Architectural process models of decision making: Towards a model database

Cvetomir M. Dimov (cvetomir.dimov@unil.ch), Julian N. Marewski (julian.marewski@unil.ch)
Department of Organizational Behavior, Université de Lausanne,
Quartier UNIL-Dorigny, Bâtiment Internef, 1015 Lausanne, Switzerland

Lael J. Schooler (lschoole@syr.edu)
509 Huntington Hall, Department of Psychology,
Syracuse University, NY 13244, USA

Abstract

We present the project aimed at creating a database of detailed architectural process models of memory-based decision models. Those models are implemented in the cognitive architecture ACT-R. In creating this database, we have identified commonalities and differences of various decision models in the literature. The model database can provide insights into the interrelation among decision models and can be used in future research to address debates on inferences from memory, which are hard to resolve without specifying the processing steps at the level of precision that a cognitive architecture provides.

Keywords: inference from memory; process model; ACT-R; decision making; model database

Introduction

How do we infer which of two cars will be more durable? Which company will be more successful in the coming year? To address such questions, in a typical *two-alternative forced-choice* task of *inference from memory* (Gigerenzer & Goldstein, 1996), two objects (e.g., two companies) are presented on a computer screen. A subject has to infer which of the two objects scores higher on a *criterion* of interest (e.g., the company growth in the next year) by relying on knowledge stored in memory.

Models of inference describe how subjects make inferences by using *attributes* of objects (e.g., who is the company's CEO) as *cues*. Many inferential models have focused on describing not just what the outcome of the inference would be, but also which processing steps a decision maker would take to reach a decision. These models include, among others, the various *fast-and-frugal heuristics* from the *adaptive toolbox of heuristics* (Gigerenzer, Todd, & the ABC Research Group, 1999), *parallel constraint satisfaction* (PCS; Glöckner & Betsch, 2008) and *sequential sampling models* (e.g., Lee & Cummins, 2004).

Such *process models* have increased substantially our understanding of how people make inferences (e.g., Bröder, 2012) and why the inferential process is successful (Gigerenzer & Brighton, 2009), but perhaps more importantly they have raised other questions and fueled important debates: Do people rely on a repertoire of strategies or on a single strategy (e.g., Lee & Cummins, 2004; Marewski, Schooler, & Gigerenzer, 2010; Newell, 2005; Glöckner & Betsch, 2008)? Which types of models (e.g., heuristics vs. more complex models) describe better people's decision processes (e.g., Goldstein & Gigerenzer, 2002; Newell & Bröder, 2008) and under what circumstances? When do people rely on non-compensatory as opposed to compensatory strategies (Glöckner & Bröder, 2011)?

One major barrier to addressing those and related questions is that many models are almost always underspecified compared to the data that they are tested against. Specifically, process models of decision making often remain silent about components of cognition that are the foundation of decision making, such as perception, motor action, or memory. We argue that specifying relevant cognitive-behavioral processes will help those models make more precise predictions about, for example, response time and other process data. The increased precision, in turn, will not only allow researchers to more easily tell potentially competing models apart, but also aid in addressing ongoing debates and open research questions.

In fact, a significant amount of research has already started to embed existing decision models into detailed cognitive theories (Dimov, Marewski, & Schooler, 2013; Fechner, Pachur, Schooler, Mehlhorn, Battal, Volz, & Borst, 2016; Marewski & Mehlhorn, 2011; Marewski & Schooler, 2011; Nellen, 2003; Thomas, Dougherty, Sprenger, & Harbison, 2008; Schooler & Hertwig, 2005). The aim of the current line of work is to expand upon these efforts by systematically implementing existing models of inference in the *cognitive architecture ACT-R* (Anderson, 2007).

In what follows, we will briefly introduce ACT-R and present a summary of the model database that we are in the process of constructing. We will then explain in detail what knowledge each of the decision strategies requires for its functioning. We will conclude by discussing the advantages and shortcomings of our models. Once finalized, we plan to make the database of architectural process models of decision making available to the public.

ACT-R

ACT-R is arguably the most advanced integrated theory of cognition. It has been used to construct models of very diverse tasks and phenomena, which include, among others, associative recognition (Schneider & Anderson, 2012), analogy making (Salvucci & Anderson, 2001) and multitasking (Salvucci & Taatgen, 2008).

Table 1: Outline of the database of architectural process models of decision making, together with summaries of hypothesized procedural and declarative, symbolic and subsymbolic knowledge.

