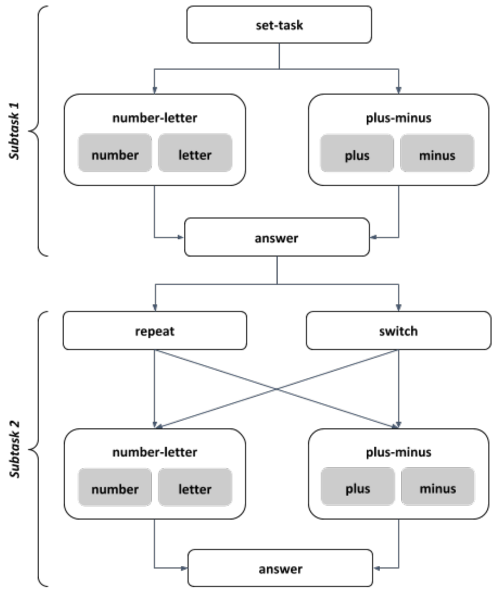

Figure 2.2: Depiction of the skill execution process in one trial. The gray boxes represent the possible instantiations of a certain skill.

training on repeat trials does not produce this activation benefit.

2.2 Parameter settings

Model parameters were tuned for purposes of fitting the model to human data (see Table 2.4), by promoting learning through production compilation and regulating chunk activation. In PRIMs, facts and operators are retrieved or fired based on their activation value (Taatgen, 2019). When the activation of a chunk is above the retrieval threshold, it can be successfully retrieved. Activation is calculated with equation 2.2, where B_i is the chunk's base level activation, S_{ji} is the strength of association between the chunk and buffer slots, W_k is spreading activation from buffers, S_{ki} is the strength of association between the chunk and skills and A_k is spreading activation from skills. Thus, a chunk receives spreading activation from buffers, slots and skills it is associated with.

$$A_i = B_i + \sum_{k}^{buffers} \sum_{j}^{slots} S_{ji} W_k + \sum_{k}^{skills} S_{ki} A_k \quad (2.1)$$

The time it takes for a chunk to be retrieved or an operator to be fired (latency) depends on activation, as indicated by the time for retrieval equation below, where A_i is the chunk's activation and F is the latency factor (a global parameter) (Taatgen, 2019).

$$t_{retrieval} = Fe^{-A_i} \quad (2.2)$$

Default activation is the parameter that determines the lower bound in base level activation B_i for all chunks (including operators) (Taatgen, 2019). Increasing default activation (default = 0.0) means that retrieval failure is less likely and that retrieval or firing latency for all chunks will be lower.

Default operator association controls the degree of spreading activation (S_{ki}) between a skill and the operators it contains (Taatgen, 2019). This parameter is set higher than the 4.0 default (6.0 for the NL task and 10.0 for the PM task), making it less likely that the model will fire operators belonging to skills that are not currently in the goal buffer.

Production compilation occurs as the model learns new production rules that allow it to fire several PRIMs at once for one operator (Taatgen, 2019). When rules are fully compiled, all operators can be fired in approximately equal time, regardless of their size. The learning rate (α) controls the speed of production compilation. While the default value for α is 0.1, the model uses a learning rate of 0.05 for the NL task and 0.07 for the PM task. This slows down production compilation, accentuating differences in firing latency, which is desirable for reproducing the effects of switch cost and skill transfer.

3 Results

3.1 Test 1

The model was run 20 times for 16 blocks of 25 trials. Each task (NL and PM) was tested independently—the model never performed both of them

Table 2.4: Non-default parameter values used

Parameter	NL	PM
Learning rate (α)	0.05	0.07
Default operator association	6.0	10.0
Default activation	1.5	0.0

