



A step-by-step tutorial on using the cognitive architecture ACT-R in combination with fMRI data



Jelmer P. Borst^{a,*}, John R. Anderson^b

^a University of Groningen, Department of Artificial Intelligence, Nijenborgh 9, 9747 AG, Groningen, The Netherlands

^b Carnegie Mellon University, Department of Psychology, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

HIGHLIGHTS

- We provide a tutorial on using the cognitive architecture ACT-R with fMRI data.
- Region-of-interest analysis and model-based fMRI analysis are discussed.
- ROI analysis is used to constrain and evaluate ACT-R models.
- Model-based fMRI analysis is used to find the neural correlates of model processes.
- Data and code to perform the analyses and recreate the figures are provided.

ARTICLE INFO

Article history:

Available online 20 June 2016

Keywords:

ACT-R
fMRI
Cognitive architecture
Model-based analysis
Tutorial

ABSTRACT

The cognitive architecture ACT-R is at the same time a psychological theory and a modeling framework for constructing cognitive models that adhere to the principles of the theory. ACT-R can be used in combination with fMRI data in two different ways: (1) fMRI data can be used to evaluate and constrain models in ACT-R by means of predefined Region-of-Interest (ROI) analysis, and (2) predictions from ACT-R models can be used to locate neural correlates of model processes and representations by means of model-based fMRI analysis. In this paper we provide a step-by-step tutorial on both approaches. Note that this tutorial neither teaches the ACT-R theory in any detail, nor fMRI analysis, but explains how ACT-R can be used in combination with fMRI data. To this end, we provide all data and computer code necessary to run the ACT-R model, carry out the analyses, and recreate the figures in the paper. As an example dataset we use a relatively simple algebra task. In the first section, we develop an ACT-R model of this task and fit it to behavioral data. In the second section, we apply a predefined ROI-analysis to evaluate the model using fMRI data. In the third section, we use model-based fMRI analysis to locate the following processes in the brain: retrieval of mathematical facts from memory, working memory updates, motor responses, and visually encoding the problems. After working through this tutorial, the reader will have learned what can be achieved with the two different analysis methods and how they are conducted; the example code can then be adapted to a new dataset.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Many psychological theories, and consequently, many computational cognitive models focus on a single task or even on part of a task. For example, decision-making models often only simulate the decision-making aspect of a task, and include 'non-decision time' to represent all non-decision processes (e.g., Forstmann et al., 2008; Turner, van Maanen, & Forstmann, 2015). Although this

approach to modeling has certainly yielded significant insights, the danger is that the mechanisms that are modeled might be task-specific, instead of representing concepts that are more generally valid. If the goal is to understand the functioning of the human mind – in contrast to understanding what is happening in a single task – this approach might be counter-productive, as famously noted by Newell in his 20-questions paper (1973).

In contrast to single-task models, *cognitive architectures* try to capture general cognitive mechanisms and are aimed at modeling complete tasks from perception to response execution (Anderson, 2007; Newell, 1990). A cognitive architecture is a psychological theory, for instance explaining how our declarative memory

* Corresponding author.

E-mail addresses: j.p.borst@rug.nl (J.P. Borst), ja+@cmu.edu (J.R. Anderson).

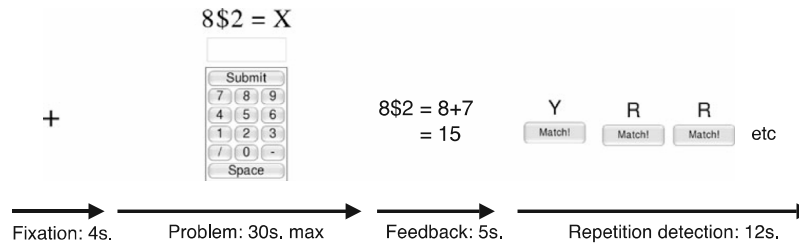


Fig. 1. Setup of the experiment. Each trial started with 4 s of fixation, followed by the pyramid problem for a maximum of 30 s or until the 'Submit' button was clicked. Between trials a repetition-detection task was displayed for 12 s to return the brain to a common baseline. Source: Reprinted with permission from Anderson et al. (2011).

system functions (ACT-R; Anderson, Bothell, Lebiere, & Matessa, 1998), how executive processes determine dual-task performance (EPIC; Meyer & Kieras, 1997), or how new tasks are learned by information transfer (PRIMS; Taatgen, 2013). In addition, cognitive architectures are implemented as computer simulations, in which cognitive models of particular tasks can be instantiated (e.g., recovering from interruptions, Borst, Taatgen, & Van Rijn, 2015; driving a car, Salvucci, 2006; associative recognition, Schneider & Anderson, 2012). Thus, the cognitive architecture itself provides the general psychological theory – the architecture of the mind – while models run on this architecture to simulate behavior in particular tasks.

This approach has a number of advantages. Most importantly, it ensures that the cognitive mechanisms used to account for different tasks are the same, and therefore have to be task-general. Second, because models cover complete tasks and are implemented as computer simulations, it is possible to directly compare their response times and accuracy to human participants. Third, because action and perception mechanisms are included in the models, the predictions of these models, such as response times, are constrained to correspond to what is physically possible for humans (e.g., Borst, Buwalda, Van Rijn, & Taatgen, 2013; Meyer & Kieras, 1997).

Although it is possible to evaluate cognitive architecture models by comparing their response times and error rates to human data, these data are often insufficient to test all the detailed assumptions of these models (problems of this sort have been noted with most kinds of modeling; e.g., Pitt & Myung, 2002; Roberts & Pashler, 2000; Shiffrin, Lee, Kim, & Wagenmakers, 2008). To increase the constraints on models in the ACT-R architecture, researchers have turned to fMRI data. There are two ways in which ACT-R can be used in combination with fMRI data (Borst & Anderson, 2015). First, each ACT-R model predicts fMRI activity in several predefined regions-of-interest (ROIs) in the brain (Anderson, 2007; Anderson, Fincham, Qin, & Stocco, 2008). Because these regions are predefined and independent of new datasets – they have remained essentially unchanged since 2003 – such predictions can be compared to human data to judge the quality of the model and guide further model development. Second, predictions from ACT-R models can be used to locate the neural correlates of model processes and representations through model-based fMRI analysis (Borst, Taatgen, & Van Rijn, 2011).

In this tutorial, we will guide the reader through both analysis methods and provide the data and computer code necessary to replicate the analyses and recreate the figures in this paper (see <http://www.jelmerborst.nl/tutorial>).¹ To perform the analyses we used the ACT-R 7.0 Standalone Version (<http://act-r.psy.cmu.edu>),

Matlab (The MathWorks, Inc.), and the fMRI analysis software SPM8 (Wellcome Trust Centre for Neuroimaging). We included all software and toolboxes, except Matlab itself. In Section 2 we will introduce the task and the ACT-R cognitive architecture, and develop the cognitive model. In Section 3 we will apply the predefined ROI analysis.² Finally, in Section 4 we will use model-based analysis to locate the neural correlates of visually encoding the task, retrieving mathematical facts from memory, updating working memory, and issuing motor responses. Note that we also provide all intermediate results, in order to allow for starting the analysis at any point, and skipping parts of the analysis. We indicated in the text when analysis steps take longer than a few minutes.