Model	Source	Declarative knowledge	Procedural knowledge	Information at the subsymbolic level
Recognition Heuristic	Goldstein & Gigerenzer (2002)	Alternatives	Try to retrieve chunks representing alternatives. Select alternative corresponding to successfully retrieved chunk.	Activation of chunks of alternatives (proportional to occurrence frequency in environment)
Fluency Heuristic	Schooler & Hertwig (2005)	Alternatives	Retrieve chunks representing alternatives and time retrieval using timing module. Select alternative with faster retrieval time.	
Exemplar Fluency	Juslin & Persson (2002); Nosofksy (1984)	Cue profiles	Retrieve cue profile most similar to alternative's cue profile and time retrieval using timing module. Select alternative with faster retrieval time.	
Exemplar Average		Cue profiles Cue profiles with direct criterion knowledge	Producing an average criterion value through blending over cue profiles similar to alternatives'. Select alternative with larger blended criterion value.	
Exemplar Individual		Cue profiles Cue profiles with direct criterion knowledge	Retrieve cue profile with direct criterion knowledge most similar to alternative's cue profile . Select alternative with higher population of most similar cue profile.	
Set of rules Prototype	Johanson & Kruschke (2005)	Cue profiles	Separate productions firing for each cue-profile-pair difference.	Variable utility of evaluative productions
Prototype Fluency		Cue profiles High criterion value prototype	Retrieve an alternative's cue profile. Retrieve high-criterion-value prototype and time retrieval using timing module. Select alternative, for which high-criterion-value prototype was retrieved more quickly.	
Instance-based learning theory average	Gonzalez, Lerch, & Lebiere (2003); Logan (1988)	Cue profiles Cue profile pairs	Retrieve cue profiles of both alternatives. Produce an average response by blending over choices with similar cue profile pairs.	
Instance-based learning theory individual		Cue profiles Cue profile pairs	Retrieve cue profiles of both alternatives. Retrieve cue profile pair most similar to cue profile pair of current alternatives.	
Parallel constraint satisfaction	Glöckner & Betsch (2008)	Cue profiles Cue profile pairs Cue profile pair prototypes	Retrieve cue profiles of both alternatives. Retrieve cue profile pair prototype most similar to cue profile pair of current alternatives.	
Take-the-best reinforcement	Gigerenzer & Goldstein (1996)	Cues Cue values	Determine which cue to consider by firing production with highest utility. Decide as soon as cue values differ.	Different production utility for each cue
Take-the-best declarative		Cues Cue values Cue validity pair	Retrieve next most valid cue. Decide as soon as cue values differ.	
Tallying		Cues Cue values	Retrieve cue with highest activation. Stop retrieval upon retrieval failure. Count positive cue values.	
Unit-weight linear model		Cues Cue values	Retrieve cue with highest activation. Stop retrieval upon retrieval failure. Count positive and subtract negative cue values.	
Weighted additive		Cues Cue values Cue validities	Retrieve cue with highest activation. Stop retrieval upon retrieval failure. Compute weighted sum of positive cue values.	
Weighted linear model		Cues Cue values Cue validities	Retrieve cue with highest activation. Stop retrieval upon retrieval failure. Weighted sum of positive and negative cue values.	
Take-the-first-cue	Marewski & Schooler (2011)	Cues Cue values	Retrieve cue with highest activation. Decide as soon as cue values differ.	Activation of chunks of cues proportional to occurrence frequency in environment
Minimalist	Gigerenzer & Goldstein (1999)	Cues Cue values	Retrieve cue with highest activation. Decide as soon as cue values differ.	Activation of chunks of cues equal
Take-the-last		Cues Cue values	Retrieve cue with highest activation. Decide as soon as cue values differ.	
Sequential sampling model	Lee & Cummins (2004)	Cue values	Retrieve cue with highest activation. Count positive cue values. Stop retrieval upon reaching threshold.	
Weighted sequential sampling model		Cue values Cue validities	Retrieve cue with highest activation. Weighted sum of positive cue values. Stop retrieval upon reaching threshold.	