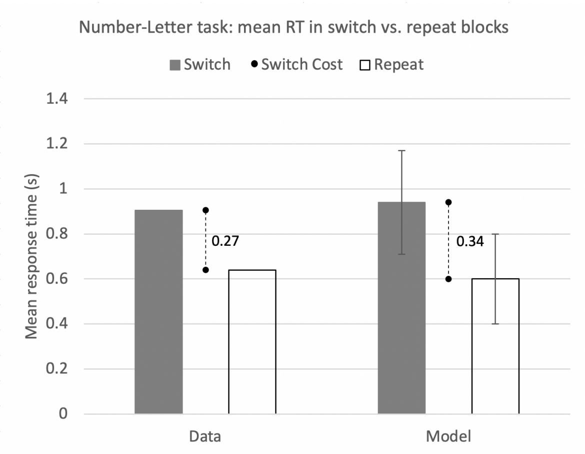

Figure 3.1: Mean RT and switch cost in the NL task. Data from Sohn and Anderson (2001). Error bars represent standard deviation.

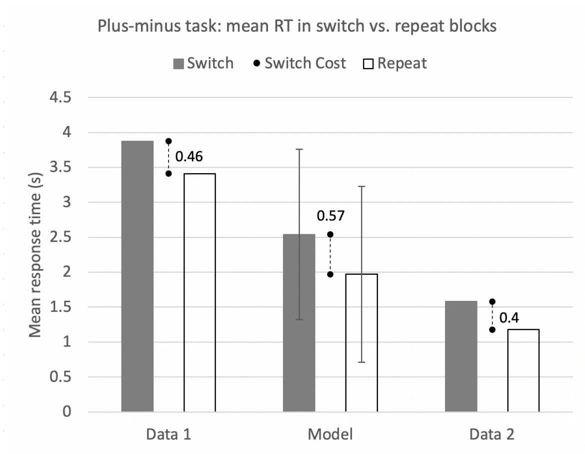

Figure 3.2: Mean RT and switch cost in the PM task. Data 1 from Rubinstein et al. (2001) and data 2 from Spector and Biederman (1976). Error bars represent standard deviation.

in one run. Blocks alternated between switch and repeat, so no consecutive blocks were the same.

The first eight blocks were practice and their data were excluded. After these eight blocks, RT had been driven to an acceptable range by production compilation. This is typical of PRIMs models, which need to perform a number of trials before activation and utility values stabilise. Data from the following eight blocks are reported, which is four repeat blocks and four switch blocks, totalling 200 trials in each run.

Model fit to human performance was assessed in terms of RT and switch cost. RT was measured as the time elapsed between the stimulus presentation and the model's response for subtask 2 in correct trials, averaged over repeat blocks and switch blocks separately, over all runs. Switch cost was measured as the difference between mean RT in switch blocks and mean RT in repeat blocks, averaged over all runs. Null values from trials with incorrect responses (17 for the PM task and none for the NL task) were replaced with mean RT from the same run and type of block, so as to equate the number of trials of each type of block for analysis.

Results for the NL task are presented in Figure 3.1. The model closely approaches RT from participants in Sohn & Anderson's (2001) experiment, with a slightly larger difference between switch and repeat blocks. Switch blocks take an additional $0.34s$ on average to provide a response than repeat blocks. A paired Wilcoxon signed-rank test shows that the difference in RT between switch blocks ($M = 0.94s, SD = 0.23s$) and repeat blocks ($M = 0.6s, SD = 0.2s$) is statistically significant, $Z = -35.38, p < .0001, r = .79$.

With respect to the PM task, the model output is compared to data from human participants in experiments by Rubinstein et al. (2001) and Spector and Biederman (1976). Both studies were considered since they report very different RT, yet they show similar switch cost ($0.43s$ and $0.4s$ respectively). Both presented stimuli in physical form and only measured total completion time for a full block of trials. RTs from these studies is presented in Figure 3.2 as the mean completion time for one block, divided by the number of trials in a block. The large RT difference between the two is due to their methods: Rubinstein et al. (2001) used printed cards, where flipping each card manually added to completion time, whereas Spector and Biederman (1976) used a list of numbers printed on paper.

