Journal of Artificial General Intelligence 2(2) 20-37, 2010 Submitted 2009-12-04; Revised 2010-06-10

Metacognition and Multiple Strategies in a Cognitive Model of Online Control

David Reitter REITTER@CMU.EDU
Department of Psychology
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213, USA

Editor: Christian Lebiere, Cleotilde Gonzalez, and Walter Warwick

Abstract

We present a cognitive model performing the Dynamic Stocks&Flows control task, in which subjects control a system by counteracting a systematically changing external variable. The model uses a metacognitive layer that chooses a task strategy drawn from of two classes of strategies: precise calculation and imprecise estimation. The model, formulated within the ACT-R theory, monitors the success of each strategy continuously using instance-based learning and blended retrieval from declarative memory. The model underspecifies other portions of the task strategies, whose timing was determined as unbiased estimate from empirical data. The model's predictions were evaluated on data collected from novel experimental conditions, which did not inform the model's development and included discontinuous and noisy environmental change functions and a control delay. The model as well as the data show sudden changes in subject error and general learning of control; the model also correctly predicted oscillations of plausible magnitude. With its predictions, the model ranked first among the entries to the 2009 Dynamic Stocks&Flows modeling challenge.

Keywords: Cognitive Modeling, Control, Metacognition, Instance-Based Learning, Blending, ACT-R

1. Introduction

Humans quickly learn to function well in new dynamic environments. They can learn to rapidly and precisely react to environmental changes, such as wind speed when flying aircraft, or water currents when steering boats. Control tasks emulate such situations, and cognitive models can predict and explain a variety of the real-time behavior that humans display in those tasks. For example, models explain gradual and sometimes sudden performance improvements, correlations of task error with task difficulty and even subject-specific variability. Despite our general ability to control such systems, humans display a variety of possibly irrational behavior, with sudden and not gradual changes in their response to environmental change that point to the presence of symbolic strategies rather than just probabilistic learning.

In the experiment underlying the Dynamic Stocks and Flows (DSF) challenge (Lebiere, Gonzalez, and Warwick, 2009), human participants were asked to control the water level in a simulated water tank by manipulating an outflow valve. Subjects reacted to a changing, independently determined inflow into the tank. The inflow was a function of time or iteration and unknown to the participant. In the DSF experiment that guided the development of the model,

the inflow function was manipulated across four conditions, combining linear and non-linear, decreasing and increasing inflow functions. The core of the participant's strategy in the water-tank task is to estimate future inflow figures, given the discrete samples from past attempts, and counteract the inflow by setting the outflow valve correctly.

The subject-by-subject analysis of the DSF data showed sudden, marked changes in behavior: often, subjects began to control the water flow effectively after a number of failed trials. The data also suggested that subjects approached the task differently. Furthermore, their strategies appeared to vary with the complexity of the external water inflow function. Purely subsymbolic learning mechanisms fail to explain such behavior. Thus, a cognitive model of this control task needs to address not just gradual learning, but also switches in task strategy. To explain these effects, our model describes explicitly how subjects manage competing task strategies.

How well a cognitive model explains a specific task is less important than whether it describes general cognitive mechanisms that transfer to other tasks or situations. As is typical for the cognitive modeling paradigm, we assume that human solutions to real-life problems emerge from a combination of general mechanisms: the core learning mechanisms embodied in a cognitive architecture, as well as decision-making strategies that are common to many cognitive models. The cognitive model presented here covers the specific DSF experiment with its specific learning and decision-making processes. It implements several strategies to deal with the basic control task (the first of two control layers in the model), as well as the means to rank and select those strategies according to their appropriateness in a given situation (the second, *metacognitive* layer). (Implicit or explicit reflection on one's behavior is often called *metacognition*.) This monitoring approach reflects the subjects' behavior and the variation between subjects. Our contribution is not primarily concerned with the control of an external variable in the strategy layer, but with *Cognitive Control* in the metacognitive layer, where the success of cognitive strategies is monitored and optimized in reaction to environmental changes.

The combination of a validated cognitive architecture and the metacognitive layer allowed the model to generalize well to further control tasks such as the evaluation conditions in the DSF competition. These experimental conditions retained the water tank paradigm, but introduced discontinuous water inflow functions and delays between a subject's outflow selection and its effects. Generally, a model's failure to generalize may be caused by *overfitting*, that is, biases introduced by automatic parameter estimation or manual specification by the modeler. In the model, we address such situations by underspecifying some of the algorithmic portions of the model (in particular the timing) and arriving at an unbiased estimate of these from the sample data. Cognitive models that can generalize may produce online measures important to many applications. They may, for instance, estimate cognitive load in control tasks at different points in the learning curve, or predict interference from multitasking.

The remainder of this article relates our model to existing literature (Section 2). We outline an exploratory analysis of the data (Section 3) that guided the design of the model, whose cognitive foundations are explained in Section 4.1, followed by a discussion of the specific modeling methodology designed for underspecification (Section 4.2). The description of the model with a discussion of our view of metacognitive processes can be found in Sections 4.3–4.7. To evaluate the model, we describe both its performance on development and on the unseen test data (Section 5). The latter puts model performance into perspective by comparing it to competing models in the DSF challenge.

2. Studies of Control Problems and Cognition

The water-tank task requires control of an external variable; however, the model we present also describes control at a metacognitive level choosing between task strategies. Indeed, real-life control tasks almost always occur in the context of other cognitive demands. In many cognitive and cognitively inspired models, multiple descriptive levels of decision-making represent strategic planning and micro-control decisions. Jones et al. (1999), for instance, describe a system that is able to simulate pilots controlling a large number of aircraft and successfully flying missions. Their system requires the use of thousands of rules, formalizing extensive experience. Some of those rules relate to the actual online control task, while the majority of their model implements the common companions of online control tasks: higher-level planning (e.g., navigation, strategic decisions) and the execution of mission goals. Executive function has also been extensively studied in connection with control, for instance, in driving (Salvucci, 2006), whose models can predict the difficulty of a primary driving task in relation to other, interrupting tasks and simulate multitasking.

These models assume knowledge about the environment, such has how an aircraft or a car react to control input, or how the outflow valve setting affects the water level in our tank. In contrast, other models can avoid such specialization. Gonzalez and Lebiere (2005) model the Sugar Factory Task (Berry and Broadbent, 1987) with very few rules using instance-based learning (Gonzalez, Lerch, and Lebiere, 2003), where subjects do not create accessible task knowledge. Instance-based learning is similarly used in some strategies (*estimation*) contained in our model, where we assume little explicit knowledge about the function governing environmental inflow or the water tank. An alternative class of strategies (*calculation*) encode more precise assumptions about how inflow changes and how the outflow counteracts the inflow.