ACT-R describes cognition as a set of modules that communicate through a *procedural module* realized as a *central production system*. The production system consists of production rules (i.e., if–then rules) whose conditions (the "if"-parts) are matched against the modules. If a rule's conditions are met, then the rule can fire and the specified action can be carried out. Modules model different cognitive processes, such as vision (*visual module*), motor action (*motor module*), declarative memory (*declarative module*), short-term information storage (*imaginal module*) and time tracking (*timing module*; Taatgen, van Rijn, & Anderson, 2007). Productions send commands to modules to perform an action or change their state, or to access content placed in modules' *buffers*. In fact, because productions can only access content placed in the buffers, these can be thought of as processing bottlenecks. For instance, a production rule cannot access all information stored in the declarative module, but only the information placed in its associated *retrieval buffer*.

Productions are the representation of choice for procedural knowledge, while declarative knowledge, such as factual and episodic knowledge, is represented as *chunks*. Perceptual and memory modules, respectively, perceive and retrieve information in the form of chunks. A chunk consists of a set of *slots*, where each slot is (a pointer to) another chunk. For example, a chunk containing information about a company's annual revenue will have a slot with the company's name and another slot with its revenue.

ACT-R distinguishes a *symbolic* and a *subsymbolic system*. Productions, modules and buffers constitute the symbolic system, whose dynamics are governed by a set of equations, describing ACT-R's subsymbolic system. At the subsymbolic level, chunks' *activations* determine, for example, retrieval time or recall probability; productions' *utilities* reflect which productions were more successful in the past and therefore more likely to fire; visual parameters determine the time needed to shift visual attention to an object in the visual field, while motor parameters determine the time to generate a motor response.

Each ACT-R model is essentially composed of specifications of how declarative and procedural knowledge interact, both at the symbolic and subsymbolic levels. We will now focus on describing the declarative and procedural knowledge used in defining the models in the database. We refer those interested in a detailed exposition of ACT-R to Anderson (2007).

Model building blocks

The models of inference that we will consider are presented in Table 1. In implementing these models in ACT-R, we relied on the building blocks that this cognitive architecture provides.

Perceptual and motor processes

All models have equivalent perceptual and motor processes, involving visual perception from a screen and manual action on a keyboard. The models first perceive each of the alternatives presented on a computer screen and, after executing a sequence of cognitive steps, they make a response by pressing the appropriate key on a keyboard. The primary contribution to behavioral predictions of the perceptual and motor processes in our models is to add a realistic estimate of perceptual-motor latency.

Declarative chunks

The factual knowledge (e.g., "Berlin is a capital") that a model relies upon to make a decision is stored in declarative memory. Ten types of chunks are needed to construct the models in the database. Table 2 provides a summary of those chunk types and examples in Lisp code for each. Note that the examples are given for the *city-size task*, in which cities act as alternatives and subjects need to infer which of two cities is larger.

The simplest chunk type contains just the name of the alternatives. For example, if the alternatives are cities, whose relative sizes need to be inferred, such a chunk contains the city name (e.g., "Berlin"). These chunks are all that is required for inferential models, which rely on accessibility information, such as the recognition and fluency heuristics.

The second chunk type contains an entire *cue profile* of an alternative (i.e., the set of cues associated with an alternative). Such chunks are used, among others, by exemplar and prototype models. Some exemplar models also require chunks with *direct criterion knowledge* in addition to the cue profile. Moreover, prototype models require not only cue profiles, but also a stored prototype of an object with a high criterion value.

Table 2: Declarative knowledge categories.

Chunk type label	Chunk examples in Lisp code
Alternative	(**berlin** *name* Berlin)
Cue profile	(**berlin** *name* Berlin *airport* yes *capital* yes ...)
Cue profile with direct criterion knowledge	(**berlin** *name* Berlin *population* 4000000 *airport* yes ...)
High criterion value prototype	(**big-city** *name* prototype *airport* yes *capital* yes ...)
Cue profile pair	(**pair1** *airport1* yes *airport2* no *capital1* yes *capital2* no …)
Cue profile pair prototype	(**prototype-left** *airport1* yes *airport2* no *capital1* yes *capital2* no …)
Cue	(**cue1** *type* airport)
Cue value	(**berlin-airport** *city* Berlin *cue* airport *value* yes)
Cue validity	(**airport-validity** *cue* airport *validity* 90)
Cue validity pair	(**cue-pair** *first* airport *second* capital)

Note. In these examples, *chunk names*, used for convenience, are presented in **bold**; *slot names*, indicating a specific attribute, are in *italics*, while *slot values*, representing the attribute values, are in normal font.