Model RT is located at an intermediate level between these two studies. The $0.57s$ difference in RT between switch blocks and repeat blocks is slightly over the average switch cost found by Rubinstein et al. (2001) and Spector and Biederman (1976). This is likely due to the low learning rate used in the model, which leads to larger differences in latency. A paired Wilcoxon signed-rank test shows that the difference in RT between switch blocks ($M = 2.54s, SD = 1.22s$) and repeat blocks ($M = 1.97s, SD = 1.26s$) is statistically significant, $Z = -13.16, p < .0001, r = .29$.

The model is thus successful in replicating human performance in terms of RT and switch cost, both for the NL and the PM task. In both cases, switch cost is statistically significant and slightly larger than that observed in human participants.

3.2 Test 2

A pretest-training-posttest approach was used to evaluate whether training on one task (for example, PM) would lead to decreased switch cost in a different task (NL), indicating skill transfer. The pretest and posttest were identical: they included 16 blocks of 25 trials, alternating between switch and repeat blocks. The first eight were practice blocks, whose data are not included. In between pretest and posttest, the model was trained on 40 blocks of 25 trials of the opposite task, either switch or repeat exclusively, for a total of 1,000 trials.

Transfer from the PM task to the NL task and from the NL task to the PM task were both assessed via a two-by-two factorial design, with training type (switch/repeat) as a between-subject variable and testing time (pretest/posttest) as a within-subject variable. The model was run 20 times for each training condition and task. RT was recorded for correct subtask 2 trials, at pretest and posttest, averaged over all runs in each training condition. Null values (29 for the PM task and 23 for the NL task) were again replaced with mean RT from the same run, type of block and test (pre/post) in order to equate the number of trials for analysis. Transfer was measured by comparing switch cost as the difference in mean RT between switch and repeat trials across each condition and task.

Figures 3.3 and 3.4 present results for the NL and PM task, respectively. Mean switch cost found in a similar experiment by Karbach and Kray (2009) is shown as a reference. In their study, at pretest and posttest participants indicated whether a picture showed a fruit or a vegetable or whether the picture was small or large (food-size task). In repeat training, they performed repeat trials of the food or size task, whereas in switch training they performed a transportation-number task where they had to decide whether the pictures showed planes or cars, or whether one or two planes/cars were presented.

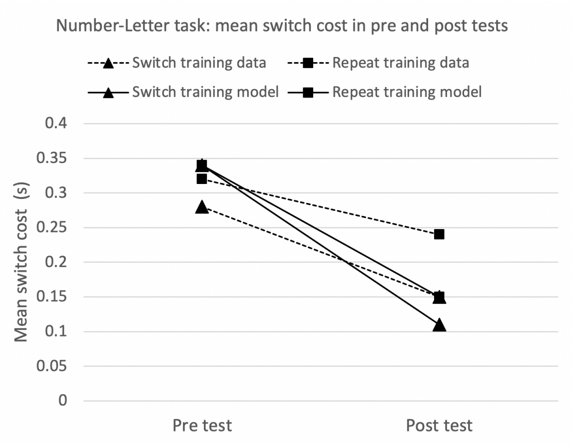

Figure 3.3: Mean switch cost in pretest and posttest per training condition in the NL task. Data from Karbach and Kray (2009).

Although a different task set was used, data from this study clearly show the effects of skill transfer as a decrease in switch cost from pretest to posttest, which is more pronounced when subjects are trained on switch trials than on repeat trials. Since Karbach and Kray (2009) found no significant difference in switch cost across age groups, the figures show the aggregated means.