In experiments, much initial task knowledge is communicated in instructions to subjects, or is already available to expert subjects as explicitly accessible, declarative information. The comparison of several potential models can reveal the kind of knowledge on which experts rely. Ball et al. (2003) examine on-line control in a series of instrument flight tasks, contrasting models that take different sources of aviation knowledge into account. Long-term acquisition of expertise in control tasks has been studied with tasks like the Space Fortress game (Mané and Donchin, 1989), which is a complex game that again involves executive function and various sensory input and motor actions. Unsurprisingly, the most knowledgeable models in Ball's study and the best-trained subjects in the Space Fortress experiments reach the highest scores.

These models primarily concern well-practiced tasks. However, how does learning proceed without much explicit knowledge? What are the early stages of learning environmental control? The Dynamic Stocks and Flows task investigates this with a relatively simple control problem, where little high-level knowledge is sufficient to describe the environment. The function governing environmental inflow is not described to subjects – it has to be inferred through experience. The data that our study is based on concern early phases of learning. In our model, rapid learning occurs the continuous evaluation of task strategies, allowing the model to develop preferences.

An important property of decision-making in the predictions of the model as well as in the water-tank data in some experimental conditions are oscillations between alternative decisions, such as task strategies or water inflow settings. Real-world control problems often lead to similar oscillations of the controlled variables. An example of a basic control task is temperature adjustment: Consider the case of somebody taking a hot shower. Suddenly, the water turns a little colder. Fearing much colder water soon, the person will overcompensate by turning up the hot

water. During a few oscillations, the water temperature converges to the desired level again. Similar examples can be made of a weaving car or the well-known phenomenon of pilot-induced oscillations in the control of aircraft. It is possible that unexpected external influences (e.g., road conditions or an aircraft entering ground effect during landing) lead to a misestimation of the effect and an increase the likelihood of such oscillations. The results presented here speak to the hypothesis that a delay between control input and observed reaction causes control oscillations. Oscillations are well-known from decision-making and attitude research (Fink, Kaplowitz, and Hubbard, 2002). In their study, subjects changed their opinions about a subject matter regularly. Fink et al. note that cognitive frameworks tend to come with a level of inertia in their choices. This is also true for the architecture used in our model: ACT-R (Anderson et al., 2004) provides a retrieval mechanism biased towards the recent past examples. It is easy to see how this can lead to a model underestimating an increasing environmental inflow in the water tank experiment. Fink et al. find that none of the oscillatory trajectories of their subject's attitudes showed constant periods; also, some amplitudes increased, while others ended without evidence of gradual amplitude damping. In the context of our two-layer model, we analyze cases of suddenly successful task execution as the result of a strategy switch. The oscillations in control tasks happen when the model is unaware of delays between changing a valve setting and that setting taking effect. Surprised by the lack of response to its control input, it overreacts, eventually leading to counter-corrections. The concrete predictions (compatible with Fink's results), are described in Section 5.2.

Humans are obviously capable of limiting oscillations, given that they can reliably learn directional control. They even master control problems that involve many interacting variables, such as altitude and airspeed control of an airplane. Long-term learning processes might be modeled through a damping mechanism that can be implemented as an additional constraint on the metacognitive level in the model.

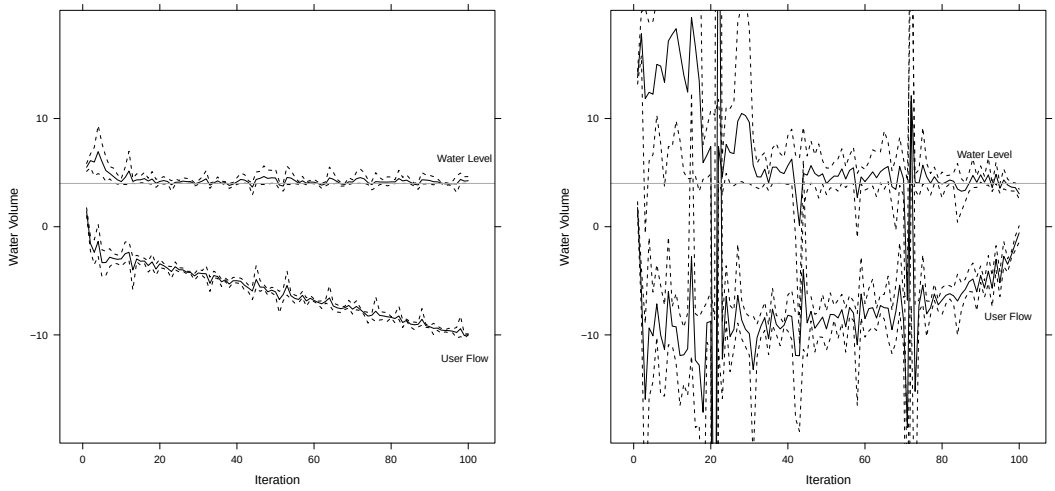
3. Exploratory Analysis of the Development Data

The data available for development of the model resulted from experiments with human subjects in two by two conditions as follows (*development conditions*). The experimental design manipulated the environmental inflow of water to the tank according to linearly and non-linearly, increasing and decreasing functions of the iteration step. Plotting the data as shown in Figures 1(a) and 1(b) reveals a number of effects that informed the model, which will be discussed in the following.

The task given to the subjects was to keep the water level in the tank at a given level. Thus, subjects had to anticipate changes to the next inflow level and produce an outflow valve setting to compensate for it. The figures show the water level (upper graphs) as well as the chosen valve setting (lower graphs, a negative value to indicate outflow). First, we look at subject performance in terms of difference between actual and target water levels and in terms of the chosen outflow compared to the actual inflow. (Actual and target water levels should match, ideally, and the subject-chosen outflow should counteract the environmental inflow). In the analysis and the model, we will collapse environmental outflow and inflow into one variable (*EnvirFlow* = *EnvirInFlow* - *EnvirOutFlow*) and do the same for outflow and inflow (*UserFlow* = *UserInFlow* - *UserOutFlow*).

The overall error of *UserFlow* is greater in the non-linear condition than in the linear one (Figures 1(a) and 1(b))[1]. Second, error is greater during the early phase of the experiment than

1. One subject stood out by causing large deviations at a late stage in the non-linear increase condition; that subject's data were excluded from this observational analysis.



