A Model of Distraction using new Architectural Mechanisms to Manage Multiple Goals

Taatgen, Niels; Katidioti, Ioanna; Borst, Jelmer; van Vugt, Marieke

Published in:
Proceedings of the 13th International Conference on Cognitive Modeling

A Model of Distraction using new Architectural Mechanisms to Manage Multiple Goals

Niels A. Taatgen (n.a.taatgen@rug.nl), Ioanna Katidioti (i.katidioti@rug.nl), Jelmer Borst (j.p.borst@rug.nl) & Marieke van Vugt (m.k.van.vugt@rug.nl)
Institute of Artificial Intelligence, University of Groningen,
Nijenborgh 9, 9747 AG Groningen, Netherlands

Abstract

Cognitive models assume a one-to-one correspondence between task and goals. We argue that modeling a task by combining multiple goals has several advantages: a task can be constructed from components that are reused from other tasks, and it enables modeling thought processes that compete with or support regular task performance. To achieve this, we updated the PRIMs architecture (a derivative of ACT-R) with the capacity for parallel goals that have different activation levels. We use this extension to model visual distraction in two experiments. The model provides explanations for the finding that distraction increases with task difficulty in a memory task, but decreases with task difficulty in a visual search task.

Keywords: Cognitive Control, Task representation, PRIMs, Distraction, Multitasking

Introduction

Whenever we are faced with something new to learn or to do, we can rely on a vast array of skills and knowledge. Given what we usually call a *task*, we need to recruit the right procedural and declarative knowledge and find the best way to piece this knowledge together, and, if necessary expand it with the missing pieces. One might think this challenge should be a centerpiece in the cognitive modeling and cognitive architecture research tradition, but unfortunately it isn't.

In almost all flavors of cognitive modeling, whether symbolic, hybrid or connectionist, it is tacitly assumed that there is a one-to-one relationship between tasks and goals. In this context, we consider a *goal* to be an internal representation that is used to recruit the appropriate knowledge to achieve that goal. Most models model just use a single task and a single goal, and all the knowledge incorporated in or acquired by the model is just for that task.

The one-task-one-goal approach puts many restrictions on what can be achieved by cognitive modeling. It ignores the question how goals are set, prioritized and abandoned. It cannot answers questions about what *other* things people are thinking about when they carry out a task, which might affect the task either positively or negatively. For instance, metacognitive planning ahead may have a positive effect on performance, whereas distraction may have a negative effect. But distraction may be a positive influence if it prevents us from pursuing a hopeless goal.

An alternative for the one-task-one-goal approach is to have several active goals to support a single task. The traditional method of doing this is through subgoaling. In particular the Soar cognitive architecture has pursued the idea that a new task or goal can use several subgoals that have already been learned as part of other tasks (Laird, Rosenbloom, & Newell, 1986). Unfortunately, in most Soar models subgoals were specifically designed for a specific main goal. Moreover, the goal stack is now considered by many as a too rigid representation, because typically only a single goal in the hierarchy is active (Anderson & Douglass, 2001), while in reality goals typically compete with each other (i.e. calling while driving).

An alternative for a goal hierarchy is to have several goals active at the same time, as for example implemented in the threaded cognition extension to ACT-R (Salvucci & Taatgen, 2008). Threaded cognition, however, has mainly been used to model multi-tasking, so although a model would have multiple tasks and multiple goals, there was still a one-to-one mapping between tasks and goals (with some exceptions, e.g., Taatgen, Juvina, Schipper, Borst, & Martens, 2009).

Another effort to break the monolithic goal structure is the PRIMs (Primitive information processing elements) theory, another extension to ACT-R (Taatgen, 2013). PRIMS allows us to go beyond the original tasks by breaking down task-specific rules into combinations of primitive information processing elements that in themselves are task-general. Learning a new task involves combining those primitive elements into task-specific rules, but the byproduct of the learning trajectory is that the model also learns task-general rules that it can use for other purposes. This means that PRIMs can model knowledge transfer from one task to another. A limitation of PRIMs is that task knowledge is still specified in terms of task-specific operators that are linked to a single goal.