In the NL task, training on switch trials of the PM task led to a notable decrease in switch cost from pretest ($M = 0.34s$) to posttest ($M = 0.11s$), while training on repeat PM trials produced a slightly smaller reduction in switch cost from pretest ($M = 0.34s$) to posttest ($M = 0.15s$). A two-way mixed ANOVA was performed to test for the effects of testing time and training type on switch cost. A statistically significant effect was found for testing time ($F(1, 38) = 909.25, p < .0001, \eta^2_G = .91$), as well as a significant two-way interaction between testing time and training type ($F(1, 38) = 7.26, p = .01, \eta^2_G = .07$). No significant effect of training type was found. Post hoc one-way ANOVAs revealed a significant effect of training type on switch cost at posttest ($F(1, 38) = 14, p = .001, \eta^2_G = .27$), but not at pretest. On the other hand, testing time was found to have a significant effect both for the switch training group ($F(1, 19) = 427, p < .0001, \eta^2_G = .92$) and for the repeat training group ($F(1, 19) = 511, p <$

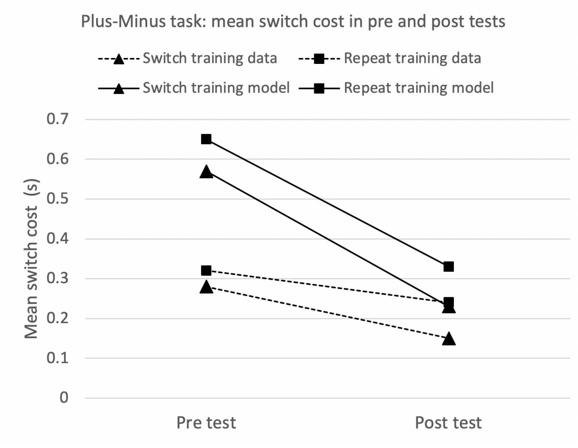

Figure 3.4: Mean switch cost in pretest and posttest per training condition in the PM task. Data from Karbach and Kray (2009).

$.0001, \eta_G^2 = .9$).

With regard to the PM task, training on switch trials of the NL task led to a considerable reduction in switch cost from pretest ($M = 0.57s$) to posttest ($M = 0.23s$). Training on repeat NL trials also elicited a reduction in switch cost from pretest ($M = 0.65s$) to posttest ($M = 0.33s$), albeit less pronounced than in the switch training group. A two-way mixed ANOVA showed a statistically significant effect of testing time ($F(1, 38) = 68.01, p < .0001, \eta_G^2 = .48$) as well as training type ($F(1, 38) = 5.4, p < .05, \eta_G^2 = .06$), with no significant two-way interaction between training type and testing time.

Hence, results from Test 2 are similar for both transfer cases. In general, switch cost was significantly decreased (by over 50%) from pretest to posttest, by training on the opposite task. Depending on the type of training, this reduction was relatively different in size: training on switch trials led to a somewhat larger decrease in switch cost than training on repeat trials.

4 Discussion

This study investigated the feasibility of modelling task switching with a skill-based approach, for which a model of two switching tasks was built on the PRIMs cognitive architecture: the number-letter (NL) task and the plus-minus (PM) task. Feasibility was evaluated based on two criteria: the fit to human RT and switch cost in each task (Test 1), and the ability to show skill transfer between tasks, by decreasing switch cost after training on the opposite task (Test 2).

Test 1 showed that the model closely approaches human performance in terms of RT and switch cost, both for the NL and the PM task, with switch cost being slightly over that of human participants. In Test 2, a significant decrease in switch cost from pretest to posttest was found for both tasks, when trained on the opposite task. For the NL task trained on the PM task, the model showed a greater reduction in switch cost when trained on switch trials than when trained on repeat trials. This additional switch training reduction did not occur for the PM task trained on the NL task.

The adequate fit of the model to human RT and switch cost suggests that the skill-based approach, whereby general skills can be directly reused across tasks, renders feasible models of task switching. PRIMs facilitates this skill-based approach, allowing for greater generalisation of procedural knowledge than ACT-R and other cognitive architectures where tasks must be implemented with very specific production rules, impeding skill reuse. Thus, it provides a partial solution for the blank slate problem, as it comes one step closer to the way human cognition imports knowledge from previous experiences (Taatgen, 2014). It is important to stress that the solution is partial, as PRIMs is still far from approaching the wealth of skills that a person may accumulate throughout life. However, it sets a starting point for what in the future could be a cognitive architecture that possesses a large storage of knowledge that can be easily reapplied in new tasks.