(a) linear increasing environmental inflow (b) non-linear decreasing environmental inflow

Figure 1: Subject data from two of four development conditions; we plot the water amount in the tank (upper graph, target is 4.0) and the user's valve setting (lower graph, neg. values indicate outflow). Aggregate data over all subjects; dashed lines show bootstrapped 95% confidence intervals over all subjects.

toward the end. This observation holds both for the non-linear increasing case, where the rate of change in environmental inflow is high at the beginning, and for the non-linear decreasing case, where the rate of change is greatest at the end. Thus, there seems to be an effect of practice. The linear case appears to be an easier task for the subjects.

An exploratory analysis of the data using linear regression suggested that the variables visible to the subject (environment inflow, water level) and the resulting needed correction correlate with the error seen in the subject's valve setting. Despite the observed sudden changes in error, there was no reliable difference in how accurately subjects reacted to the input over the course of the 100 steps of each experiment. Taking a number of observable variables as predictors in the model, we also examined whether the addition of a subject-specific grouping factor could improve the fit of the regression. Estimating subject-specific parameters for just the observable variables did not improve the models significantly; i.e., subjects did not systematically under- or overestimate environmental inflows.

However, aggregate data are deceiving. Plotting the data separately for each subject reveals that many of them learned to handle the linear cases extremely well; there is initial error, but then subjects usually arrive at the target water level with mathematical precision. Once or twice per subject, the water level changes for a few steps–often, dramatically. This causes the impression of a constant variance in the data aggregated across subjects. A good model will approximate the behavior of subjects separately.

A subject-by-subject, qualitative analysis of the DSF data reveals that subjects do adopt varying strategies. One subject, for instance, estimates the linearly increasing inflow relatively well without consistent errors. Large deviations occur until approximately step 25 (out of 100), after which the

inflow is well estimated and also well compensated. Another subject seems to calculate the inflow trend precisely, making successful corrections from step 5, and miscalculating only once (which leads to short oscillation for just a few steps). The majority of subjects seem to have chosen a precise method for the linearly increasing task. For the non-linear example, such a strategy is less likely to be successful and also less commonly seen. Two subjects, for instance, show regular oscillations throughout the experiment.[2] A purely linear model with comparatively little dynamic behavior is not a satisfactory level of description for such data. Instead, these observations motivated the multi-strategy approach of the model.

4. Modeling Metacognitive and Cognitive Control

To discuss the strategies embodied in the cognitive model, we first introduce the cognitive architecture within which the model is situated. The architecture enforces plausibility by constraining the computational resources of the model, but it also supports generalizability, as all components of the model have been independently motivated.

4.1 Cognitive Architecture

The "Adaptive Control of Thought–Rational" theory (ACT-R, Anderson et al., 2004) is a well-validated cognitive framework that primarily contributes a detailed theory of memory to the model. The ACT-R theory combines symbolic and subsymbolic reasoning and compartmentalizes cognitive function, providing modules to store and retrieve information in *declarative memory* and to track task goals and maintain working memory in *buffers*. Other modules, whose function is not further specified in our cognitive model, execute motor functions and gather auditory or visual sensory information.

Information may be sent to or retrieved from declarative memory (DM). Accessibility, as encoded by the *activation* of a piece of knowledge (*chunk*, Figure 2), is determined by the pattern of prior retrievals and contextual cues, plus noise. Base-level activation is boosted by recency and frequency of access and decays logarithmically over time (*base-level learning*). The mechanisms with which DM learns and accesses chunks follow the idea that information most likely to be used in the future will also be most readily accessible. That is, the learning mechanism has evolved to suit the environment–a core idea of Rational Analysis (Anderson, 1991).

We use techniques based on declarative memory to derive novel decisions from a blend of previous experiences. *Instance-based learning* (IBL) and Blending are our methods to aggregate experience (Gonzalez, Lerch, and Lebiere, 2003): IBL formulates the idea to store rated episodes (Figure 2), and *Blending* provides means to merge such stored experiences into new decisions. Our cognitive model makes use of these techniques in both of its decision-making layers.

4.2 Accountable Modeling in Converging Cognitive Frameworks

The ability to not just explain known, but also predict novel data requires a model to be maximally faithful to the invariant cognitive abilities of human subjects (A), and to the invariant task properties (B). But it also requires the model to be minimally adapted to the task properties that may change. In other words, a predictive model must avoid overfitting and biases, which can be introduced by

2. We refer to the following subjects in the competition dataset: *t11* (linearly increasing inflow), *t26* (precise calculation), *t29* and *t39* (oscillations).

episode-#42		*(Activation: 1.0)*
STRATEGY	'calculation-method-3'	
ERROR	'0.4'	
episode-#43		*(Activation: 0.5)*
STRATEGY	'estimation-method'	
ERROR	'0.15'	
episode-#44		*(Activation: 0.25)*
STRATEGY	'calculation-method-3'	
ERROR	'0.8'	

Figure 2: Some ACT-R chunks as learned by the model.

automatic parameter estimation or by the (manual) design of the model. To avoid biases and reduce overfitting, we used a modeling methodology we call *accountable modeling*. This methodology produces a model that describes the subject's task strategies at different descriptive levels.

We abstract away from portions of the model that cannot be motivated by data and focus on only those non-deterministic, sub-symbolic aspects of the model, which explain variance in the (development) data. The model does not specify within the ACT-R mechanisms, for instance, how simple arithmetic calculations are performed. Concretely, that means that we do not use ACT-R procedures (defined in procedural If/Then rules) or ACT-R transient data structures (buffers) for these algorithms. Similarly, the model does not specify perceptual or motor processes to interface with the experimental environment. For the latter, we estimate timing from the development data set. We assume that this estimate is less biased than it would be if it was derived timing from first principles, i.e., from perceptual, motor and procedural ACT-R modules, whose predictions depend on the way we represent information and implement task execution sequences. Timing is relevant for the predictions of the model due to the activation decay of information stored in declarative memory. While we do not explicate the model's strategies in form of fine-grained ACT-R procedural rules, the model remains constrained by the ACT-R 6 theory. Information storage follows the principles of declarative memory, where recency and frequency determine the accessibility of limited-size chunks of knowledge. These principles form the core of the instance-based learning approach.

The model is implemented with the *ACT-UP* toolbox. For a description of ACT-UP and an evaluation of its validity (w.r.t. ACT-R), its efficiency and scalability, refer to Reitter and Lebiere (2010). During the development of the water-tank model, ACT-UP was instrumental in allowing us underspecify parts of the model and to explore a number of possible strategies by concentrating on the higher-level interaction of strategies and metacognition.