2. Developing the ACT-R model

2.1. The task

The dataset that we use as an example in this tutorial concerns a relatively simple algebra task, which was part of a larger study used to investigate metacognitive activity in mathematical problem solving (Anderson, Betts, Ferris, & Fincham, 2011). In addition, it has been used to validate new regions-of-interest for ACT-R (Borst, Nijboer et al., 2015). We will follow the analysis stream and trial selection from this latter paper.

In the experiment, participants solved so-called *pyramid problems*, which take the form *base \$ height = value*, for instance $5 \$ 3 = 12$ (see Anderson et al., 2011, for details on the study and the rationale for using pyramid problems). A pyramid problem is solved by a process of repeated addition, where the *base* indicates the starting value and the *height* the number of additions, where each addend is one less than the previous: $5 \$ 3 = 5 + 4 + 3 = 12$. Participants were asked to solve three types of problems:

- solve-for-value: $6 \$ 3 = x$
- solve-for-height: $6 \$ x = 15$
- solve-for-base: $x \$ 3 = 15$.

The *base* ranged from 4 to 9 and the *height* from 2 to 5, resulting in *values* between $4 \$ 2 = 4 + 3 = 7$ and $9 \$ 5 = 9 + 8 + 7 + 6 + 5 = 35$. For analysis purposes, we divided the problems into four groups based on whether they had a small (4–6) or a large (7–9) *base*, and whether they had a small (2–3) or a large (4–5) *height*.

Twenty subjects participated in the experiment. They each solved 72 problems in the fMRI scanner, equally divided over conditions. Responses were entered by clicking on an on-screen numeric keypad with the mouse, followed by clicking the 'submit' button. After submitting a response, subjects immediately progressed to a feedback screen, followed by a repetition detection task (Fig. 1; detailed methods are reported in Anderson et al., 2011).

¹ Note that this is neither a tutorial on ACT-R, nor on fMRI analysis. For more information about ACT-R we refer to act-r.psy.cmu.edu, for more information on fMRI analyses one can visit for example <http://www.fil.ion.ucl.ac.uk/spm/>, <http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/>, or <http://www.freesurfer.net>.

² The results of this analysis have been previously reported in Borst, Nijboer, Taatgen, Van Rijn, and Anderson (2015).

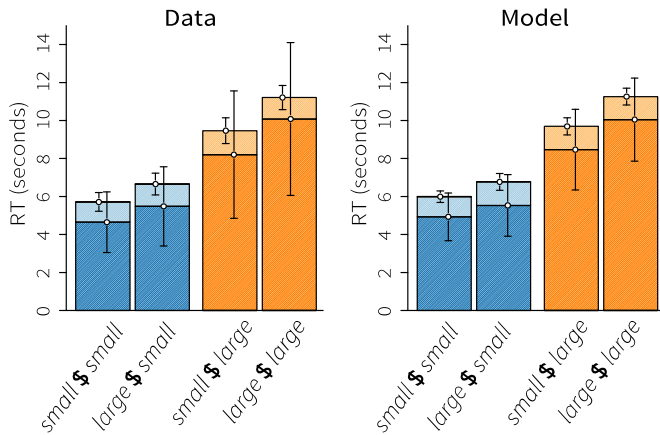


Fig. 2. Response times. Data on the left, model on the right. The lower, darker parts of the bars indicate the time until the first click, the top, lighter parts of the bars the time between the first click and the submit button. Error bars indicate the average standard deviation. Source: Reprinted with permission from Borst, Nijboer et al. (2015).

2.2. Behavior

For this tutorial we use correct trials only. Two subjects were excluded for solving only 49% and 76% of the problems correctly, where all others scored $>85\%$ (Borst, Nijboer et al., 2015). The remaining 18 subjects solved on average 93.4% (SE = 0.9%) of the problems correctly. Of these problems, we removed outliers with response times (RT) exceeding 2 standard deviations from the mean per subject and condition (4.2%). This leaves 1160 problems for analysis in total, on average 64.4 problems per subject.

The left panel of Fig. 2 shows the response times. Height had a clear effect on RTs, with large heights (orange bars) resulting in longer RTs than small heights (blue bars). As height determines the number of terms in the repeated addition, this is not surprising. Base also had a positive effect on RT: a large base resulted in longer RTs than a small base. Fig. 2 shows that most of the variability in RTs is located in the 'cognitive phase' (dark), before the first mouse click, and not in the 'response phase' (light), between the first click and clicking the submit button. These effects were confirmed by a repeated-measure ANOVA (Borst, Nijboer et al., 2015).

2.3. ACT-R

Before we develop the model to explain these data, we will first briefly introduce ACT-R. For more detail, see Anderson (2007, book); or Anderson (2005, article); and/or follow the tutorial units included with the ACT-R distribution.³

ACT-R simulates the human cognitive system as a set of independent modules that are coordinated by a central production system (the procedural module; Fig. 3). Two modules process input from the external world (visual, aural) and two modules send output to the external world (manual, vocal). The other modules process information internally: the control state keeps track of the current goal, the problem state holds intermediate representations (also referred to as the imaginal module), and declarative memory stores facts (called chunks). Information is processed by production rules in procedural memory, which are IF-THEN rules. For example, a production rule might say 'IF there is a pyramid problem in the visual module with a base of 6 and a height

of 3, THEN represent in the control state that a pyramid problem is being solved, and retrieve an addition fact from declarative memory with 6 as the first addend and 5 as the second addend'.

Each module adheres to a set of precisely described properties (Anderson, 2007), which are instantiated in the computational simulation. For instance, it takes 85 ms to visually process an object on the screen, and it takes 200 ms to store a chunk in the problem state. Each fact in declarative memory has an activation level, which represents the likelihood that it is currently needed (Anderson & Schooler, 1991). This activation value is dependent on the recency and frequency of usage of the chunk, as well as on the current information in the other modules. The activation value determines how easy it is to retrieve a chunk from memory: the higher the activation value, the more likely it is that it can be retrieved, and the faster it will be retrieved. For instance, ' $2 + 2 = 4$ ' has a higher activation value and can be retrieved faster than ' $14 \times 19 = 266$ ', if the latter is even represented in memory.

All modules of ACT-R have been mapped onto small regions in the brain (Fig. 3(b)). The original mapping was based on a reading of the literature on regional functions, and adapted slightly based on experience with early experiments. Since 2003 the mapping has essentially remained unchanged (Anderson, 2007; Anderson et al., 2008). As the most straightforward example, ACT-R's motor module was mapped onto the region of the precentral gyrus that represents the hand (Anderson, 2005). The ROIs are assumed to be active when the corresponding module is active; thus, the motor region should be active when an ACT-R model moves its hand. Note that it is not claimed that these regions are the only parts of the brain that are active when a module is active, nor that these regions are only active when a module is active.

Because these regions are predefined, and because it is assumed that these regions are necessarily active when a module is active, they can be used to constrain models by comparing the predicted activity of the ACT-R modules to the data in these regions (Borst & Anderson, 2015; Borst, Nijboer et al., 2015). For example, if ACT-R's motor module is active but no activity is found in the motor ROI, this implies that the model is incorrect. In the next sections we will demonstrate such an analysis not only for the motor module, but also for the visual module, declarative memory, and the problem state. Especially comparing different condition levels of the more central cognitive models often provides important information about the quality of a model.