Resembling exemplar and prototype models, instance-based learning theory and parallel constraint satisfaction consider cue configurations to make inferences. However, they differ from the former in that they require chunks, which contain *pairs of cue profiles*. For example, the model "Instance-based learning theory individual" retrieves the cue profiles of both alternatives and then retrieves a cue profile pair from a successful previous trial. It then makes an inference based on the decision outcome of the retrieved cue profile pair. Similarly, our implementation of the parallel constraint satisfaction model requires a prototype of a successful cue profile pair.

Unlike *configural models*, like exemplar models, *cue-abstraction models* (Newell & Bröder, 2008) operate on individual cues. Such models, like take-the-last, retrieve cues one by one. Take-the-last requires separate chunk types for a cue and for the values of the alternatives on that cue. In addition to these chunks, other models, like take-the-best, require information about *cue validities* (i.e., the probability of making a correct inference using only this cue if the cue discriminates; see, Gigerenzer, Hoffrage, & Kleinbölting, 1991), which, if taught in the experiment (e.g., Bröder, & Schiffer, 2003), are stored numerical values. Finally, in some experiments one is provided only with the validity hierarchy, which can be represented as validity pairs of subsequent cues.

Procedural knowledge: The sequence of processing steps

The procedural knowledge of a model consists of a fine-grained sequence of processing steps (i.e., productions) that the model relies upon to make a decision. In all models, the sequence of processing steps includes commands to the visual module to encode the information presented on the screen and to the motor module to press a key to respond in a computerized experiment. As for the rest, the exact sequence of processing steps follows the original model definitions.

For example, fast-and-frugal heuristics usually rely on separate cues, on which detailed search, stopping and decision rules operate. Those models often theorize about the order, in which cues are considered. This ordering can be modeled through productions. In addition, productions can also determine if the model weighs cues equally, as in tallying, or differently, as in the weighted additive model, and execute this process. If cues are weighted equally, productions are required to send a request to declarative memory to retrieve the cue values. Productions then increment by 1 the number, which tracks the count of cues with a positive cue value of the alternative of interest. Other models, such as exemplar models, rely on all available cue information stored in a single chunk to make a decision. In such models, procedural knowledge is more peripheral to the decision process and mostly focuses on retrieval attempts.

Productions not only initiate retrieval, but are also dependent on what is retrieved, because a key determinant of which productions can fire is the available declarative knowledge. Specifically, at each point in time only those productions, whose condition match the buffer states, will be considered to fire. Ultimately, which chunks are retrieved from memory will determine what could be placed in the buffers and therefore which productions will match.

Information at the subsymbolic level

At the subsymbolic level, there is continuously valued information, which is necessary for the execution of some inferential strategies. However, productions cannot directly read out subsymbolic values. Instead, the model needs to let subsymbolic values guide symbolic knowledge. Thus far, we have identified four ways in which subsymbolic values play a key role in the execution of strategies.

First, the activation of chunks representing alternatives contains information about the alternatives' occurrence frequency in the environment. Specifically, *base-level activation* is a function of prior history of a chunk, which partially depends on environmental occurrence frequency, which, in turn, is related to many criteria of interest (Hertwig, Herzog, Schooler, & Reimer, 2008). Accessibility-based strategies, such as the fluency heuristic, track the retrieval speed of alternatives as determined by their activation and choose the alternative, which was retrieved noticeably faster.

Second, activation can order cues, because cues which have a higher occurrence in the environment likely will have a higher activation. Thus, these cues may be more likely to be considered first in lexicographic strategies, such as take-the-first-cue or a sequential sampling model.

A third way in which information at the subsymbolic level can be used is as an implicit cue weighting mechanism. This weighting can take place through *spreading activation*, which is determined by the degree of association between the chunks placed in buffers and the chunks in declarative memory. If the cue profile of one of the alternatives is currently placed in the imaginal buffer, then it will activate cue profiles in memory through spreading activation. Those cue profiles will then have precedence in retrieval. Exemplar models rely on this process to make an inference about the alternative's criterion value.

Finally, production utility contains information about prior success. Production utility determines which production is more likely to fire when two or more productions are competing. If such a competition takes place between productions, which select which cue will be considered next, the utility of these productions can act as a cue's importance (e.g., as its validity, see Gigerenzer, Hoffrage, & Goldstein, 2008, for the hypothesis that such a reinforcement learning process can teach cue validities) in lexicographic cue-abstraction models. This is the mechanism used in the model "Take-the-best reinforcement", which encodes the selection of each cue with a separate production and then learns the success of those cues through trial and error.

