Learning from Errors by Counterfactual Reasoning in a Unified Cognitive Architecture

Andreea Danielescu (lavinia.danielescu@asu.edu)
David J. Stracuzzi (david.stracuzzi@gmail.com)
Nan Li (nanan9177@gmail.com)
Pat Langley (langley@asu.edu)
Computing Science and Engineering
Arizona State University, Tempe, AZ 85287 USA

Abstract

A key characteristic of human cognition is the ability to learn from undesirable outcomes. This paper presents a computational account of learning from errors based on counterfactual reasoning, which we embed in Icarus, a unified theory of the cognitive architecture. Our approach acquires new skills from single experiences that improve upon and mask those that initially produced the undesirable behavior. We demonstrate the operation of this model in a simulated urban driving environment. We also relate our approach to other research on error-driven learning and discuss possible improvements to the framework.

Keywords: cognitive architecture, learning from error, counterfactual reasoning, problem solving

Background and Motivation

The ability to acquire knowledge from experience is a fundamental component of human intelligence. There exist many accounts of learning from positive experiences, most often based on successful problem-solving attempts (Anzai & Simon, 1979; Laird, Rosenbloom, & Newell, 1986). In this paper, we focus instead on learning from undesirable outcomes, an ability that plays an important role in human cognition by providing a mechanism for avoiding past failures in the future. We provide a computational model for one type of error-driven learning that uses counterfactual reasoning to determine both the error's cause and the correct behavior.

Counterfactual reasoning is a strategy that considers what might have occurred if causal events were changed in some way. Psychological studies suggest that people employ counterfactual reasoning in a variety of situations (Roese, Hur, & Pennington, 1999). Byrne and McEleney (2000) also show that they tend to employ counterfactual reasoning mainly in response to negative outcomes, such as failure to achieve or maintain goals. Finally, Epstude and Roese (2008) make the connection to learning, based on their theory that the primary motivation for counterfactual reasoning is to improve future performance.

The work described here offers a computational account of the role of counterfactual reasoning in learning from failures. We embed this account within Icarus (Langley & Choi, 2006), a unified theory of the human cognitive architecture that makes a commitment to hierarchical, composable knowledge structures. We claim that these structures, along with the mechanisms for using and acquiring them, provide Icarus with basic support for benefiting from undesirable outcomes. Our approach to learning from errors responds to a single negative experience, which distinguishes it from connectionist, reinforcement-based, and Bayesian techniques, which typically require many experiences.

We begin our discussion with a motivating task domain and a review of the Icarus architecture. After this, we present our approach to learning from errors via counterfactual reasoning, including methods for determining the source of the error, acquiring new concepts and skills in response, and utilizing these structures in future behavior. We then describe the extended architecture's operation in the task domain, discuss connections to other work on error-driven learning, and consider directions for further research in this important area.

An Illustrative Domain: Urban Driving

In modern society, the task of operating a vehicle in an urban setting is both common and cognitively challenging. People perform a variety of tasks in this context, such as navigation, obstacle avoidance, and signal response, along with higher-level tasks such as package delivery. Successful performance relies on substantial domain expertise, making urban driving a useful domain in which to study embedded cognition and learning.

For this reason, we have developed a three-dimensional urban driving environment based on the Torque Game Engine produced by Garage Games.[1] The driving simulator provides the driver with control over the gas pedal, brake and steering, with objects obeying realistic laws of physics. The simulator also generates detailed perceptual information, in egocentric polar coordinates, about nearby entities, including road segments, intersections, lane lines, buildings, pedestrians, and other vehicles.

The driving task we examine here requires the agent to overtake a stalled vehicle, which it decides to passs on the left. However, in taking this step, the agent crosses a double yellow line, thereby violating the rules of driving and risking collision with oncoming traffic. The problem is not that the agent lacks knowledge about this constraint; the error occurs because it was focusing on a different goal that interacts with the one it violates. We

[1] http://www.garagegames.com

maintain that learning to avoid such errors relies on a form of counterfactual reasoning, which we describe in detail later. Our analysis is not limited to urban driving, nor do we intend it to model how humans learn to operate a vehicle, but it provides a useful setting to illustrate our ideas. However, we must embed our account within some theoretical framework, to which we now turn.