2.4. Model

The model that we present here is the result of a series of consecutive efforts (Anderson, 2007; Wintemute, Betts, Ferris, Fincham, & Anderson, 2012; the current model was reported before in Borst, Nijboer et al., 2015, and updated to ACT-R 7.0 for the current paper). Developing a model in ACT-R involves specifying the contents of procedural and declarative memory—the IF-THEN production rules and the information those rules can use. To solve pyramid problems, the model requires declarative knowledge about addition problems, for example ' $6 + 5 = 11$ '. It then applies production rules that use this knowledge to solve pyramid problems. The model is included in the material on <http://www.jelmerborst.nl/tutorial>, in the next section we will explain how to run the model and generate predictions.

The model starts by reading the pyramid problem on the screen using the visual module. It encodes the base, height, and value in turn, and follows its own gaze with the mouse pointer (cf. Chen, Anderson, & Sohn, 2001). Based on which of the terms read x , it goes down one of three solution paths. If the value is unknown, it follows a process of iterative addition. It starts by adding the base to the 'base - 1'. To solve such an addition, the model retrieves the solution from declarative memory, which takes a certain amount of

³ The latest version of ACT-R can be downloaded from <http://act-r.psy.cmu.edu/>; the version that we used is included in the material on <http://www.jelmerborst.nl/tutorial>.

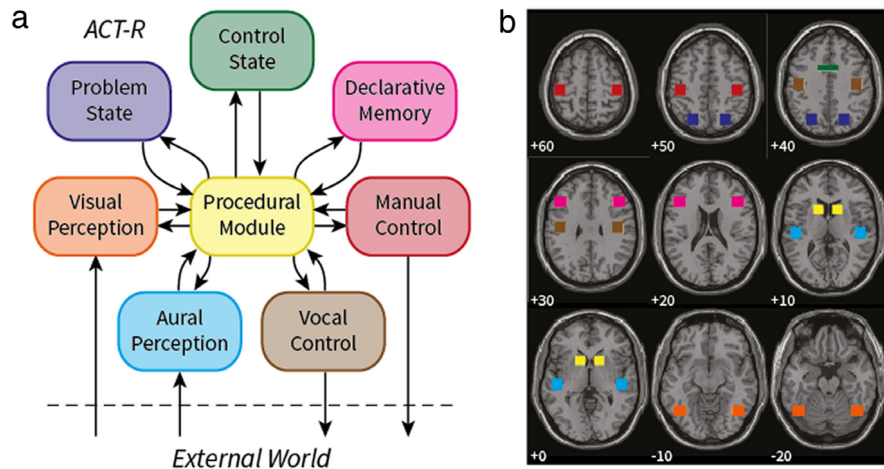


Fig. 3. The ACT-R cognitive architecture (a) and its mapping to brain regions (b). The colors of the modules correspond to the colored squares in the brain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Source: Reprinted with permission from Borst and Anderson (2015).

time (estimated from the behavioral data). It repeats this process until it has added as many terms as the *height* indicated. It then continues to the response phase. If the *height* is unknown, the model follows the same process, but terminates when the total sum is equal to the *value*, and continues to the response phase to report the number of additions it had to perform. The most difficult case is when the *base* is unknown. Subjects reported using a guess-and-check procedure in this case (Anderson et al., 2011), which is simulated by the model. The model first guesses a *base*, and then follows the unknown-*value* procedure described above to calculate the corresponding *value*. If this is equal to the *value* on the screen it reports the guessed *base*, if the calculated *value* is smaller it reports the 'guessed *base* + 1', if it is larger it reports the 'guessed *base* - 1'. After the model determined a solution by following one of these procedures, it looks at the button with the first digit of its answer, moves the mouse cursor to this button, clicks it, and reads what it entered. It then moves on to the next digit, and continues in this manner until the full answer is entered. It finishes by clicking the 'Submit' button, and reading the feedback to determine whether it was correct.

The right panel of Fig. 2 shows the response times of the model. These times are a combination of all the processes the model has to go through: visual perception, memory retrievals, production rules, and manually moving the mouse and entering the response. The duration of the perception and response processes are relatively constant over the different trial types, but the length of the addition process varies considerably with the difficulty of the problems. First, larger *heights* result in more addition steps, and thus in longer total RTs. Second, larger *bases* result in harder additions that include more carries, which add additional computation steps to the model (Borst, Nijboer et al., 2015). Overall, the model reflected the data closely—the greatest discrepancy was the underestimation of the variability of the cognitive phase by the model (i.e., the standard deviation of the model's response times is smaller than of the human subjects). In the next section we will investigate whether the model can also account for the fMRI data that was collected during this task (Anderson et al., 2011), which would provide additional evidence that the model is a good simulation of the cognitive processes underlying this task.

3. Predefined ROI analysis

As explained above, all modules of the ACT-R architecture have been mapped onto small regions-of-interest of the brain, to provide additional constraints for models developed in the architecture. Because these ROIs are predefined and remain unchanged

between studies, they can be used to evaluate new models by comparing model predictions to human data (Anderson et al., 2008; Borst, Nijboer et al., 2015). In this section we will provide a step-by-step tutorial of this method. First, we will use the model to generate fMRI blood-oxygen-level dependent (BOLD) signal predictions for four modules: visual, manual, declarative memory, and problem state. Second, we will select the corresponding fMRI data and evaluate the model.

3.1. Generating model predictions

The fMRI BOLD signal – which we simulate with the model – is a very slow signal, as fMRI scanners measure the oxygenation level of the blood. It peaks around 4–6 s after neuronal activity, and can be described by a so-called hemodynamic response function (HRF; Fig. 4(a); e.g., Friston, Ashburner, Kiebel, Nichols, & Penny, 2007). In contrast, the model's activity occurs immediately and is therefore more comparable to neural activity. To simulate the BOLD response, one can convolve the model's activity on a module-by-module basis with an HRF (Fig. 4(b); Anderson et al., 2008; Borst & Anderson, 2015). In essence, this mathematical procedure assumes a hemodynamic response for each activity of a module, and these responses are summed to produce the final signal. The process consists of two steps.

Step 1. Collecting module activity. First, we run the model on the same pyramid problems that the subjects performed. The file containing the model code, 'pyramid-model_jmp.lisp', is located in the 'model' directory. This file contains Lisp code to run the experiment⁴ and the ACT-R model code (starting at line 548). You can open this file in a text editor to take a first look at the model.

To develop models in ACT-R, it is generally recommended to use a full-fledged Lisp environment that provides a GUI and debugging tools.⁵ However, the ACT-R Standalone version includes a Lisp distribution that can also be used to run models. This is the version included with this tutorial. To start ACT-R, navigate to the 'ACT-R Standalone Mac' or 'ACT-R Standalone Windows' directory and double click on the file 'Run ACT-R.command' on a Mac or on

⁴ ACT-R is written in Lisp, and for that reason experimental code that ACT-R interfaces with is also typically written in Lisp. ACT-R contains a set of functions that can be used to implement simple experiments.