Altmann and Gray (2008) explore a different aspect of goals: the current goal is not set by production actions, as is the case in most models, but the goal with the highest activation determines the actions. Their goal representations are susceptible to decay, and therefore the reaction times for a particular goal gradually increase as subjects continue doing the same task. Rehearsal processes can influence goal activations, which is the primary process to control what the current goal is. Still, at any moment a single goal is active for a single task.

In this paper we will combine these three approaches as part of a new version of PRIMs (PRIMs 2.0), in which a single task is implemented by multiple goals. These goals are specified in such a way that they are can be reused for other tasks, and have associated activation levels that determine which goal is most influential at a certain moment. We will then use this to build a model of distraction. Distraction, or self-interruption, is a major problem in our information society, because regular work progress is threatened by email-checking and Facebook updating. It is therefore of importance what factors influence self-interruption, which may enable us to find ways to mitigate or control it. First, we will explain the new version of PRIMs in detail.

The PRIMs 2.0 goal representation

As an example to illustrate the goal representation we will use part of a task that we will use later on: solving simple equations. The task of solving an equation like $5x + 2 = 12$ is represented by four parallel goals: reading the equation into working memory, transforming the equation, doing arithmetic, and giving the answer (Figure 1). The four goals are not carried out in parallel, of course, but their representations are all active. Active goals spread activation to *operators* in memory that can carry out that goal. Operators are the declarative counterpart of production rules, so they have conditions that are tested, and actions that are carried out when the conditions are satisfied (see Taatgen, 2013, for details). When solving an equation, first operators are retrieved that are associated with the reading goal, because there is no mental representation of the equation yet. Once there is a representation, operators associated with the transformation and arithmetic goals alternate in solving the equation, until the answer goal can key in the answer.

In this example all four goals are active throughout problem solving, and the conditions of the operators ensure that they are carried out in the right order. This is not always possible, so sometimes goals have to be added or removed. However, this requires a particular control strategy, which makes learning harder.

Ultimately, goals only influence the activation of operators. This means that a goal doesn't guarantee a matching operator is selected, it only makes it more likely. Other factors can influence the retrieval of operators, though, for example external stimuli or the content of particular declarative retrievals. It is therefore possible that an operator is retrieved that has nothing to do with the current goals, leading to distractions. As we will see later on, it is in between operators for different goals that distractions have an opportunity to intervene.

Key Principles

The key principles of the PRIMs 2.0 goal representation are as follows:

Goals are carried out by operators that are associated with that goal. This is of course true for almost any symbolic architecture, but uniquely in PRIMs multiple goals can be active, and operators can also be associated with multiple goals. Operators do not refer directly to goals or vice versa, there are connected through strength of association only.

Goals have an activation value that is susceptible to all ACT-R memory processes. As a consequence, not all goals are equal, and the goal with the highest activation has a higher probability of recruiting operators it needs.

The activation value of a goal determines how much activation it spreads. All active goals are stored in slots in ACT-R's goal buffer. This means that they are sources of spreading activation. However, instead of the standard spreading activation of W/*n*, as is in regular ACT-R, the W_j of a goal is equal to its activation. This means that operators that are associated with a certain goal receive spreading activation proportional to the activation of that goal.