The replication of switch cost in the model supports the proposal that switching involves taking extra cognitive steps in order to update goals, which need not be taken when the same task is repeated, as theorised by Jersild (1927). In the model, switching requires either recalling the opposite task (proactive strategy) or waiting for the external cue in order to update the goal (reactive strategy), both of which involve many operators and memory retrievals. Repeating, however, can be done with only one operator and one retrieval.

The model is therefore able to reproduce the lack of preparation component of switch cost, proposed by Sohn and Anderson (2001). Lack of preparation is enforced by the reactive strategy, which forces the model to re-encode the subtask when it has to switch. Meanwhile, when repeating is due, the goal set in the previous subtask is directly reused—hence, the model is always prepared in repeat trials. Note that, through production compilation, the model eventually learns to prepare for subtask 2 within the inter-stimulus interval. If it only used the proactive strategy, provided that the inter-stimulus interval is long enough, it would ultimately be prepared at all times when the stimulus appears, showing no difference in RT compared to repeat trials. Therefore, the reactive strategy is key for switch cost.

In general, the model showed positive transfer from the PM task to the NL task and vice versa, as measured by switch cost decrease. The primary source of this transfer was the reuse of a set of common skills, which allowed the model to strengthen their activation and perform production compilation even while training on a different task, thus reducing overall latency. This is in line with findings from Karbach and Kray (2009), who reported an overall decrease in switch cost when human participants were trained on one type of switching task and then tested on another switching task. In agreement with Thorndike and Woodworth's (1901) identical elements theory, transfer in the model is realised by the direct application of procedural knowledge in multiple tasks, so this can be considered a case of near transfer.

Also important to reducing switch cost was training the `switch` skill, which was favoured when the model was trained exclusively on switch trials. As reported in Karbach and Kray (2009), it was found that switch training led to a stronger effect of transfer than repeat training. Sohn and Anderson (2001) and Taatgen (2013, 2016) attribute this reduction in switch cost to the fact that training on switch trials promotes the application of the proactive switching strategy, which is more time-efficient than the reactive strategy. However, this reinforcement of the proactive strategy did not occur in the model, since the operators for both strategies were contained within the same skill. That means they spread activation to each other as they fired, creating an approximately equal distribution throughout trials.

Still, training the model on switch trials led to a greater reduction in switch cost than repeat training, suggesting that reinforcing proactive switching may not be the only factor contributing to reducing switch cost. Production compilation may have therefore played a central role in facilitating transfer based on the switching skill. Switch training may have promoted compilation of the `prepare` operator production rules for posttest, thus ensuring that it could fire before the inter-stimulus interval ended. Considering that the learning rate was low, full compilation of the `prepare` operator rules may not have been possible with repeat training, occasionally leading to failure of the proactive strategy. The model thus shows that production compilation alone can suffice to reproduce a visible differential effect of switch training with respect to repeat training. Even though that effect is modest, it serves to demonstrate how practicing switching as an executive control function could, in and of itself, lead to a significant decrease in switch cost, with no need to account for a specific switching mechanism.

The limitations of the study are presented hereafter. It should first be noted that the feasibility of the model was assessed by comparing its output to data from other studies with human participants, which often used a different experimental design and a different task set—the PM task, especially, was implemented very differently in studies with human participants. Also, their full data set was not available, which meant that no statistical analysis was possible. Comparisons with primary data could have provided greater validity and insight, though this was not possible at the time of the study.

The investigation was based on two tasks with similar characteristics (namely, they required switching) but greatly different complexity, as the PM task requires several more operators and memory retrievals than the NL task. Also, the model could alternate between direct retrieval and calculation for solving the PM task, which caused RT to be more widely spread. This could have biased the analysis of skill transfer in particular, which is why considering tasks with equal cognitive load, and repeating the analyses with more than two tasks, would contribute to increasing validity.