⁵ For example Allegro Common Lisp (<http://franz.com>) or Lispworks (<http://www.lispworks.com>) both have free versions available.

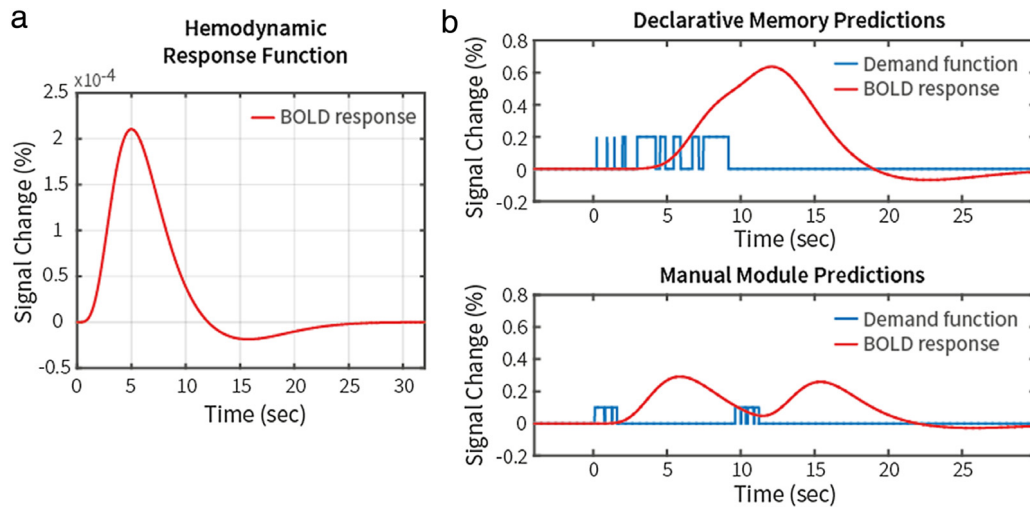


Fig. 4. (a) Hemodynamic response function (HRF) as used by SPM. (b) Declarative memory (top) and manual module (bottom) demand functions and predicted BOLD response for a single trial of the experiment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

'Start Environment.exe' on Windows.⁶ Click on the 'Load Model' button in the environment, and select the model file ('pyramid-model_jmp.lisp'). This loads both the experimental code as well as the ACT-R model itself into Lisp. To use the model in a full-fledged Lisp environment, first load ACT-R, and then the model.⁷

To display the experimental interface and have the model solve four problems in real-time, in the **Listener** window type

```
(run-4-trials)
```

and hit the 'return/enter' key. When the model has finished running, at the bottom of the **Listener** window you will see 'NIL'. To have the model perform all trials of all subjects virtually (no window is displayed and the simulation runs as fast as your computer allows), and save the results to disk, navigate to the **Listener** window and type

```
(run-real-problems).
```

The model will now run through all subjects and trials. This takes between 2 and 6 min, depending on the speed of your computer. The resulting behavior is saved to 'model_#.txt' in 'ACT-R Standalone/model_data/', where '#' indicates the subject number. The corresponding module activity that we need for generating fMRI predictions is saved to 'model_act_#_#_#.txt' in 'ACT-R Standalone/model_data_mri/', where the #-marks indicate subject, block, and trial, respectively.

For more information on the ACT-R theory and running ACT-R models, see the tutorial units included with the ACT-R distribution, or visit the annual summer school at Carnegie Mellon University (Pittsburgh, USA) or spring school at the University of Groningen (Groningen, the Netherlands).

Step 2. Convoluting module activity with the HRF. The files that were generated in Step 1 contain the times in a trial that ACT-R's modules were active. Each line in a file lists the module name, the start time, and the end time.⁸ To make these direct measures of activity comparable to the slow BOLD response, we convolve them with an HRF (Fig. 4). First, we represent module activity as a 0–1 demand function (blue line, Fig. 4(b)). Next, we convolve

these demand functions with the HRF commonly used in SPM (Friston et al., 2007). The code to do this is included in the Matlab script 'ROlanalysis/convolution.m'. You can follow the instructions in this file to recreate Fig. 4. Note that you have to change the working directory to the location of the 'ROlanalysis' folder on your computer at the beginning of the file. First, the HRF from SPM is loaded and plotted (Fig. 4(a)). Next, the module activity for one trial is read in, represented as a 0–1 demand function, convolved with the HRF, and displayed (Fig. 4(b)).

Fig. 4(b) shows the results of convolving a single trial of the *large base–large height* condition. Several declarative memory retrievals occur in quick succession and lead to one large bump in the predicted BOLD profile. The manual module shows a different pattern, with mouse movements at the start of the trial (while reading the pyramid problem) and at the end of the trial to enter the response. This results in a very different BOLD prediction with two clearly distinguishable bumps. Generally speaking, the convolution with the HRF combines and smooths out several short module activities. This is a problem inherent to fMRI, and prevents us from getting a direct measurement of module activity. However, by using the right experimental designs, one can still make a distinction between several possible models. Next, we will perform the convolution for all different subjects, trials, and conditions, and see how the BOLD response predictions react to the experimental manipulations.

Fig. 5 shows the average of the predictions for four modules, split up by experimental condition. This analysis is also part of the Matlab 'convolution.m' script, starting at line 127. All of the model activity (for all subjects and trials) is read in, convolved with the HRF, and aggregated. This takes between 1 and 10 min, depending on the speed of your computer. Fig. 5 shows the results. Starting at the top left, the predictions for declarative memory show a clear increase with condition, with the largest effect for *height*. This effect mirrors the RT effect, and is explained similarly: the more difficult pyramid problems require additional declarative retrievals. ACT-R's problem state module shows a similar pattern. It is used to store intermediate solutions to the pyramid problems, and has to be updated more often for more difficult problems. The manual and visual modules show opposite patterns: the easier the problems, the higher and narrower the BOLD predictions. The explanation is straightforward: according to the model, the same number of mouse and eye movements has to be made in the different conditions, but they are spaced further apart in the more difficult conditions because the mathematical solving phase is longer. As a result, the conditions with large *heights*

⁶ In case the ACT-R version included is not compatible with your machine, you can download the latest version from <http://act-r.psy.cmu.edu/>.

⁷ When using a full-fledged Lisp environment, start the ACT-R interface by calling (run-environment) before running the model. Without the ACT-R interface, the model might not run because of issues with button placements in the graphical windows.

⁸ Note that these times will be slightly different each time the model is run, as declarative memory processes have an associated noise component.

Table 1
ACT-R modules and associated brain regions.

ACT-R module	Brain region (left hemisphere)	Talairach–Tournoux coordinates	MNI coordinates
Declarative memory	Inferior frontal sulcus (BA 45/46)	−42, 23, 24	−43, 24, 25
Problem state	Intraparietal sulcus (BA 7/39/40)	−24, −63, 40	−24, −67, 44
Manual	Precentral gyrus (BA 3)	−42, −20, 50	−42, −23, 54
Visual	Fusiform gyrus (BA 37)	−41, −61, −9	−43, −60, −16

MNI = Montreal Neurological Institute.