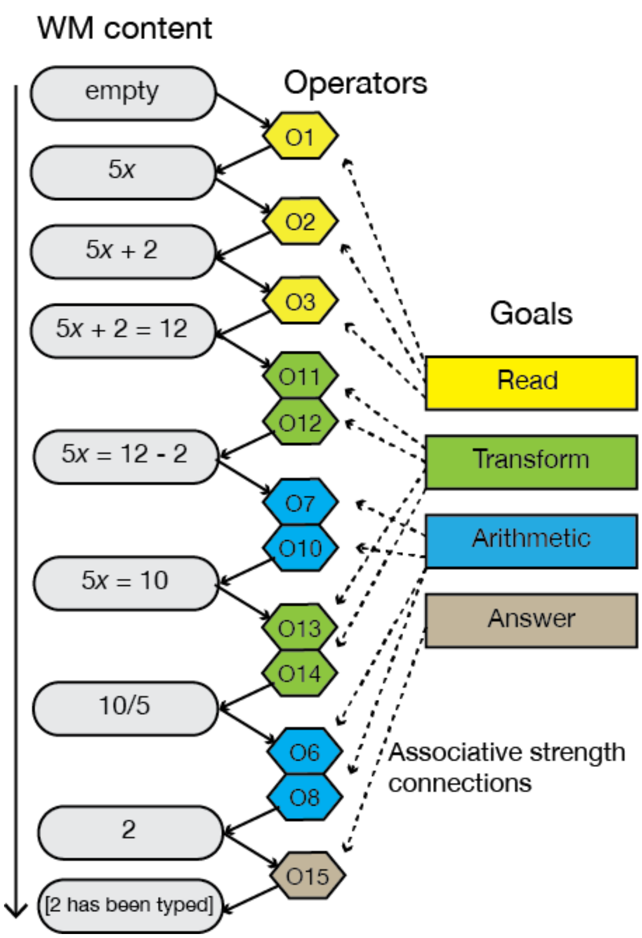

Figure 1. Illustration of how operators associated with different goals together solve an equation

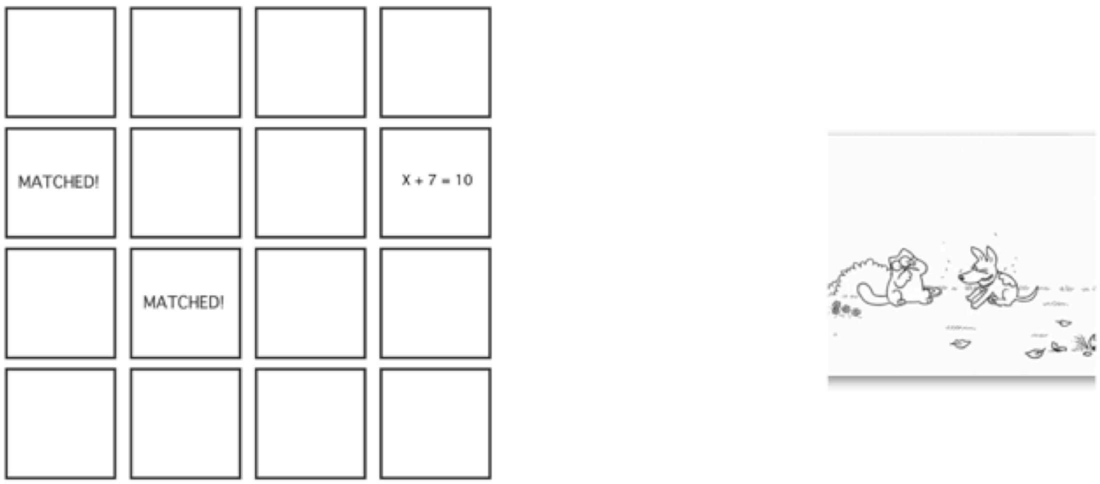

Figure 2. Example of the memory game on the left, and the movie on the right. The memory game is in the medium condition, and two of the cards have already been matched.

The most active operator *whose conditions are satisfied* is carried out. This is not necessarily an operator for the most active goal, because it may have no operators that currently match. Moreover, other influences can add activation to operators to influence the selection, in particular spreading activation from other sources (perception, working memory, memory retrieval, etc.).

Activation of a goal can be increased by explicitly retrieving it (possibly repeatedly). Retrieval is a deliberate strategy to influence the priority of goals. Increased influence can be achieved by multiple retrievals (rehearsal, cf. Altmann & Trafton, 2002).

Activation of a goal decays over time. This means that if goals aren't maintained, or reinforced in any other way, they decay and disappear.

Operators associated with the same goal are also associated with each other. This makes it more likely that an operator for the same goal as the previous operator is chosen.

To demonstrate the power of this approach, we will use it to model a distraction experiment.