Lastly, PRIMs is a young cognitive architecture with little previous research. Information on how the architecture works is limited and few models have been made on PRIMs thus far (compared to more established architectures such as ACT-R). There was a scarcity of examples to guide the research and many of the platform's details had to be learnt by experimentation, which prolonged development time and could have made the model prone to error.

These observations invite potential topics for future research. Initially, the task switching model should be validated by collecting data from human participants with the same experimental design and tasks. This would allow to perform statistical analysis to compare model output to human data, providing a way to directly test the model.

Feasibility of the model can also be assessed with the same experimental design, but using different tasks with various levels of complexity. Thus, the model's ability to fit human performance and to show near and far transfer could be further validated. The new tasks would have to be implemented with the same skill-based approach, maximising skill reuse across tasks.

The first step in this line of research would be to test the current model with two tasks of equivalently low complexity, such as the number-letter task and the food-size task used by Karbach and Kray (2009), to examine whether an adequate fit to RT and switch cost are maintained, and whether the skill transfer effect on switch cost can be replicated. Going further, the model could be extended to examine far transfer between switching tasks and tasks that involve different executive functions, such as the Stroop task, which is based on inhibition. Miyake et al. (2000), who study correlations between tasks that tap on various executive functions, may provide a blueprint for this research.

Other dynamics of task switching, besides switch cost and skill transfer, could also be studied with this modelling approach. For example, Sohn and Anderson (2001) found that shorter inter-stimulus intervals produced higher switch cost than longer intervals, which they attributed to subjects not having enough time to prepare for subtask 2 before the second stimulus appeared, forcing them to use a reactive strategy. The current model can easily be used to test this by adapting its experimental design. Also, Taatgen (2016) built PRIMs models of various tasks which were able to replicate the effects of diminishing returns in training, as well as transfer between tasks that require skills at progressive stages of development. Attempting to show these effects with the task switching model could provide an interesting perspective of the versatility of the skill-based approach.

The model, designed with a skill-based approach where skill reuse across tasks is maximised, provided an adequate fit to human RT and switch cost for the NL and PM tasks. It also showed near transfer across these tasks through switch cost reduction, with an additional decrease from training on switch trials. Since the model is able to match human performance in switching tasks, it can be concluded that modelling task switching with a skill-based approach is feasible.

Moreover, the study adds to an incipient body of research on PRIMs, validating it as a cognitive architecture with potential to provide a more realistic account of human cognition, which is characterised by frequent reuse and adaptation of existing knowledge. The hope is that more researchers will be encouraged to develop skill-based models that, in the long term, may set the ground for creating more comprehensive intelligent systems with a rich set of skills.

References

E. M. Altmann and W. D. Gray. An integrated model of cognitive control in task switching. *Psychological review*, 115(3):602, 2008.

J. R. Anderson. *How can the human mind occur in the physical universe?* Oxford University Press, 2007.

J. R. Anderson, N. A. Taatgen, and M. D. Byrne. Learning to achieve perfect timesharing: Architectural implications of hazeltine, teague, and ivry (2002). 2005.

M. Bassok and K. J. Holyoak. Interdomain transfer between isomorphic topics in algebra and physics. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(1):153, 1989.

J. W. Brown, J. R. Reynolds, and T. S. Braver. A computational model of fractionated conflict-control mechanisms in task-switching. *Cognitive psychology*, 55(1):37–85, 2007.

J. I. Campbell and Q. Xue. Cognitive arithmetic across cultures. *Journal of Experimental Psychology: General*, 130(2):299, 2001.

D. Carraher and A. Schliemann. The transfer dilemma. *Journal of The Learning Sciences - J LEARN SCI*, 11:1–24, 01 2002. doi: 10.1207/S15327809JLS1101_1.

A. Chuderski. A model of flexible control in task switching. In *Proceedings of the European Cognitive Science Conference 2007*. Taylor & Francis, 2017.

S. Day and R. Goldstone. The import of knowledge export: Connecting findings and theories of transfer of learning. *Educational Psychologist - EDUC PSYCHOL*, 47:153–176, 07 2012. doi: 10.1080/00461520.2012.696438.