4.3 A Blend of Explicit Knowledge Encodes Procedural Skills

Learning to control the water level in the DSF experiment can be seen as skill acquisition. The prevailing top-level perspective on learning in ACT-R includes two components relevant to two stages of skill encoding. The first is declarative memory (DM), which retrieves pieces of knowledge, an initially slow process that becomes faster and more reliable with repeated knowledge retrieval. The second is procedural memory, in which competing production rules are kept that specify the serial execution of a model's strategy. Rules may be in competition with each other, and those rules

that lead to successful task completion are preferred (a form of learning related to reinforcement learning). Quickly executed production rules often provide a shortcut to slower DM retrievals, and procedures can be acquired over time if sequences of rules are repeatedly and successfully executed. Procedural learning is thought to become visible at late training stages (Taatgen and Lee, 2003).

In the water-tank task, we assume to see the early stages of this learning process, that is, base-level learning affecting the retrieval of information from DM. Thus, the model presented here implements learning and strategy choice within the first skill acquisition stage, that is, the more explicitly accessible DM. Subjects are not considered to become highly trained during the 100 steps of the DSF experiment. Section 5.2 will point out empirical advantages and disadvantages of a similar model that implements multiple strategies using utility learning.

At this point, we could implement specific mechanisms to deal with the kinds of environmental inflow functions we expect to see. For instance, we could write procedures or memory items to deal with linearly decreasing water flows. Instead, we define a general-purpose learning mechanism to be able to deal with changing environmental conditions. We adopt *instance-based learning* (Gonzalez, Lerch, and Lebiere, 2003), in which experienced situations are encoded as knowledge pieces and stored in DM. To make a decision based on such experience, we retrieve the most similar, recent experience as a chunk from DM. This mechanism would work if the model was to observe correct (and possibly incorrect) decisions first before making its own decisions.

In the water-tank task, the model has to learn from scratch without the benefit of initial observations. Also, the decisions taken are never exactly the same as those previously made. For instance, we could have selected 3 and 7 as outflow valve positions, but now require a setting of 4 to correctly counteract the inflow. In such cases, *Blending* (Gonzalez, Lerch, and Lebiere, 2003) lets the model bootstrap a new estimate based on prior experience. Blending not only constructs estimates, but also a new strategy decision as a combination of prior ones. Consider the chunks shown in Figure 2. Blending can combine two of these exemplars (episode-#42 and episode-#44) into a new decision. When blended, each continuous attribute in the chunks, such as a value indicating inflow, is calculated as a weighted mean between the values in each chunks. The weights of the chunks in this blending average are proportional to the retrieval probability of the chunk, which, in turn, is quantified by the chunk's activation in DM. As described earlier, activation depends on recency and frequency; thus, the produced estimate is biased towards recent valve settings and their associated errors.[3]

4.4 The Metacognitive Layer

As discussed in the introduction, the interest in the Dynamic Stocks and Flows task lies in its similarity to general control problems. However, there are significant differences between the water-tank task and a control problem involving rapid estimates and analogue inputs using, e.g., a steering wheel. First, the experimental setup requires entering numbers, i.e., subjects make discrete decisions as opposed to giving a continuous, graded motor response. Second, the subject-specific data strongly suggests that many subjects acquire the slope of a linear environmental inflow very quickly and then apply it very accurately with only minimal deviations. Third, the design does not introduce noise in the crucial controlled variable (environmental in/outflow) visible to the subject.

3. The weight of each chunk i is determined by $w_i = e^{a_i/t}$ (Note: Chunk activation in ACT-R is in log-space). a_i is the chunk activation of chunk i, t is the blending *temperature*, a parameter moderating the influence of less-active chunks and balancing explorative vs. conservative behavior.

These observations are compatible with strategies that treat the task as an *arithmetic* problem rather than as a sensory-motor control problem. Still, this does not apply to all conditions, and our analysis of the DSF data detected sudden performance changes, suggesting that subjects do indeed avail of several different strategies, beyond the precise calculation. Besides arithmetic calculation, our model may alternatively *estimate* the trend in water inflow.

Metacognition refers to processes that monitor and regulate the strategies: they evaluate the outcome of each decision with respect to the stated goals and relate the level of success to the strategy. For the water-tank task in its basic form, the time point when to evaluate is clearly defined, namely at each iteration for the strategy chosen previously. The stated goal is also easy to quantify: it is the discrepancy between achieved and target water levels. This is the way the model implements metacognition as a layer that observes the current success, combines it with the previously chosen strategy and stores this as a piece of evidence in memory. It then blends previous evidence to evaluate each strategy, retrieving the best one according to previous success, but also other criteria that are known to influence retrieval performance, such as recency.

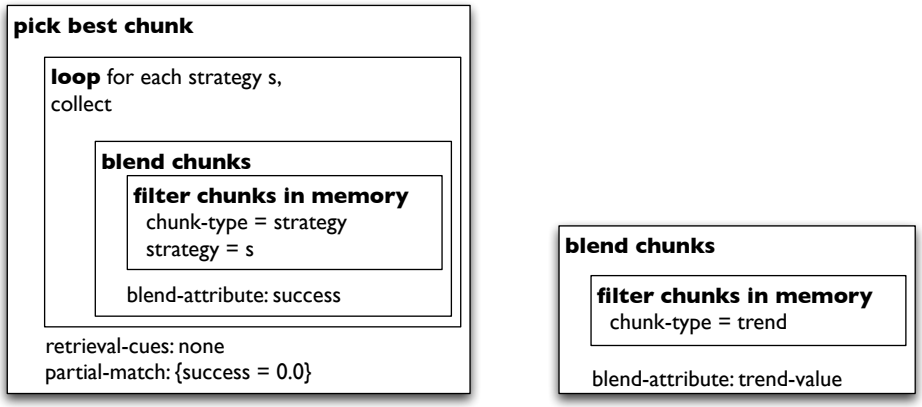
Notably, monitoring and choice of strategies is not necessarily a conscious process in the model; learning individual instances of evaluated success does not imply that these instances reach activation levels that make them explicitly retrievable (Reder and Schunn, 1996; Anderson, 1996). However, an aggregate judgment may well be accessible after a few trial runs. Conceivably, a subject would be able to answer the questions: does strategy X work, and does it work better than strategy Y?

Any plausible strategies for the task need to produce a prediction of the next environment inflow value (w'_1). A reasonable resulting setting outflow valve setting is $w + w'_1 - T$, given the target level T. We observe what is visible on the screen: the current water level (w), the current environmental inflow (w') and the current valve setting.

An initial set of five strategies is provided within the model. Specifically, the strategies are *estimation*, and four variations of *calculation*. The choice of strategy, determined by the metacognitive process, is guided by prior experience. The model produces a new prediction during each iteration of the task. In this case, it always expects the water level to be 4.0 after application of the new outflow value. The model monitors the success of the chosen strategy after its execution. It commits to memory, at each iteration, an explicit ACT-R chunk encoding the type of strategy (estimation, or one of the four calculation strategies) and the success criterion. The success criterion is defined as the deviation from the target water level after using the given type of strategy. Figure 2 provides an example of such a chunk. This mechanism lets the model acquire instance-based knowledge. Instances contain descriptions of the strategy and the success criterion.