A Review of the Icarus Architecture

We have explored these issues within Icarus, a theory of the cognitive architecture (Newell, 1990) that makes commitments to the representations, performance mechanisms, and learning processes that underlie intelligence. Like Soar (Laird et al., 1986) and ACT-R (Anderson, 1993), Icarus encodes content as symbolic list structures, matches long-term structures against short-term ones in a recognize-act cycle, combines goal-driven with data-driven processing, and interleaves an incremental form of learning with performance. Key differences include separate memories for concepts and skills, the hierarchical organization of knowledge, and cascaded integration in which problem solving builds on skill execution, which in turn relies on conceptual inference. In this section, we review Icarus' structures and processes, drawing brief examples from urban driving. Langley and Choi (2006) describe the framework in more detail.

The most basic process in Icarus is inference, which matches the agent's perceptions against long-term conceptual structures to produce beliefs about the environment. On each cycle, the environment deposits descriptions of perceived objects into the *perceptual buffer*, with each percept giving the object type, a unique identifier, and (typically numeric) attribute-value pairs. Icarus links these perceptions to long-term structures in its *conceptual memory.* Each concept describes a class of situations in logical form that includes the concept's name, its arguments, and the conditions under which the concept applies. Conditions may refer to percepts or to simpler concepts, thus creating a hierarchical organization on memory. For instance, the conceptual structure

```
((aligned-in-lane ?agent ?line1 ?line2)
 :percepts   ((self ?agent) (lane-line ?line1))
              (lane-line ?line2 angle ?angle)
 :relations  ((steering-straight ?agent)
              (in-lane ?agent ?line1 ?line2))
 :tests      ((≥ ?angle -3) (≤ ?angle 3)))
```

states when one should infer that `?agent` is driving parallel to the lane lines on either side, with the conditions referring to percepts, numeric tests among perceived attributes, and other conceptual relations.

On each cognitive cycle, Icarus matches these concepts against elements in its perceptual buffer to produce inferences that it deposits in a *belief memory*, which in turn produce higher-level inferences. These typically describe relations among objects in the environment, with each belief being an instance of some defined concept. For example, the belief `(aligned-in-lane me line23 line24)`, that the agent is aligned in a lane bounded by `line23` and `line24`, is an instance of the `aligned-in-lane` concept above. A recent extension of Icarus includes time stamps on beliefs to indicate when the architecture inferred them and when they became false, with the symbol `now` indicating that a belief holds on the current cycle. These time-annotated beliefs serve as a simple episodic trace to which we will return later.

Icarus includes a separate long-term memory for skills. These are similar in form to concepts but specify methods for achieving goals rather than conditions for recognizing their achievement. Each skill includes a generalized goal (which must refer to a defined concept), a set of perceptual and conceptual conditions that must hold for the skill to match, and ordered subgoals that, once satisfied, should achieve the parent goal. For instance, the skill clause

```
((driving-well-in-lane ?agent ?line1 ?line2)
 :percepts   ((self ?agent) (lane-line ?line1)
              (lane-line ?line2))
 :start      ((in-lane ?agent ?line1 ?line2))
 :subgoals   ((at-speed ?agent)
              (centered-in-lane ?agent ?line1 ?line2)
              (aligned-in-lane ?agent ?line1 ?line2)))
```

specifies three subgoals the agent should achieve, once it is in a lane, to be driving well in that lane. Primitive skills have the same structure but replace subgoals with a set of executable actions the agent should carry out.

The Icarus execution process uses the beliefs produced during conceptual inference to determine which skills to select. This process begins by choosing an unsatisfied goal – a concept instance the agent wants to be true – stored in a separate *goal memory.* The architecture retrieves all skills indexed by this goal, then attempts to find an applicable path downward through the skill hierarchy. A skill path is applicable if, for each skill on the path, the associated goal is not satisfied and the associated conditions are met. Such a path must terminate in a primitive skill with executable actions that affect the environment. On each cycle, Icarus selects the first such path through the skill hierarchy that it finds, incorporating a preference for continuing an activity it has initiated over starting new ones.

When Icarus can find no applicable skill in memory to achieve its current goal, it resorts to a form of means-ends problem solving (Newell & Simon, 1961; Carbonell, Knoblock, & Minton, 1990) that attempts to dynamically compose known skills, which it executes as they become applicable. The process begins when the architecture cannot find an applicable path through the skill hierarchy, in which case it attempts to retrieve a skill that would achieve the goal, then creates subgoals for its unsatisfied preconditions. If Icarus cannot find such a skill, it instead retrieves the definition for the goal concept and creates subgoals for each of its unsatisfied conditions. This process continues recursively, chaining backward off subgoals, until it retrieves a relevant applicable skill, which it then executes in the environment. Upon achieving the current subgoal, the system turns its attention to others, continuing this activity until achieving the top-level goal that initiated problem solving.