Discussion and conclusion

We aim to provide a database of ACT-R implementations of decision models used in the literature on inferences from memory. We have divided these models into their key components. The models can serve as a basis for model tests and further model developments. Specifically, this database can be used, first, in model comparison simulations on the outcome and process level, whereby one identifies regions in the parameter space where these models diverge. Second, this database can be used in future studies to identify decision processes using both behavioral and neural data. This is an important advantage of ACT-R, because any ACT-R model can generate fMRI predictions on top of behavioral process predictions, such as response time, because of the established module-to-brain mappings (for an introduction, see Borst & Anderson, 2015).

In addition, we think that the systematic examination of the building blocks of existing decision models will help us gain insights into how the models are related to each other. For example, through these implementations, we see that the parallel constraint satisfaction model can be conceived as functionally similar to an instance-based learning model, which stores and retrieves prototypical cue profile pairs.

It is important to note that in creating our ACT-R models we were forced to work with the mechanisms that ACT-R provides. For example, the original parallel constraint satisfaction model is cast as a connectionist network, in which connection weights are iteratively updated after each decision. This leads to cues effectively changing their validities as trials progress. As currently conceived, our model does not reproduce this behavior. Nevertheless, the model "Instance-based learning theory average", which in our database is very similar, effectively provides such a mechanism and can be thought as functionally analogous to the original parallel constraint satisfaction.

Such redefinitions and novel distinctions introduced in our modeling endeavor were due to the partial overlap between the various decision models in the literature. Another such distinction that we decided to introduce was in the declarative representation, which cue-abstraction models, like take-the-best and the sequential sampling model introduced by Lee and Cummins (2004), rely on. Originally, both models were conceived as, first, considering a cue, and only then examining the values of that cue for both alternatives. We have kept this definition for take-the-best and other heuristics. However, we have decided to label those models, which retrieve cue values directly, in a manner purely determined by declarative principles, sequential sampling models. These models can, for example, consider the value of cue 2 for alternative A, followed by the value of cue 4 for alternative B, and so on.

Another remark concerns the high degree of detail, which ACT-R introduces when decision models are implemented in it. The fine-grained way in which ACT-R models are specified has forced us, in many cases, to make assumptions about processes, about which the original models remained silent. For example, we had to rely on assumptions about how cues are ordered in take-the-best. We have considered two ways to order cues in this work. Our first implementation relies on declarative retrieval to order cues, while the second one relies on procedural knowledge and utility learning. These assumptions reflect, so we hope, realistic ways of learning. On the one hand, in many experiments on take-the-best, one is explicitly taught the cue hierarchy, which is then stored as declarative knowledge. On the other hand, in natural settings, ordering cues according to validity is likely to occur through reinforcement learning, whereby one has had significant experience with considering several cues in the same setting.

To conclude, we would like to stress that Table 1 does, naturally, not include all possible tweaks and modifications that one can introduce when constructing models in ACT-R. It will be left to input from the different researchers working on inference from memory to determine which of our current ideas will survive, and which ones will be replaced or extended by others.

Acknowledgments

We thank the Swiss National Science Foundation for generously funding this research with Project 146702 "Architectural Process Models of Decision Making".

References

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

Borst, J. P., & Anderson, J. R. (2015). Using the ACT-R Cognitive Architecture in combination with fMRI data. In *An introduction to model-based cognitive neuroscience* (pp. 339-352). Springer New York.

Bröder, A., & Schiffer, S. (2003). Take the best versus simultaneous feature matching: Probabilistic inferences from memory and effects of representation format. *Journal of Experimental Psychology: General*, *132*, 277–293.

Bröder, A. (2012). The quest for take the best - Insights and outlooks from experimental research. In P. Todd, G. Gigerenzer, & the ABC Research Group, *Ecological rationality: Intelligence in the world* (pp. 216-240), New York: Oxford University Press.

Dimov, C. M., Marewski, J. N., & Schooler, L. J. (2013). Constraining ACT-R models of decision strategies: An experimental paradigm. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (pp. 2201–2206). Austin, TX: Cognitive Science Society.

Fechner, H. B., Pachur, T., Schooler, L. J., Mehlhorn, K., Battal, C., Volz, K. G., & Borst, J. P. (2016). Strategies for memory-based decision making: Modeling behavioral and neural signatures within a cognitive architecture. *Cognition*, *157*, 77-99.