We assume a fixed cost of 2.5 seconds to monitor the strategy. While we consider delays as underspecified with respect to temporal predictions, they are relevant to the preference of the model for chunks learned more recently, as activation decays with time. We will estimate another main delay parameter from the empirical data provided.

To choose a strategy, we need to evaluate the quality of each strategy. The retrieval mechanism chooses a chunk out 5 blended chunks: one for each strategy. To estimate the quality (success) of the strategies, the model calculates a blended chunk for each known strategy. Blending takes into account prior experience and implies a bias for more recent experiences. The strategy retrieval mechanism is described in Figure 3(a). A *best-chunk* function carries out retrieval, choosing the best of a given set of chunks; *blend* provides those chunks, as new, chunks blended from prior



(a) Strategy decision-making (b) Estimating the trend

Figure 3: Instance based learning and blending and decision-making in two parts of the model.

experiences (one for each known strategy). The model then retrieves the most active one of the blended strategies.

The selection of the best blended experience chunks is guided by the ERROR value: obviously, the model aims to minimized error. ACT-R models realize such an optimization during retrieval. When retrieving chunks from declarative memory, models specify a template with a set of attribute-value pairs to select a set of eligible chunks. In the absence of a perfectly matching chunk, the declarative memory module retrieves a chunk that best matches the template (*partial matching*). The activation of a partially matched chunk will increase with its similarity to the template. A request for a chunk with ERROR 0 will boost the activation of the best chunks. From the chunks in Figure 2, *episode-#43* might be retrieved, as it has an ERROR value closest to 0, and also a reasonable high base-level activation. To select a strategy, the model combines partial matching with blending. Blending first determines the model's overall assessment of the strategies, producing a blended chunk for each strategy. Then, the model selects between them using partial matching, retrieving the one with the lowest estimated error. In this process, noise leads to explorative behavior; at the beginning of the task, we find that noise has a stronger effect than towards the end, when chunk activations have settled.

After executing a strategy, an additional 6 seconds are spent, reflecting delays present in the experimental environment. Empirical timing data from the experiment was not available for this value, but it is plausible given the experimental system used to collect the DSF data.

4.5 Calculation Strategies

To calculate the inflow, subjects have to explicitly comprehend the workings of the simulated water tank and the gradual change in environmental inflow. They need to translate this into an arithmetic problem to calculate the needed valve adjustments in each iteration. First, the model calculates the trend of the water inflow; second, it adds the current water level to the expected trend (to predict the next step), and calculates the necessary inflow/outflow in order to reach the target water level.

We expect that a number of subjects, at least initially, arrive at a flawed formula describing the next inflow. Thus, the model includes four kinds of calculation strategies, whereas only one is appropriate to the task. The other ones reflect faulty inflow/outflow calculations: for instance, one of them assumes that the given environmental inflow needs to be subtracted from the water level to obtain the next level. The strategies attempt calculation of the valve setting (*UserFlow*, i.e., negative user outflow) on the basis of the visible current water level w and inflow w'. Generally, the calculation strategies remember the inflow value in a buffer, i.e., the model can recall the value precisely during the next iteration to estimate the trend of the inflow, w''. In the following strategies, T identifies the target water level.

- $UserFlow_A(w, w', w'') = T - (w + (w' + w''))$
- $UserFlow_B(w, w', w'') = T + (w + (w' + w''))$
- $UserFlow_C(w, w', w'') = T - (w - (w' + w''))$
- $UserFlow_D(w, w', w'') = T + (w - (w' + w''))$

Only $UserFlow_A$ is an acceptable valve setting. All strategies (calculation and estimation, see next section) are equally likely to be chosen initially: their activation is the same.

The model abstracts away from actual mental arithmetic (see McCloskey and Lindemann, 1992; Vicuso, Anderson, and Spoehr, 1989; Lebiere, 1999; Campbell, 2008). We expect that further variance in the model may be explained by a more detailed model of mental arithmetic. Still, the implementation of the addition and subtraction procedures assumes more reliable addition (3% error probability) than subtraction (6%); this parameterization was guided by the training data, which have contrasting conditions with both increasing and decreasing environmental inflows.

4.6 Estimation Strategy

Estimation is a fuzzy analogue to the calculation strategy. The model estimates the inflow trend. It then predicts the next valve setting using the current environmental inflow, the estimated future trend for the inflow, the target water level. This estimate of the future trend is a blend of previously observed trends. In each iteration, it commits to memory the observed trend, which arises from the difference between expected and observed water levels. This means that the model purposefully fails to distinguish between actual changes in trend, and simple calculation or estimation errors during the previous steps.

Estimation differs from calculation in that exact values are not retained in buffers, but stored in and retrieved from memory. Environmental inflow and water level are shown on the screen, so the crucial variable to be estimated is the first derivative of the inflow (*trend*). We store the environmental inflow in memory and retrieve the last stored value (noise may lead to misretrievals). We then determine the difference between this last stored inflow and the current shown inflow, learning this value in a *trend chunk*, which is committed to declarative memory as an ACT-R chunk. The current estimate of the trend is then a blend of previously learned trends (Figure 3(b)). As explained before, blending results in a weighted average of previous estimates, with more recent ones carrying more weight. The estimation takes a fixed 5 seconds in the model.

The estimation strategy has two notable characteristics. First, unlike the human subjects in the experiment, it consistently *over*estimates inflows in an underlying linearly decreasing function;

it *under*estimates them in an underlying linearly increasing function, as its prediction is directly derived from past experience. Second, a model containing only the estimation strategy may deal with non-linear functions worse than the humans do. The strong drop in inflow in the final steps in the non-linear decreasing condition caused the model to vastly overestimate inflow (i.e., it did not notice the drop). Human subjects did not seem to make such systematic errors in this case. However, note that Bott and Heit (2004) present data from a subject in their study on learning of functions, where the subject extrapolated from recent history and lagged in his or her response, especially during the initial learning blocks.

4.7 Parameter Optimization

We optimized a small set of model parameters across a relatively coarse grid of plausible values. Model-specific parameters that influence the outcome include assumed certain durations of the tasks; longer delays imply lower activations at retrieval, rendering noise relatively more important. Using the development data only, we estimated the delay after setting valve $d_1 = 8$ seconds (in range $[3; 10]$) as well as strategy execution time for estimation $d_e = 5$ seconds (range $[2; 10]$). Execution time for calculation was held constant at $d_c = 2 * d_e$. Blend temperature was set to $t = 0.5$ (optimization between values 0.5, 1.0). Relevant framework parameters (specifically transient noise and base-level learning delay) were not optimized and kept at values typically found in the ACT-R literature (transient activation noise $ans = 0.2$, base-level learning decay $bll = 0.5$).