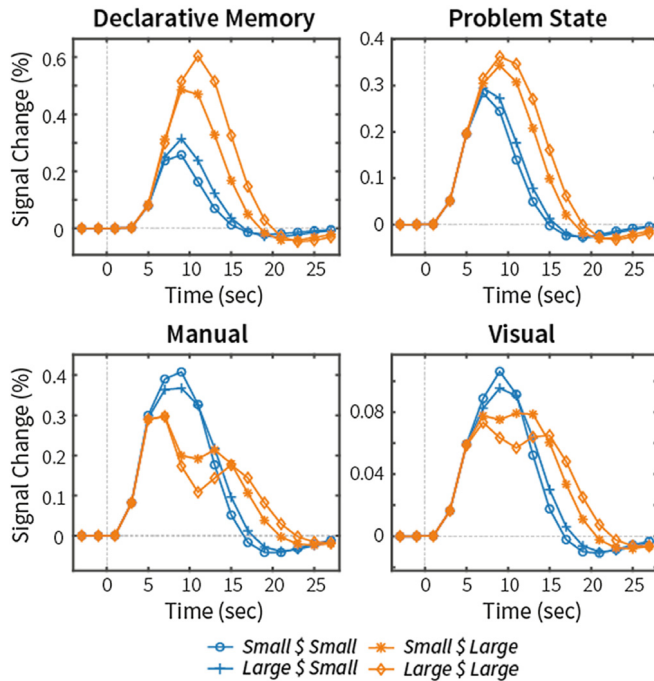


Fig. 5. Model predictions for four ACT-R modules.

show two bumps: one before the solving phase and one after (cf. Fig. 4(b), bottom). However, the area under the curve is similar in all conditions, reflecting total activity (it is a little larger in the more difficult conditions, because the answers require more key presses in these conditions). In the next section we will test if the data is similar to the model's predictions.

3.2. Selecting fMRI data

To select the data, we use preprocessed fMRI scans (realigned, normalized, smoothed; Borst, Nijboer et al., 2015) and a definition of the ROIs. The selection is done using SPM8 in combination with the MarsBaR toolbox (Brett, Anton, Valabregue, & Poline, 2002). The preprocessed fMRI scans, the ROI definitions, SPM8 and MarsBaR are all included in the material on <http://www.jelmerborst.nl/tutorial>. The properties of the ROIs are listed in Table 1, the ROIs are shown in Fig. 3(b), and MarsBaR compatible ROI files are included in 'ROIanalysis/rois/'. Each ROI is $15 \times 15 \times 12$ mm. The preprocessed scans are located in the 'data' directory, including a design file 'pyramids.df' that describes which scans belong to which subject, trial, and condition. The Matlab script 'ROIanalysis/roi_data_selection.m' contains the code and instructions to select the data and recreate Fig. 6.

First, all scans of a subject are read in and scaled to an average value of 100 over all scans and voxels, as is common in fMRI analysis (e.g., Friston et al., 2007). Second, the data in each ROI is collected, and a mean value per ROI is calculated. These two steps take between 5 and 36 min. Third, the percent BOLD change with respect to the average of the first two scans in a trial (during the fixation screen, Fig. 1) is calculated. Depending on your

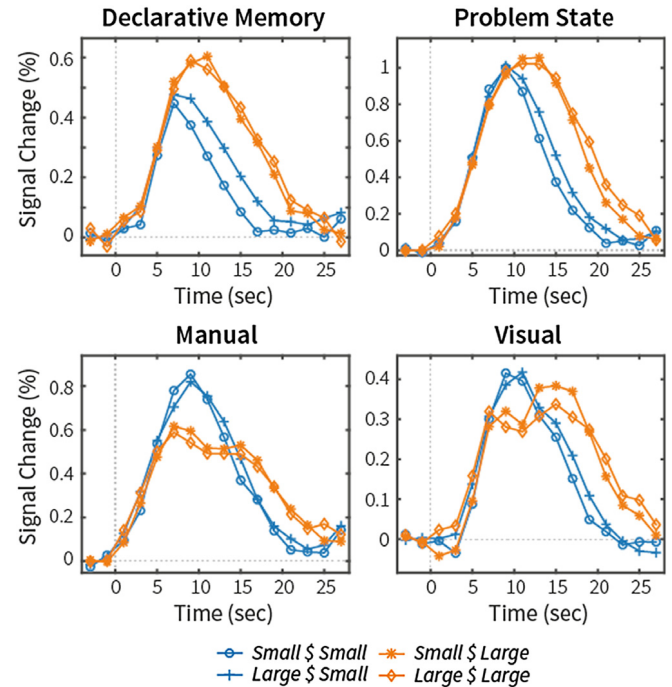


Fig. 6. Data in four ROIs, corresponding to the model predictions in Fig. 5.

experimental design, one can also calculate %change compared to only the first scan or to longer fixation periods at the start of a block. Calculating %change on a trial-by-trial basis ensures that slow signal changes are removed (in standard fMRI analysis a high-pass filter is typically applied for this purpose). Finally, the data are averaged within each ROI and condition. The results are shown in Fig. 6.

3.3. Discussion

Comparing the data in Fig. 6 to the model predictions in Fig. 5 shows that the model gives a good qualitative account of the data: the order of the conditions was correctly predicted for all four modules. In the declarative memory module, however, there is hardly a difference between small and large bases for the large heights (orange lines) where the model predicted a sizeable difference. The problem state module has a very good fit on first sight, but it turns out that in the data the activity is more prolonged (i.e. the curves return to baseline only at 20 s, in the model predictions at 15 s). This might indicate that working memory also plays a role in processing the feedback – which the model ignored –, not an unreasonable assumption. In addition, it could reflect the greater variability of the RT's in the data. Finally, the manual and visual predictions are generally very good, except that the difference between the small \$ large and large \$ large problems is less clear in the data, which might be due to the smaller variation in the model's RTs (Fig. 3).

Quantitatively, large differences can be observed between the model and data. The ACT-R model predictions do not take into account differences in the HRF between regions, and are generally

bad predictors of amplitude. As a solution, one can fit a separate scale parameter to each module. For instance, Anderson, Qin, Jung, and Carter (2007) estimated a scale parameter of 2.38% for the visual module, 3.16% for the manual module, 1.79% for the problem state, and 1.15% for declarative memory. Although multiplying the current predictions in Fig. 5 with these values would lead to a better fit, it would by no means remove the whole difference between Figs. 5 and 6. So far, we have not been able to find stable scale parameters (see e.g., Stocco & Anderson, 2008, for very different estimates), and have therefore concentrated on qualitative predictions.

Based on the current fit, we argue that the model gives a reasonable account of what is happening when participants solve pyramid problems. Several things could be improved, as indicated above, but the basic cognitive processing is accounted for. However, in other cases we have had to abandon a first model, and propose different processing streams to account for the data—processing streams that were inspired by the neural data (Nijboer, Borst, Van Rijn, & Taatgen, 2016).

In summary, the ROI analysis can be used to evaluate models because we use predefined ROIs, and do not change these ROIs on a study-by-study basis. Although the current ROIs are based on the neuroscientific literature, we have recently performed a large data-driven meta-analysis in which we have shown that these ROIs are also close to the best predictors of the ACT-R modules (Borst & Anderson, 2013; Borst, Nijboer et al., 2015). In the next section we discuss a second technique, model-based fMRI analysis, which can be used to locate model constructs in the brain.