Experiment

The main idea of the experiment is that subjects had to carry out different tasks of varying difficulty level. While they carried out the task, a video played at the other side of the screen. The video was unrelated to the task. The extent to which subjects in the experiment were distracted by the video was measured with an eye tracker.

The experiment involved two different tasks, one focusing on mental operations, and the other on visual operations. Each had three different levels of difficulty. In the *memory game*, subjects played the game of *Memory* or *Concentration* with cards with equations instead of pictures. Sixteen cards were displayed on the screen. Subjects had to click on the cards, which would reveal the equation on the back. Clicking on two consecutive cards with the same solution to the equation would remove them, with the eventual goal of removing all sixteen cards (Katidioti, Borst, & Taatgen, 2014). There were three levels of difficulty: the easiest level had equations of the form $4 + 2 = x$, basically simple arithmetic. Medium level equations were of the form $x + 4 = 16$, requiring a transformation followed by arithmetic, and hard question had the form $5x + 2 = 12$, as illustrated in Fig. 1, requiring several operations to solve. Figure 2 shows a screen-shot of the experiment.

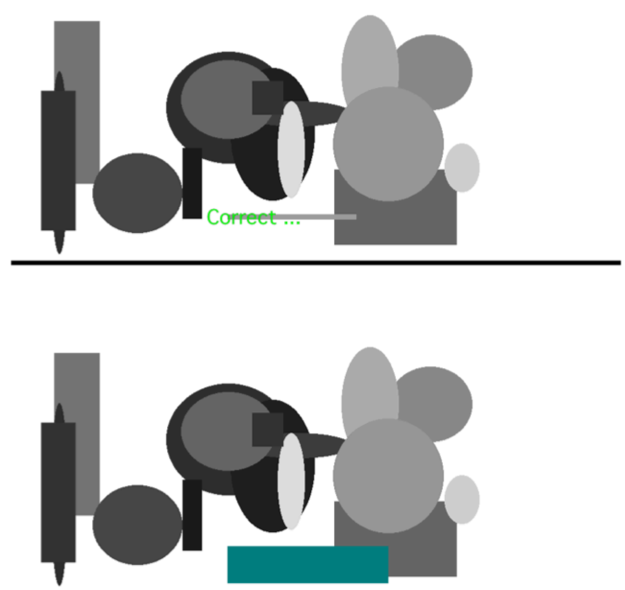

Figure 3. Example of the find-the-differences game in the medium condition, just after feedback.

In *find-the-differences game*, subjects were presented with two pictures on the top and bottom of the screen that each consisted of a number of random shapes (colored ovals and rectangles). Both pictures were identical except for one small difference in one of the shapes (Figure 3). The task was to find the difference, and click on it. In the easy condition, each picture consisted of 2-4 shapes, in the medium condition 15-17 shapes, and in the hard condition 40-42 shapes. Like in the memory game, a movie that was unrelated to the task also played at the other side of the screen.

Subjects in the experiment either did the find-the-differences game or the memory game (25 participants per task). They performed one of the tasks for 15 minutes at each level of difficulty, for a total of 45 minutes.

Our theory about choice in multitasking is that people tend to switch to another task if that task needs resources that are not currently used by the present task. In the experiment, the distraction requires use of the visual resource, so the prediction is that if the main task needs fewer visual resources, the frequency of distractions by the video will be higher. We therefore predict that the frequency of distraction increases with difficulty for the memory game, because solving an equation temporarily requires fewer visual resources, which are then free to move to the video. In contrast, distraction decreases with difficulty for the find-the-differences game, because visual resources are more occupied by the more difficult games.