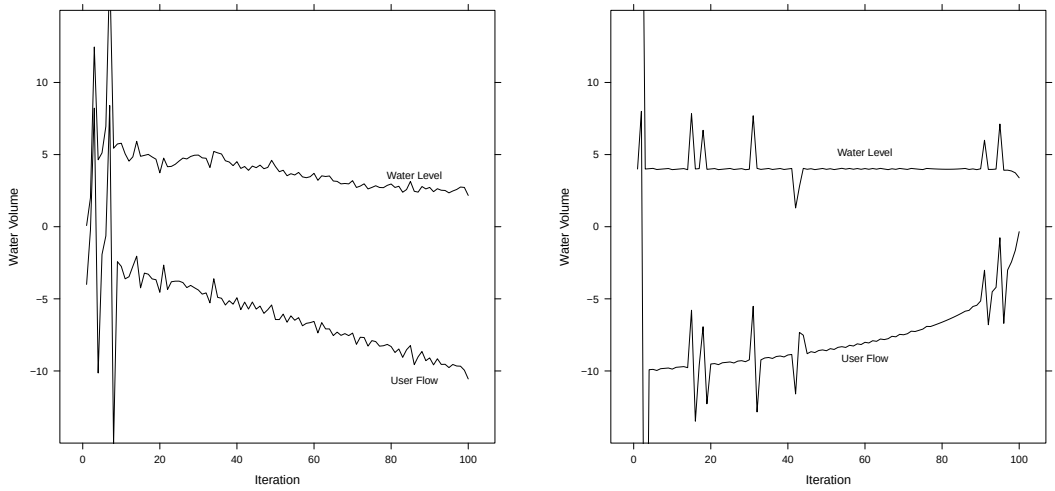
We optimized parameters against two target criteria. The first one was the correlation of water level means for each time step $(1 - 100)$ across all subjects. It resulted in the above parameter settings and yielded a correlation 0.78 for linear increasing, but only 0.24 for nonlinear decreasing condition.

We were also interested in replicating the error as it changes over time. Thus, as second optimization criterion, we estimated the error in water level across blocks of 10 steps for each subject and condition by taking the variance of each 10 data points. The means of these variances across subjects gives an impression of the overall error made during the different phases of each condition. Then, we calculated the correlation between these variance means for the subjects and the model (which has generated predictions for a comparable number of subjects). Optimizing parameters towards this second criterion independently, the best average correlation (0.42) across all four development conditions occurred at parameters very similar to the ones chosen during the first optimization step ($d_1 = 9sec, d_e = 5, t = 0.5$).

Subject performance varies greatly for estimation and calculation strategies; thus we expect different individual a-priori biases for specific strategies (calculate vs. estimate). Thus, the model starts out with no bias for either strategy.

5. Evaluation

The evaluation of the model summarizes the model's performance in the development conditions, but reserves the use of model comparison and descriptive statistics for the novel portion of the data (*transfer conditions*). The model was developed using only the data from four development conditions.



(a) Linear increasing (model)

(b) Nonlinear decreasing (model)

Figure 4: Single-subject model data for water level (upper) and valve setting (lower); reflecting variability within subjects. We comparing the two conditions also shown empirically in Figure 1.

5.1 Performance in the Development Conditions

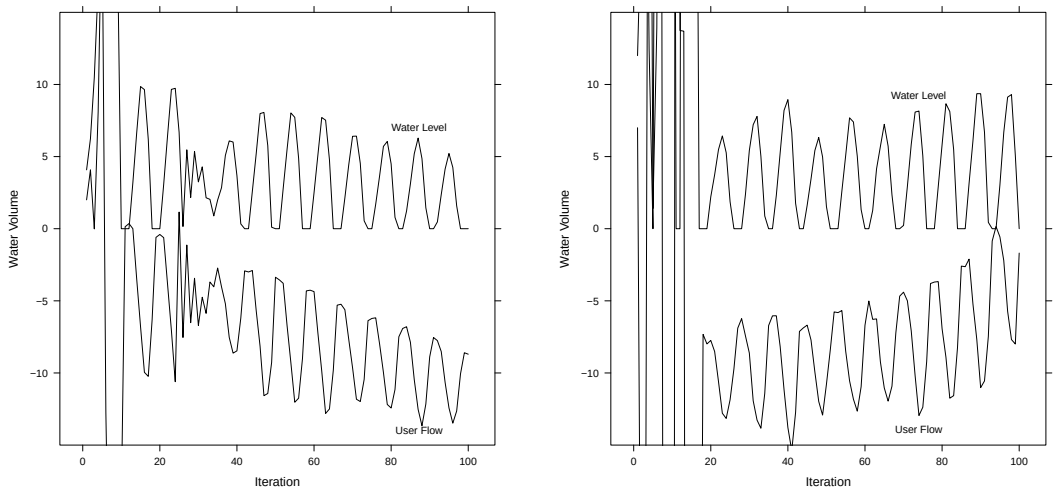
The development conditions comprise all four combinations of linear and non-linear, increasing- and decreasing environmental inflow functions. There was no additional delay in the effect of the chosen valve setting, i.e., the set outflow was in effect in the next iteration.

Because this modeling effort was focused on producing a generalizable model that would fit data from unknown conditions, we consciously avoided fitting the development set data very closely at the cost of manipulating a large number of parameters. Qualitatively, the model explained the following empirical observations (Figures 1(a) vs. 4(a), and 1(b) vs. 4(b)):

- Subjects hold water levels around the target;
- linear functions provoke more accurate responses than do non-linear ones;
- there is an initial phase of high variance; some subjects appear to match the environmental inflow very precisely, but still show the occasional failure.

5.2 Generalization: Performance in the Transfer Conditions

To evaluate how well a model generalizes, we may use held-out portions of a dataset sampled in the same conditions, but perhaps produced by different subjects, or sampled during additional trials. In order to determine whether the model generalizes to different control tasks, however, we require unseen data sampled in novel conditions. For this reason, we present detailed results from the DSF challenge, comparing this model to the other ones entered into the 2009 competition.



(a) 2-step delay, linear increasing, model prediction (b) 2-step delay, non-linear decreasing, model prediction

Figure 5: Single-subject predictions of the model for two conditions: as in Figure 4, but with a 2-step delay between model's valve setting and the valve setting taking effect. (Negative values for valve setting indicate outflow.)