4. Model-based fMRI analysis

The ROI analysis requires an existing mapping between ACT-R modules and brain regions, and has the goal to constrain models. However, at other times we want to discover new connections between model constructs and brain areas, or evaluate existing mappings. To this end one can apply model-based fMRI analysis (Borst et al., 2011; Gläscher & O'Doherty, 2010; O'Doherty, Hampton, & Kim, 2007). In a *standard* fMRI analysis, one typically uses the condition-structure of the experiment as the basis for regressors in a General Linear Model (GLM; e.g., Friston et al., 2007). For each voxel (a pixel in three-dimensional space), an independent linear model is fit using these regressors (after convolving the regressors with an HRF, cf. Fig. 4), and conditions are compared to determine in which voxels the BOLD level differs significantly between conditions. The drawback of this approach is that a cognitive function is assumed to be 'on' or 'off' in each condition. In addition, especially with tightly coupled processes such as declarative memory retrievals and working memory updating, it is very difficult to have one process without the other, making it hard to isolate such functions.

Model-based fMRI analysis alleviates this problem by using predictions of a computational model as regressors in the GLM. These predictions can take the form of parameter values per trial (e.g., Daw, Gershman, Seymour, Dayan, & Dolan, 2011; Gläscher & O'Doherty, 2010; O'Doherty et al., 2007; van Maanen et al., 2011) or of activity of higher-level processes (i.e. module activity; Borst & Anderson, 2013; Borst et al., 2011). As in standard fMRI analysis, a linear model is fit for each voxel. Voxels in which the BOLD activation correlates significantly with the model predictions are then assumed to be involved in the simulated process. Using a model has the advantage that a different predictor can be used for each trial, and that multiple processes can be active simultaneously and to different degrees within each trial, making model-based fMRI analysis more powerful than standard fMRI analysis (Borst et al., 2011).

In the remainder of this section we will demonstrate a model-based analysis of the pyramid dataset. We will attempt to locate declarative memory retrievals, updates to the problem state, manual actions and, visual perception. As the first step we generate module predictions and rescale them with respect to the behavioral data. Next, we fit a GLM and discuss the results. Note that we included the intermediate results of all analysis steps, making it possible to skip steps or subjects in the analysis, as some of the analyses below take a significant time.

4.1. Generating model predictions and rescaling

For a model-based fMRI analysis on the basis of an ACT-R model, we use module activity patterns as regressors on a trial-by-trial basis. Thus, the model is run on the exact same trials as the participants. Analogous to the ROI analysis, the resulting activity patterns are first convolved with an HRF (Fig. 4), and then regressed against the fMRI data. However, before the GLM can be fit, one additional step has to be performed: the model predictions have to be matched to the participants' behavior on a trial-by-trial basis. That is, if a participant took 6 s to give a response on a particular trial while the model predicted 10 s, the model predictions have to be scaled back to the 6-s duration. Otherwise, we might be comparing predicted module activity to, for instance, a fixation screen (Borst et al., 2011; Gläscher & O'Doherty, 2010). To avoid such problems, all module activities are transformed linearly – either increased or decreased in duration – to create a perfect match to the behavioral data. This process is illustrated in Fig. 7. The resulting activity is used to fit the GLM.

Note that such a rescaling process could also have been applied in the ROI analysis in the previous section. This would have provided a perfect RT fit,⁹ ensuring that behavioral differences between the model and data did not cause the differences in the fMRI fit (Figs. 5 and 6). However, this has not been done traditionally, and could not be done in the case of testing true *a priori* model predictions (Borst, Taatgen, Stocco, & Van Rijn, 2010). We did not apply this rescaling process to the ROI analysis above, as we wanted to illustrate the connection between RT predictions and fMRI predictions.

Step 1. Collecting module activity. This step is identical to the one for the ROI analysis in Section 3.1. However, while for the ROI analysis a model is not necessarily run on the exact same trial sequence as the participants received, this is required for a model-based analysis, because matching predictions are needed for each trial. As before, open ACT-R, load the model, and type

```
(run-real-problems)
```

to generate all model predictions. See Section 3.1 for detailed instructions.

Step 2. Rescaling. To perform the linear transformation and recreate Fig. 7, open 'ModelBasedAnalysis/model_predictions.m'. At the start of this file, Fig. 7 is recreated to illustrate the rescaling process. First, the behavior of the model and the participant is read in, as well as the module activity for this particular trial. The module activities are then rescaled by increasing the durations and changing the onsets, on the basis of the RT-difference. This is done separately for the first key press and the time between the first key press and pressing the submit button. Note that the resulting model fits the RT data perfectly, as was the goal of this procedure.

After recreating Fig. 7, the rescaling procedure is applied to all trials of all participants, and the results are saved to SPM-compatible 'multiple condition'-files in the 'ModelBasedAnalysis/design_SPM' directory. The code for this is also included in

⁹ Naturally, the resulting RTs could *not* be used to constrain the model.

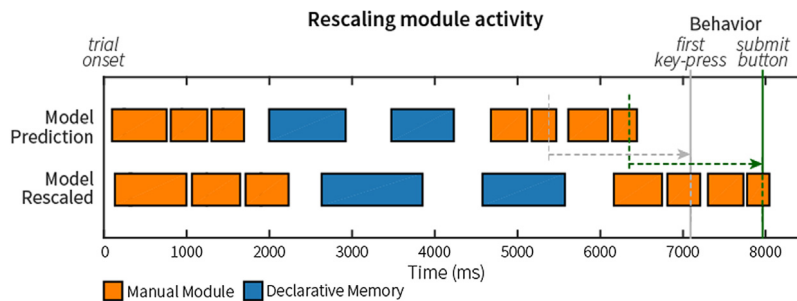


Fig. 7. Rescaling module activity for model-based fMRI analysis. In the first row, the *a priori* module predictions for the manual module (orange) and declarative memory (blue) are shown (cf. Fig. 4(b)). The model's key-presses are indicated by the dashed lines. In the second row, these module predictions have been linearly transformed to match the actual behavior on this trial (indicated by the solid lines). Note that this changes both the onsets as well as the durations of the module's activities. In this example the module activities were increased in duration, but in cases where the model's predictions were too slow, these activities were decreased in duration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

'model_predictions.m', starting at line 187. The resulting files contain four regressors that are used for the GLM: one for declarative memory, one for the problem state, one for the manual module, and one for the visual module. These regressors are described in three variables: names, onsets, and durations. The convolution with the HRF will be done automatically in SPM when the GLM is fit.

4.2. Fitting the GLM

To fit the GLM we used SPM8, which is a Matlab software package for analyzing fMRI data. The Matlab script 'ModelBasedAnalysis/model_based_analysis.m' uses SPM's batch system to fit a GLM for each subject separately, using the rescaled model predictions from the previous step, as well as movement regressors from realigning the scans. This takes between 20 min and 3 h 40 min (1–11 min per subject). For each subject, SPM will ask if you want to overwrite the subject, because we included the results of the analysis as well (we recommend choosing 'continue' for the first subject to observe the process, afterwards you can choose 'stop' to skip the rest of the subjects and move on to the second-level analysis). SPM then performs a second-level analysis (starting at line 104 of 'model_based_analysis.m'; this takes between 30 s and 6 min) and displays the results of Fig. 8. Note that these analyses can be performed with any fMRI analysis software and are not restricted to SPM.