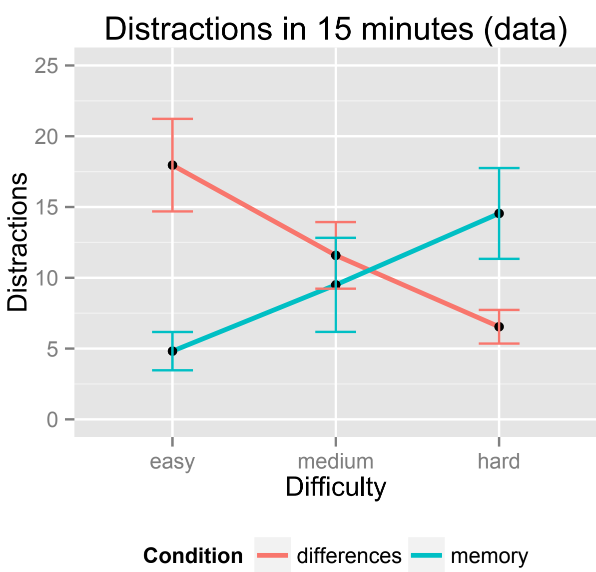

Figure 4. Results of the experiment. Error bars represent 1 standard error.

Figure 4 shows the mean number of distractions (i.e., eye-movements to the video) per subject in each of the 15 minutes blocks of playing either game (distractions between games were removed). The effect of the level of difficulty on the number of distractions is highly significant: when we fit generalized linear mixed effect models based on a Poisson distribution to the data, all differences between different levels of difficulty are significant with p values less than 0.001.

To conclude, the experimental results support the theory that distraction increases if the resources that are available match the resources that the distraction requires (in this case visual resources). We have similar, although weaker, evidence that this is also the case for working memory, where subjects tended to switch to a secondary task involving memory more often at moments that the memory requirements of the main task had just decreased (Katidioti et al., 2014). The next challenge is to construct a model that reproduces this behavior.

A Model of Distraction

In the new PRIMs 2.0 representation, tasks are represented by multiple goals. However, these goals only spread activation to applicable operators, so they do not enforce that only task-relevant operators are chosen. This means that at any point an operator can be retrieved that is not related to the task, assuming that operator has a high enough activation. In our case, the video is a perceptual input that is associated with operators that propose to attend the video. A condition of these operators is that the visual resource is available.

A model of the memory game

For simplicity, we have not constructed a model that plays the whole game, but a model that just solves equations of varying difficulty. We think that this partial model captures the essential characteristics of the task as a whole. Figure 1 already gave a clear representation of the structure of the model: the task is represented by four goals, read, transform, arithmetic and answer. Each of these goals has a small set of associated operators that implement that goal. The hard equations require the sequence as it is shown in Figure 1, involving 12 operators. The medium and easy equations use the same operators, but omit some of them: the medium equations skip the second transform and arithmetic sequence, and therefore only require 8 operators. The easy equations require no transformations, just the final arithmetic, and one fewer read operator, so a total of 5 operators.

In addition to the task-related operators, the model has two operators that respond to the distraction. The first of these has as a condition that the visual resource is not used, and moves attention to the video. The second operator activates at the moment that the video is attended, and disengages immediately. This reflects the empirical fact that in the experiment, subjects typically attended the video for only 200-400ms.

Most of the time, the distraction operators do not have much of a chance to intervene in the equation solving process. When reading operators are engaged they have no chance at all, because the visual resource is in use, whereas a condition of distraction is that the visual resource is free. Whenever the model transitions between an operator for a transform step and one for an arithmetic step (or the other way around), however, distraction has a small probability of winning the competition due to activation noise.

The model can explain the data, because the distraction operator only competes when the model is not reading the equation (which takes proportionally more time as the equation is easier), and because in the harder conditions the model switches more often between transformation and arithmetic, providing more opportunities for distraction.

A model of the find-the-differences game

The model of find-the-differences is relatively simple. It consists of three goals: a *search* goal that attends an unattended feature in the top picture, a *compare* goal that, given an attended feature in the top picture, finds the corresponding location in the bottom picture, and then compares the two. If they are the same, search continues, and if they are different, the *click* goal then clicks on the location where the difference was found.

Figure 5 illustrates the process: it first attends an arbitrary unattended feature, in this case an oval feature in the top figure at location (10,10). It represents this as "Oval4", which in this simplified representation stands for an oval of a particular shape and size. The compare goal then takes over, and finds the matching location in the bottom picture (O2), and concludes that that location also contains an Oval4. It therefore clears the visual buffer, allowing the search goal to find a new feature. In the second attempt, the feature in location (20,15) turns out to be different, which allows the click goal to retrieve the operator that clicks the location. The model keeps track of object it has checked, so will no revisit those.