We report the comparative analysis as provided by the DSF challenge organizers. The measures of model fit were standard R^2 measures, comparing the model's and subject's valve settings over all 100 steps, and Root Mean Squared Errors (RMSE). Nine models from different groups were compared in the competition.

The model was evaluated on several transfer conditions:

- Delay of 2 steps. The development conditions included a delay of 1 step, which meant that the subject specified his or her valve setting during the iteration immediately before it is applied. In this condition, the subject's outflow valve setting was inserted into a queue of length 2, so that it did not take effect in the next iteration, but only in the iteration after that one. (For the model's predictions in this case, see Figure 5.) In this condition, the model reached ranks 5 (RMSE) and 3 (R^2), whereas R^2 values for the top six contenders were very similar. During model development, we simulated two-step delay (Figure 5). For this case, the model predicted considerable oscillations (Figures 5(a) and 5(b)). They are relatively regular if only the correct calculation strategy is used (for a lack of noise), but in the final model, these oscillations vary in amplitude.

- Delay of 3: as before, but with a longer queue (see Figure 6 for model predictions and empirical data). In this condition, the model ranked 1 and 2.

- Sequence 2: The inflow function was not smooth, but followed a short sequence of length 2, which could obvious to a human observer. In this condition, the model ranked 2 and 2.

- Sequence 2, noisy: As above, but with additional noise. In this condition, the model ranked 3 and 5.

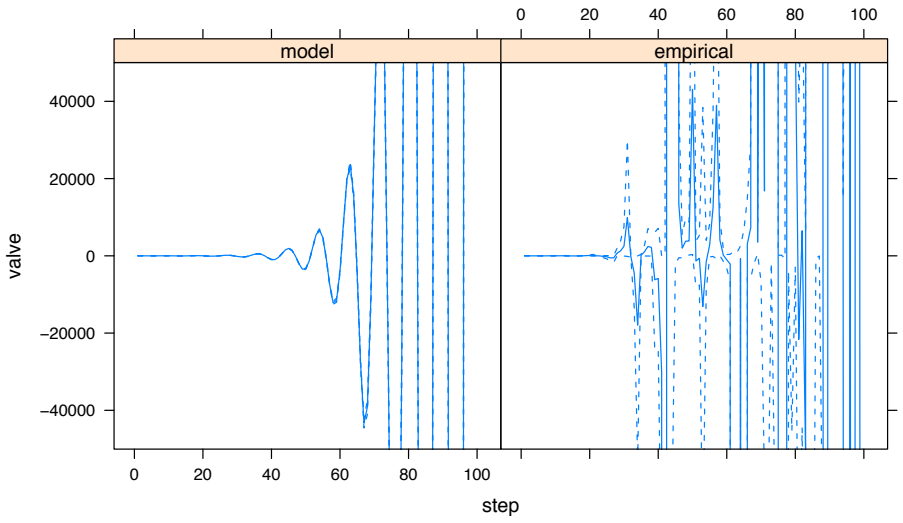


Figure 6: Model predictions and human data in the unseen "Delay 3" condition. Dashed lines mark 95% confidence intervals via bootstrapping.

- Sequence 4. A sequence of inflow amounts as in Sequence 2, but of length 4. In this condition, the model ranked 2 and 1.

In the aggregate ranking provided by the DSF challenge organizers, the model reached rank 1. This ranking is the rank sum, i.e., for each competing model, the sum of its rankings across all transfer conditions. The range of rank sums of all models submitted for both measures (RMSE and R^2) was 26 to 75, mean 50. The first four ranks were 26, 39, and 41.

Aggregating ranks just over the Delay and Sequence conditions, the model also achieved ranks 1, and 1. The model's total ranking was not owed to a particularly good performance in either RMSE or R^2 measure, or (see above) to a single outstanding condition. Average RMSE of the model is *relatively* low ($1.2 * 10^6$ (over each RMSE for each condition) compared to $3.9 * 10^{35}$ for all models in all conditions), average R^2 was *relatively* high (0.0158 model vs. 0.0067 overall). Non-parametric tests failed to establish significant differences between the model entries (Friedman test), nor between the two top-ranked models (paired, one-sided Wilcoxon test; each test for R^2 and RMSE values, across all 5 transfer conditions).

5.3 Discussion

The independently obtained results from the unseen dataset and unknown conditions underscore the model's claim to generalizability. However, the goodness-of-fit measures fail to indicate how well a model captures some of the essential properties of human behavior, such as the frequency and amplitude of the oscillations observed in the delay condition. Overall, all competition models produced relatively poor fits in the transfer condition.

A second caveat is that the dynamic water tank experiment produces longitudinal data, with each step depending on the previous ones. Errors of the model in fitting the data will mount and be penalized by the goodness-of-fit measures used. Inter-subject variability is not evaluated; variability

of error over time (see Section 4.7) was also not explicitly assessed by the DSF challenge. Further work is needed to develop and standardize goodness-of-fit measures.

The model selects its task strategy based on a blend of declaratively stored episodes. Within the same cognitive framework, there are alternatives to this design choice. Halbrügge (2010) describes a model of the DSF task that chooses between alternative estimation and calculation strategies. Halbrügge's model uses ACT-R utility learning to propagate rewards to rules representing those strategies in place of our instance-based learning and blending mechanism. While the models are sufficiently different to not allow for a direct comparison of the two types of learning, the comparison between the two models may serve to motivate new hypotheses. Halbrügge's model ranked third (rank sum 41). It reached similar fits to the empirical data in the Sequence 2 (noisy) and Sequence 4 conditions, but fared worse (particularly by R^2 measure) in the two Delay conditions. On the other hand, Halbrügge's model does well on the development conditions when noise and learning parameters are optimized, as Gluck et al. (2010) point out, and it clearly outperforms ours in the transfer conditions after optimization. We conclude that model performance on unseen data, within the competition, does not compare to a model's performance that was optimized to a specific condition, or developed with knowledge of those conditions.

6. Conclusion

We have presented a model of a control task that predicts data seen in the *Dynamic Stocks and Flows* task. The key to the model's relatively good generalization lies in its metacognitive ability to monitor chosen high-level task strategies and select appropriate ones for the conditions encountered.

The fit to the unseen data is difficult to put into perspective, as most cognitive models are written to fit data from known experimental conditions. It is for this reason that the comparison with competitor models is informative. Still, we caution that the comparison with other models may be weak, given that none of the models achieves very precise fits. The longitudinal nature of the data as well as variability between subjects make it difficult or non-sensical derive data for a hypothetical, *average subject*. It is for this reason that we advocate a range of measures to evaluate the models. An example of such measures were obtained by Gluck et al. (2010). Their results show that our model's different task strategies are necessary: removing either class of strategy worsens performance, either in the development or in the transfer conditions. This demonstrates that the metacognitive layer indeed adapts to the novel conditions.