4.3. Discussion

Fig. 8 shows the results for all four modules. At the top significant regions for declarative memory are shown. In essence, the fronto-parietal network was found, which is a sensible result as this network is known to be important for controlled episodic memory retrievals (Borst & Anderson, 2013; Cabeza et al., 2003; Cole & Schneider, 2007; Dosenbach et al., 2007; Spreng, Stevens, Chamberlain, Gilmore, & Schacter, 2010). ACT-R's predefined region for declarative memory is part of this network (around the inferior frontal sulcus, slice +28, see also Fig. 3(b)). However, the parietal part of the network (slices +40 and +52) is normally attributed to ACT-R's problem state module. This was not confirmed by the current model-based analysis: problem state activity was limited to the supplementary motor area, left inferior frontal gyrus, and left and right precuneus. As expected, the right-handed mouse movements of the model were mapped onto the left motor cortex, but also to the medial frontal gyrus and to a large area in the occipital cortex. Finally, the visual module mapped most strongly onto extended activation in the occipital cortex.

Although these results are generally sensible, they are not fully in line with ACT-R's predefined mapping and with results

of a recent meta-analysis of five model-based analyses (Borst & Anderson, 2013). Especially the results of the problem state module and the manual module are different than expected (note that the *t*-values were also much lower for those modules). This emphasizes the importance of proper experimental design: The current study was not designed for a model-based analysis, and the predicted activity was therefore very highly correlated between modules (Fig. 5), making it problematic to isolate a particular module. If one wants to apply model-based analysis to locate a certain model process, it is important to vary the use of this process while keeping other processes constant. Another option is to use a meta-analysis over different experiments, because it avoids the high inter-module correlations that often occur in a particular experiment.

5. Final remarks

In this paper we provided a tutorial of two different ways in which ACT-R models can be used with fMRI data. The predefined ROI analysis can be used to constrain ACT-R models and guide model development, while model-based fMRI analysis can be used to locate neural correlates of model processes. Note that in most situations one would not perform both analyses on a single dataset (at least not for the same modules): after adjusting a module based on an ROI analysis it would be obvious to find the ROI used for the adjustments in a model-based analysis. Finally, these analyses are not limited to ACT-R models. While model-based analysis was originally used for other types of models (e.g., Daw et al., 2011; O'Doherty et al., 2007; van Maanen et al., 2011), the ROI analysis could also be applied to other modeling paradigms. In addition, while we used Matlab and SPM as software in this paper, the same analyses could be performed with different software.

Acknowledgments

We thank Lael Schooler, Ioanna Katidioti, and Trudy Buwalda for testing the tutorial code. This research was supported by NWO Veni grant 451-15-040 awarded to Jelmer Borst, and by ERC-StG grant 283597 awarded to Niels Taatgen.

References

- Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, 29, 313–341.
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?*. New York: Oxford University Press.
- Anderson, J. R., Betts, S., Ferris, J. L., & Fincham, J. M. (2011). Cognitive and metacognitive activity in mathematical problem solving: prefrontal and parietal patterns. *Cognitive, Affective, & Behavioral Neuroscience*, 11(1), 52–67. <http://dx.doi.org/10.3758/s13415-010-0011-0>.

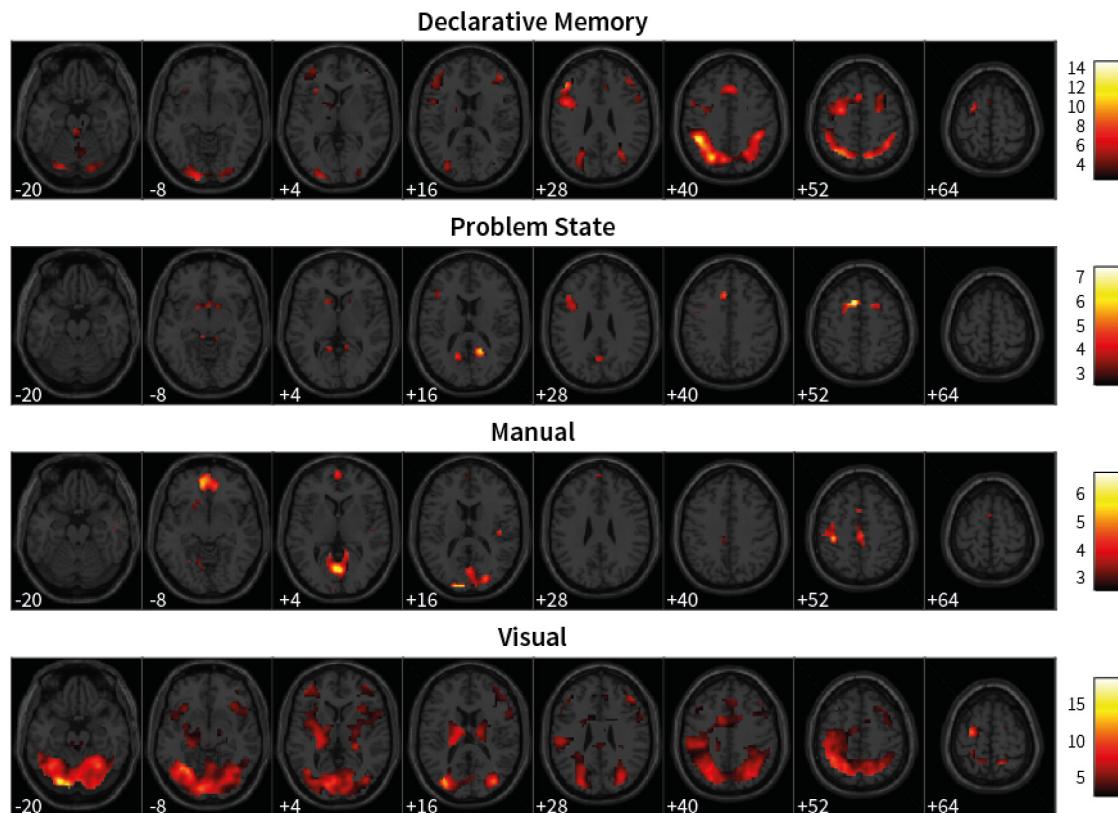


Fig. 8. Results of the model-based analysis, thresholded at $p < 0.01$ and at least 200 contiguous voxels.

- Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, 38, 341–380.
- Anderson, J. R., Fincham, J. M., Qin, Y., & Stocco, A. (2008). A central circuit of the mind. *Trends in Cognitive Sciences*, 12(4), 136–143. <http://dx.doi.org/10.1016/j.tics.2008.01.006>.
- Anderson, J. R., Qin, Y., Jung, K. J., & Carter, C. S. (2007). Information-processing modules and their relative modality specificity. *Cognitive Psychology*, 54(3), 185–217. <http://dx.doi.org/10.1016/j.cogpsych.2006.06.003>.
- Anderson, J. R., & Schooler, L. (1991). Reflections of the environment in memory. *Psychological Science*, 2(6).
- Borst, J. P., & Anderson, J. R. (2013). Using model-based functional MRI to locate working memory updates and declarative memory retrievals in the Fronto-Parietal network. *Proceedings of the National Academy of Sciences of the United States of America*, 110(5), 1628–1633. <http://dx.doi.org/10.1073/pnas.1221572110>.
- Borst, J. P., & Anderson, J. R. (2015). Using the cognitive architecture ACT-R in combination with fMRI data. In B. U. Forstmann, & E.-J. Wagenmakers (Eds.), *Model-based cognitive neuroscience*. New York: Springer.
- Borst, J. P., Buwalda, T. A., Van Rijn, H., & Taatgen, N. A. (2013). Avoiding the problem state bottleneck by strategic use of the environment. *Acta Psychologica*, 144(2), 373–379.
- Borst, J. P., Nijboer, M., Taatgen, N. A., Van Rijn, H., & Anderson, J. R. (2015). Using data-driven model-brain mappings to constrain formal models of cognition. *PLoS One*, 10(3), e0119673. <http://dx.doi.org/10.1371/journal.pone.0119673>.
- Borst, J. P., Taatgen, N. A., Stocco, A., & Van Rijn, H. (2010). The neural correlates of problem states: Testing fMRI predictions of a computational model of multitasking. *PLoS One*, 5(9), e12966.
- Borst, J. P., Taatgen, N. A., & Van Rijn, H. (2011). Using a symbolic process model as input for model-based fMRI Analysis: Locating the neural correlates of problem state replacements. *NeuroImage*, 58(1), 137–147. <http://dx.doi.org/10.1016/j.neuroimage.2011.05.084>.
- Borst, J. P., Taatgen, N. A., & Van Rijn, H. (2015). What makes interruptions disruptive? A process-model account of the effects of the problem state bottleneck on task interruption and resumption. In *Proc. CHI*. Seoul, Korea: ACM Press.
- Brett, M., Anton, J. L., Valabregue, R., & Poline, J. B. (2002). Region of interest analysis using an SPM toolbox. *NeuroImage*, 16(2).
- Cabeza, R., Dolcos, F., Prince, S. E., Rice, H. J., Weissman, D. H., & Nyberg, L. (2003). Attention-related activity during episodic memory retrieval: a cross-function fMRI study. *Neuropsychologia*, 41(3), 390–399.
- Chen, M. C., Anderson, J. R., & Sohn, M. H. (2001). What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *Human factors in computing systems, CHI'01* (pp. 281–282). Seattle, Washington: ACM.
- Cole, M. W., & Schneider, W. (2007). The cognitive control network: Integrated cortical regions with dissociable functions. *NeuroImage*, 37(1), 343–360. <http://dx.doi.org/10.1016/j.neuroimage.2007.03.071>.
- Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., & Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6), 1204–1215. <http://dx.doi.org/10.1016/j.neuron.2011.02.027>.
- Dosenbach, N. U. F., Fair, D. A., Miezin, F. M., Cohen, A. L., Wenger, K. K., Dosenbach, R. A. T., ..., & Petersen, S. E. (2007). Distinct brain networks for adaptive and stable task control in humans. *Proceedings of the National Academy of Sciences of the USA*, 104(26), 11073–11078. <http://dx.doi.org/10.1073/pnas.0704320104>.
- Forstmann, B. U., Dutilh, G., Brown, S., Neumann, J., von Cramon, D. Y., Ridderinkhof, K. R., et al. (2008). Striatum and pre-SMA facilitate decision-making under time pressure. *Proceedings of the National Academy of Sciences of the USA*, 105(45), 17538–17542. <http://dx.doi.org/10.1073/pnas.0805903105>.
- Friston, K. J., Ashburner, J. T., Kiebel, S. J., Nichols, T. E., & Penny, W. D. (2007). *Statistical parametric mapping. The analysis of functional brain images*. Academic Press.
- Gläscher, J. P., & O'Doherty, J. P. (2010). Model-based approaches to neuroimaging: combining reinforcement learning theory with fMRI data. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(4), 501–510. <http://dx.doi.org/10.1002/wcs.57>.
- Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, 104(1), 3–65.
- Newell, A. (1973). You can't play 20 questions with nature and win: projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual information processing* (pp. 283–308). New York: Academic Press.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Nijboer, M., Borst, J. P., Van Rijn, H., & Taatgen, N. A. (2016). Contrasting single and multi-component working-memory systems in dual tasking. *Cognitive Psychology*, 86, 1–26. <http://dx.doi.org/10.1016/j.cogpsych.2016.01.003>.
- O'Doherty, J. P., Hampton, A., & Kim, H. (2007). Model-based fMRI and its application to reward learning and decision making. *Annals of the New York Academy of Sciences*, 1104, 35–53. <http://dx.doi.org/10.1196/annals.1390.022>.
- Pitt, M. A., & Myung, I. (2002). When a good fit can be bad. *Trends in Cognitive Sciences*, 6(10), 421–425.
- Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, 107(2), 358–367.
- Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, 48(2), 362–380.
- Schneider, D. W., & Anderson, J. R. (2012). Modeling fan effects on the time course of associative recognition. *Cognitive Psychology*, 64(3), 127–160. <http://dx.doi.org/10.1016/j.cogpsych.2011.11.001>.

- Shiffrin, R. M., Lee, M. D., Kim, W., & Wagenmakers, E.-J. (2008). A survey of model evaluation approaches with a tutorial on hierarchical Bayesian methods. *Cognitive Science*, 32(8), 1248–1284. <http://dx.doi.org/10.1080/03640210802414826>.
- Spreng, R. N., Stevens, W. D., Chamberlain, J. P., Gilmore, A. W., & Schacter, D. L. (2010). Default network activity, coupled with the frontoparietal control network, supports goal-directed cognition. *NeuroImage*, 53(1), 303–317. <http://dx.doi.org/10.1016/j.neuroimage.2010.06.016>.
- Stocco, A., & Anderson, J. R. (2008). Endogenous control and task representation: an fMRI study in algebraic problem-solving. *Journal of Cognitive Neuroscience*, 20(7), 1300–1314. <http://dx.doi.org/10.1162/jocn.2008.20089>.
- Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, 120, 439–471.
- Turner, B. M., van Maanen, L., & Forstmann, B. U. (2015). Informing cognitive abstractions through neuroimaging: the neural drift diffusion model. *Psychological Review*, 122(2), 312–336. <http://dx.doi.org/10.1037/a0038894>.
- van Maanen, L., Brown, S. D., Eichele, T., Wagenmakers, E.-J., Ho, T., Serences, J., et al. (2011). Neural correlates of trial-to-trial fluctuations in response caution. *The Journal of Neuroscience*, 31(48), 17488–17495. <http://dx.doi.org/10.1523/jneurosci.2924-11.2011>.
- Wintermute, S., Betts, S., Ferris, J. L., Fincham, J. M., & Anderson, J. R. (2012). Brain networks supporting execution of mathematical skills versus acquisition of new mathematical competence. *PLoS One*, 7(12), e50154. <http://dx.doi.org/10.1371/journal.pone.0050154>.