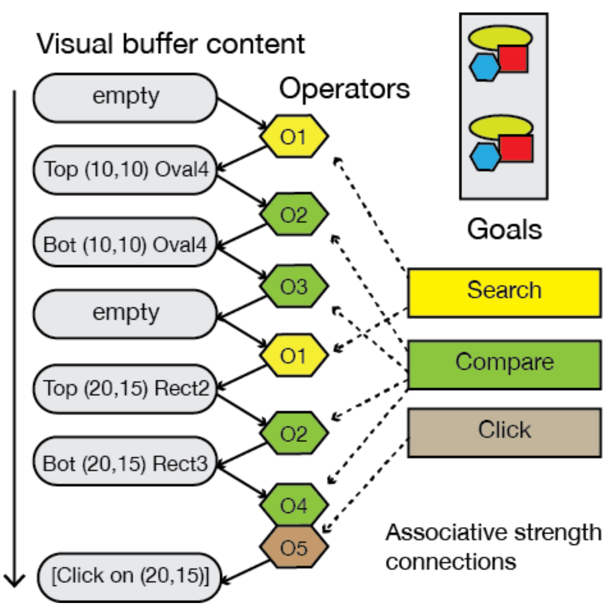

Figure 5. Illustration how the find-the-differences model operates on the picture in the top-left of the Figure.

Distraction is modeled in exactly the same way as in the memory game: whenever the visual resource is unused (i.e., when it is empty, so after each comparison), the distraction operator can direct visual attention to the movie. However, the model makes an additional assumption about the strength of this distraction, namely that it is proportional to the number of yet unattended visual features on the screen. If there are many unattended features on the screen, the video spreads less activation to the distraction operator than when that number is low. Nyamsuren and Taatgen (2013) have extensively modeled this spreading activation from perception to declarative memory: the spreading in this model is based on that work.

The model can explain the data because in the easier version of the task there are fewer visual objects on the screen, causing the movie to spread more activation to the distraction operator, and therefore increasing the probability that it is selected.

Results

Figure 6 shows the results of the simulation. Although the quantitative fit with the data is very good, it required fitting of the parameter that determines how much activation the distraction operator receives. The main quality of the model is therefore the ability to match the qualitative nature of the data.

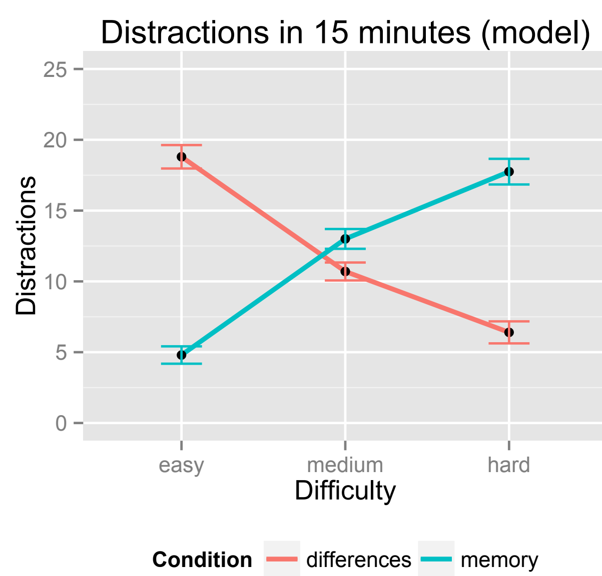

Figure 6. Model fit of the experimental data (from Fig. 4) as a function of task and difficulty level. Error bars indicate 1 standard error.

Although we cannot compare the performance on the memory task directly with the data, because we only partially modeled that task, we can compare performance on the find-the-differences task. Figure 7 shows the results.