Perhaps, the particular strategy reflected in aggregated subject data in an experimental design is less important to architectural questions than are the questions of how strategies are conceived of by the subject, and how successful strategies emerge as dominant ones. We optimized the water tank model to reflect performance variability within subjects. However, its predictions for between-subject variance were not evaluated. Such future work will require more knowledge of the distributions of architectural parameters as well as larger datasets reflecting between-subject variability in different control tasks. The metacognitive principle itself applies to many domains. For instance, the metacognitive layer in our model has since been re-used in a different, game-theoretic decision-making context where adaptation to a dynamic environment pays off (Reitter et al., 2010).

This model is designed to contribute to the convergence of modeling approaches: it uses a validated cognitive framework, ACT-R, and existing learning and retrieval mechanisms. It aims to represent cognitive processes, has been evaluated for generalization and is simple, with evaluation

supporting the necessity of its components. At the same time, the model does not specify algorithms which we believe would not contribute to the explanation of the data. The model also underspecifies portions that are best parametrized using existing data.

Our two-level, multiple-strategy model outperformed its competitors in the DSF challenge (the only tasks evaluated there). This performance reflects a true *prediction*: the evaluation was carried out on unseen data, collected under novel experimental conditions that were unknown during model development and parameter fitting. We know that post-hoc design changes and parameter optimization would produce better fits to the data, but contend that only prediction exercises such as the DSF challenge will lead to generalizability in cognitive modeling.

Acknowledgements

The author, while affiliated with some of the DSF challenge organizers, created the model without knowledge of any relevant details beyond those publicly available. The author acknowledges funding from the Air Force Office of Scientific Research (MURI 7, FA95500810356) and would like to thank Dario Salvucci, the editor, and further, anonymous reviewers as well as Sherice Clarke, Jerry Vinokurov and Christian Lebiere and for their comments on an earlier version of this manuscript.

References

Anderson, J. R.; Bothell, D.; Byrne, M. D.; Douglass, S.; Lebiere, C.; and Quin, Y. 2004. An integrated theory of mind. *Psychological Review* 111:1036–1060.

Anderson, J. R. 1991. *The place of cognitive architectures in a rational analysis*. Hillsdale (NJ): Lawrence Erlbaum Associates. 1–24.

Anderson, J. R. 1996. Implicit Memory and Metacognition: Why is the glass half full? In Reder, L. M., ed., *Implicit Memory and Cognition*. Hillsdale (NJ): Lawrence Erlbaum Associates. 123–136.

Ball, J. T.; Gluck, K. A.; Kursmark, M. A.; and Rodgers, S. M. 2003. Comparing three variants of a computational process model of basic aircraft maneuvering. In *Proceedings of the 12th Conference on Behavior Representations in Modeling and Simulations*, 87–98.

Berry, D. C., and Broadbent, D. E. 1987. The combination of explicit and implicit learning processes in task control. *Psychological Research* 49:7–15.

Bott, L., and Heit, E. 2004. Nonmonotonic extrapolation in function learning. *Journal of Experimental Psychology* 30:38–50.

Campbell, J. D. 2008. Addition by Subtraction. *Memory and Cognition* 36:1094–1102.

Fink, E.; Kaplowitz, S.; and Hubbard, S. M. 2002. Oscillation in Beliefs and Decisions. In *The persuasion handbook: developments in theory and practice*. Thousand Oaks, CA: Sage.

Gluck, K.; Halbruegge, M.; Moore, R.; Reitter, D.; and Stanley, C. 2010. Parameter space exploration in cognitive models. *Journal of Artificial General Intelligence* 2(2).

Gonzalez, C., and Lebiere, C. 2005. Instance-based cognitive models of decision making. In Zizzo, D., and Courakis, A., eds., *Transfer of knowledge in economic decision making*. New York: Palgrave McMillan.

Gonzalez, C.; Lerch, F.; and Lebiere, C. 2003. Instance-based learning in dynamic decision making. *Cognitive Science* 27(4):591–635.

Halbrügge, M. 2010. Keep it simple–A case study of model development in the context of the Dynamic Stocks and Flows (DSF) task. *Journal of Artificial General Intelligence* (this issue).

Jones, R. M.; Laird, J. E.; Nielsen, P. E.; Coulter, K. J.; Kenny, P.; and Koss, F. V. 1999. Automated intelligent pilots for combat flight simulation. *AI Magazine* 20:27–42.

Lebiere, C.; Gonzalez, C.; and Warwick, W. 2009. A comparative approach to understanding general intelligence: Predicting cognitive performance in an open-ended dynamic task. In *Proceedings of the Second Conference on Artificial General Intelligence*, 103–107. Amsterdam/Paris: Atlantis Press.

Lebiere, C. 1999. The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft* 8(1):5–19.

Mané, A. M., and Donchin, E. 1989. The Space Fortress game. *Acta Psychologica* 71(1-3):17–22.

McCloskey, M., and Lindemann, M. 1992. MATHNET: preliminary results from a distributed model of arithmetic fact retrieval. In Campbell., ed., *The Nature and Origin of Mathematical Skills*. Amsterdam, Netherlands: Elsevier.

Reder, L. M., and Schunn, C. D. 1996. Metacognition does not imply awareness: Strategy choice is governed by implicit learning and memory. In Reder, L. M., ed., *Implicit Memory and Cognition*. Lawrence Erlbaum Associates. 45–78.

Reitter, D., and Lebiere, C. 2010. Accountable Modeling in ACT-UP, a Scalable, Rapid-Prototyping ACT-R Implementation. In *Proceedings of the 10th International Conference on Cognitive Modeling (ICCM)*, 199–204. Philadelphia, PA.

Reitter, D.; Juvina, I.; Stocco, A.; and Lebiere, C. 2010. Resistance is Futile: Winning Lemonade Market Share through Metacognitive Reasoning in a Three-Agent Cooperative Game. In *Proceedings of the 19th Behavior Representation in Modeling & Simulation (BRIMS)*. Charleston, SC.

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

Taatgen, N., and Lee, F. J. 2003. Production Compilation: A Simple Mechanism to Model Complex Skill Acquisition. *Human Factors* 45(1):61–76.

Vicuso, S. R.; Anderson, J. A.; and Spoehr, K. T. 1989. Representing simple arithmetic in neural networks. In Tiberghien, G., ed., *Advanced Cognitive Science: Theory and Applications*. Chichester, England: Horwood. 144–164.