S. J. Gilbert and T. Shallice. Task switching: A pdp model. *Cognitive psychology*, 44(3):297–337, 2002.

J. Grange and G. Houghton. *Models of Cognitive Control in Task Switching*, pages 160–199. 01 2014. ISBN 978-0-19-992195-9. doi: 10.1093/acprof:osobl/9780199921959.003.0008.

C. Hoekstra, S. Martens, and N. A. Taatgen. A skill-based approach to modeling the attentional blink. *Topics in Cognitive Science*, 2020.

A. T. Jersild. Mental set and shift. *Archives of psychology*, 1927.

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

P. Lemaire and L. Arnaud. Young and older adults' strategies in complex arithmetic. *The American Journal of Psychology*, 121(1):1–16, 2008. ISSN 00029556. URL `http://www.jstor.org/stable/20445440`.

G. D. Logan and C. Bundesen. Clever homunculus: Is there an endogenous act of control in the explicit task-cuing procedure? *Journal of Experimental Psychology: Human Perception and Performance*, 29(3):575, 2003.

N. Meiran. Modeling cognitive control in task-switching. *Psychological research*, 63(3-4):234–249, 2000.

N. Meiran, Y. Kessler, and E. Adi-Japha. Control by action representation and input selection (caris): A theoretical framework for task switching. *Psychological Research*, 72(5):473–500, 2008.

M. Minear, F. Brasher, C. B. Guerrero, M. Brasher, A. Moore, and J. Sukeena. A simultaneous examination of two forms of working memory training: Evidence for near transfer only. *Memory & Cognition*, 44(7):1014–1037, 2016.

A. Miyake, N. P. Friedman, M. J. Emerson, A. H. Witzki, A. Howerter, and T. D. Wager. 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, 2000.

T. J. Nokes-Malach and J. P. Mestre. Toward a model of transfer as sense-making. *Educational Psychologist*, 48(3):184–207, 2013.

N. Pennington, R. Nicolich, and J. Rahm. Transfer of training between cognitive subskills: Is knowledge use specific? *Cognitive psychology*, 28(2): 175–224, 1995.

D. Purić and M. Pavlović. Executive function of shifting: Factorial structure and relations to personality and intelligence domains. *Suvremena psihologija*, 15(2):191–191, 2012.

R. Rogers and S. Monsell. Costs of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: General*, 124:207–231, 06 1995. doi: 10.1037/0096-3445.124.2.207.

J. S. Rubinstein, D. E. Meyer, and J. E. Evans. Executive control of cognitive processes in task switching. *Journal of experimental psychology: human perception and performance*, 27(4):763, 2001.

D. D. Salvucci. Integration and reuse in cognitive skill acquisition. *Cognitive Science*, 37(5):829–860, 2013.

D. D. Salvucci and N. A. Taatgen. Threaded cognition: An integrated theory of concurrent multitasking. *Psychological review*, 115(1):101, 2008.

M. H. Sohn and J. R. Anderson. Task preparation and task repetition: Two-component model of task switching. *Journal of Experimental Psychology: General*, 130(4):764, 2001.

A. Spector and I. Biederman. Mental set and mental shift revisited. *The American Journal of Psychology*, pages 669–679, 1976.

C. P. Speelman and K. Kirsner. The specificity of skill acquisition and transfer. *Australian Journal of Psychology*, 49(2):91–100, 1997.

N. A. Taatgen. The nature and transfer of cognitive skills. *Psychological review*, 120:439–471, 06 2013. doi: 10.1037/a0033138.

N. A. Taatgen. Between architecture and model: Strategies for cognitive control. *Biologically Inspired Cognitive Architectures*, 8:132–139, 2014.

N. A. Taatgen. *Theoretical Models of Training and Transfer Effects*, pages 19–29. Springer International Publishing, Cham, 2016. ISBN 978-3-319-42662-4. doi: 10.1007/978-3-319-42662-$4_3.URL$