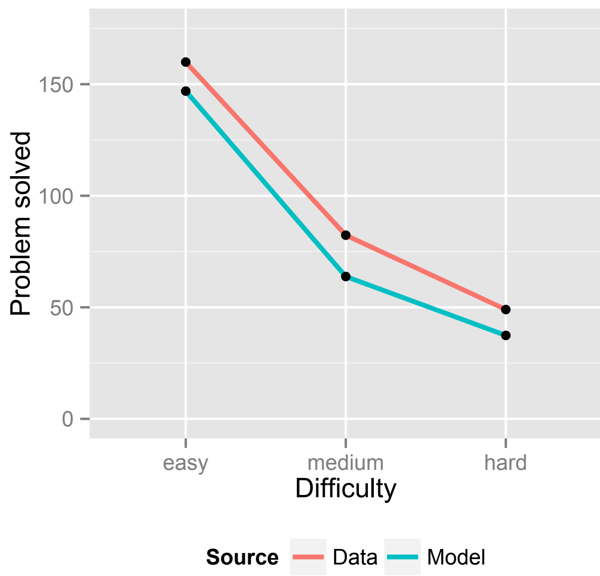

Figure 7. Comparison of number of problems solved between model and data in the find-the-differences game

Discussion and Conclusions

An important problem not addressed by previous models is how task- and non-task goals interact in producing behavior. This makes it hard, or impossible, to model why people suddenly change or give up on tasks. The example of distraction shows that the extension of PRIMs offers new options of modeling phenomena that would be hard to model in existing architectures. More monolithical approaches would probably have a hard time explaining why in some cases distraction increases with workload, and in other cases decrease with workload. In the case of a task that is heavy on reasoning, like the memory game, distraction can slip in at moments where one goal takes over processing from another goal, explaining why problems that require more of such switching are more susceptible to distraction. However, distraction can only intervene if it can latch on to resources that are currently unused. In the experiment, the distraction is visual, and can therefore only succeed if the visual resource is unused.

There is, of course, a danger in weakening the role of the task goal, because we don't want it to be derailed all the time by distraction or irrelevant other skills. This will require a more robust approach to modeling. However, other skills are not necessarily always distractions, but can be beneficial for the actual task, and thereby increase robustness.

In this experiment, distractions were relatively neutral: it didn't help nor hurt performance on the main task. However, in general other thought processes may be added as goals with slightly lower priority than the main task goals. For example, planning ahead is useful to do at moments that resources are available for such planning. A more "neutral" form of parallel processing may involve mind-wandering, which may be harmless, or may eventually turn into a self-induced distraction, and explain why people sometimes suddenly decide to check their email while doing something that is mentally taxing. PRIMs 2.0 opens possibilities for investigating mental processes that we all believe are taking place during experiments, but that we never model.

Another feature of PRIMs 2.0 is the option to build a model of a task by selecting certain existing goals that are connected to appropriate operators. That means that learning a new task involves the selection of goals, filling in certain specific values for those goals, and specifying a control strategy. A control strategy can be very simple: in our example here all goals were active in parallel. But in other cases it may be necessary to actively reinforce goals, in the same manner as Altmann & Gray (2008) did in their model of task switching.

Acknowledgements

This research was supported by ERC StG 283597 MULTITASK from the European Research Council.

References

Altmann, E. M., & Gray, W. D. (2008). An Integrated Model of Cognitive Control in Task Switching. *Psychological Review*, 602-639.

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. Cognitive Science, 26, 39–83.

Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the Cost of Goal Retrieval. *Cognition*, *27*(6), 1331–1346. doi:10.1037//0278-7393.27.6.1331

Katidioti, I., Borst, J. P., & Taatgen, N. A. (2014). What happens when we switch tasks: pupil dilation in multitasking. *Journal of Experimental Psychology: Applied, 20*(4), 380–396.

Laird, J. E., Rosenbloom, P. S., & Newell, A. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning*, (1), 11–46.

Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive Systems Research*, 62-71.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, *115*(1), 101–130.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*(3), 439–471.

Taatgen, N. A., Juvina, I., Schipper, M., Borst, J. P., & Martens, S. (2009). Too much control can hurt: A threaded cognition model of the attentional blink. *Cognitive Psychology*, *59*(1), 1–29.