Gigerenzer, G., & Brighton, H. (2009). Homo heuristicus: Why biased minds make better inferences. *Topics in Cognitive Science, 1*, 107-143.

Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review, 104*, 650–669.

Gigerenzer, G., & Goldstein, D. G. (1999). Betting on one good reason: The take the best heuristic. In G. Gigerenzer, P. M. Todd, & the ABC Research Group, *Simple heuristics that make us smart* (pp. 75-95). New York: Oxford University Press.

Gigerenzer, G., Hoffrage, U., & Goldstein, D. G. (2008). Fast and frugal heuristics are plausible models of cognition: Reply to Dougherty, Franco-Watkins, & Thomas (2008). *Psychological Review, 115*, 230–239.

Gigerenzer, G., Hoffrage, U., & Kleinbölting, H. (1991). Probabilistic mental models: A Brunswikian theory of confidence. *Psychological Review, 98*, 506-528.

Gigerenzer, G., Todd, P. M., & the ABC Research Group. (1999). *Simple heuristics that make us smart.* New York: Oxford University Press.

Glöckner A., & Betsch T. (2008a). Modeling option and strategy choices with connectionist networks: Towards an integrative model of automatic and deliberate decision making. *Judgment and Decision Making, 3*, 215-228.

Glöckner, A., & Bröder, A. (2011). Processing of recognition information and additional cues: based analysis of choice, confidence, and response time. *Judgment and Decision Making*, 6, 23-42.

Goldstein, D. G., & Gigerenzer, G. (2002). Models of ecological rationality: The recognition heuristic. *Psychological Review, 109*, 75–90.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science, 27*, 591-635.

Hertwig, R., Herzog, S. M., Schooler, L. J., & Reimer, T. (2008). Fluency heuristic: a model of how the mind exploits a by-product of information retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 34*, 1191-1206.

Johansen, M. K., & Kruschke, J. K. (2005). Category representation for classification and feature inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31*, 1433-1458.

Juslin, P., & Persson, M. (2002). PROBabilities from EXemplars (PROBEX): A “lazy” algorithm for probabilistic inference from generic knowledge. *Cognitive Science, 26*, 563-607.

Lee, M. D., & Cummins, T. D. R. (2004). Evidence accumulation in decision making: Unifying the ‘take the best’ and the ‘rational’ models. *Psychonomic Bulletin & Review, 11*, 343–352.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review, 95*, 492-527.

Marewski J. N. & Mehlhorn K. (2011). Using the ACT-R architecture to specify 39 quantitative process models of decision making. *Judgment and Decision Making, 6*, 439–519.

Marewski, J. N., & Schooler, L. J. (2011). Cognitive Niches: An ecological model of strategy selection. *Psychological Review*, *118*, 393-437.

Marewski, J. N., Schooler, L. J., & Gigerenzer, G. (2010). Five principles for studying people’s use of heuristics. *Acta Psychologica Sinica, 42*, 72-87.

Nellen, S. (2003). The use of the “take-the-best” heuristic under different conditions, modelled with ACT-R. In F. Detje, D. Dörner, & H. Schaub (Eds.), *Proceedings of the fifth international conference on cognitive modelling* (pp. 171–176). Bamberg, Germany: Universitätsverlag Bamberg.

Newell, B. R. (2005). Re-visions of rationality? *Trends in Cognitive Sciences, 9*, 11–15.

Newell, B. R., & Bröder, A. (2008). Cognitive processes, models and metaphors in decision research. *Judgment and Decision Making, 3*, 195–204.

Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10*, 104-114.

Salvucci, D. D., & Anderson, J. R. (2001). Integrating analogical mapping and general problem solving: The path-mapping theory. *Cognitive Science, 25*, 67-110.

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

Schooler, L. J., & Hertwig, R. (2005). How forgetting aids heuristic inference. *Psychological Review, 112*, 610–628.

Schneider, D. W., & Anderson, J. R. (2012). Modeling fan effects on the time course of associative recognition. *Cognitive Psychology, 64*, 127-160.

Taatgen, N. A., van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: the role of cognition, attention, and learning. *Psychological Review, 114*, 577-598.

Thomas, R. P., Dougherty, M. R., Sprenger, A. M., & Harbison, J. (2008). Diagnostic hypothesis generation and human judgment. *Psychological Review, 115*, 155-185.

Tversky, A. (1972). Elimination by aspects: A theory of choice. *Psychological Review, 79*, 281–299.