This mechanism lets Icarus overcome situations in which it lacks skills to achieve its goals, but it often requires substantial search. However, the architecture also includes a learning process that generalizes and stores solutions for future use in similar situations. Briefly, this creates a skill whenever problem solving achieves a goal or subgoal, with the new structure being indexed by that goal, including subgoals that it satisfied along the way, and having conditions that were present when it began working on the problem. The details differ depending on whether the problem solver chained off skills or concept definitions, but the results are similar for both cases. Icarus can then apply the new skills in the same manner as the other, older skills during subsequent execution.

Learning from Undesirable Outcomes

We have used Icarus to develop cognitive models in a number of complex domains, including urban driving (Langley & Choi, 2006) and American football (Li, Stracuzzi, Cleveland, et al., 2009). Nevertheless, the architecture lacks some important functionalities, including the ability to learn from undesirable outcomes. The work we report here has started to address this limitation by adding a mechanism for learning from errors. More specifically, we consider the case in which the agent manages to achieve a given goal, but in which this causes it to inadvertently violate another, higher-priority, goal, which it must then attempt to repair.

As we discuss in detail below, the extended Icarus responds to such situations in three steps. First, it determines which goals conflicted and constructs a new concept that it uses to encode the combined goal. Next, the architecture employs counterfactual reasoning to identify the primitive skill that produced the error and to determine another sequence that achieves this goal. Note that the skill which led to the conflict may have been executed many cycles prior to the actual goal violation. Finally, Icarus learns new skills that mask the original structures and achieve the joint goal in similar situations. These mechanisms are not completely new, in that they build upon many existing Icarus processes, making them more elaborations of the architectural framework than separate modules.

Combining Interacting Goals

The first step toward learning from errors is to determine which goals conflicted to cause the failure. A conflict here refers to a case in which the system violates a previously satisfied, higher-priority goal in the course of pursuing its current goal. For example, suppose a driving agent (`me`) has two goals, `(on-right-side me)` and `(at-speed me)`. The agent first maneuvers the vehicle onto the right side of the road and then begins accelerating toward its desired speed. If another, slower-moving vehicle then enters the road in front of the agent, it may execute a skill for passing on the left. This violates the agent's first goal, and causes it to abandon the second goal in an effort to restore the first.

If the agent prefers to pass on the left when approaching a slow vehicle, it will never satisfy both goals in this situation. To address this stalemate, the architecture constructs a new goal by conjoining the two concepts that supported the original goals, then replacing the goals with one based on the new concept. Returning to the driving example above, Icarus first creates a concept `(on-right-side-and-at-speed ?agent)` with relations `(on-right-side ?agent)` and `(at-speed ?agent)`. Note that the new concept extends the current hierarchy by building on existing concepts. Icarus then replaces the two original goals with `(on-right-side-and-at-speed me)`, giving the agent a new goal for which it can learn more specific skills.

Assigning Blame and Finding Alternatives

After identifying which goals conflicted and creating a revised top-level goal, Icarus attempts to understand the reasons for its failure and how it might have been avoided. To this end, it attempts to identify the most recently executed primitive skill that, if replaced by a better choice, would avoid violating the high-priority goal and achieve the newly constructed one. This constitutes a form of counterfactual reasoning. The system begins by considering each primitive skill executed in the episode in reverse chronological order. For a selected skill, it rolls back episodic belief memory to the cycle on which it first selected and executed the skill.[2] Icarus does not consider non-primitive skills, as they could lead it to backtrack farther than necessary.

After rolling back the episodic trace, Icarus invokes problem solving with the new, conjoined goal created earlier. Recall that the architecture normally interleaves problem solving with execution, but here the agent can only suppose what might have happened if it had taken another path. For this reason, we introduced a more traditional version of problem solving that uses mental simulation based on skills' effects. Each time the problem solver selects a skill for imaginary execution, it updates belief memory with the expected changes, then triggers inference, which updates belief memory as though it had received new percepts.

[2] Primitive skills may be durative, which means they may require several cycles to achieve their intended goals. Backtracking over such a skill may therefore involve jumping backward in time by several cognitive cycles.

If the problem solver fails to achieve the combined goal, then the system rolls back another step, past the preceding primitive skill in the original episodic trace. The process outlined above then begins again with a new round of problem solving. Note that this approach to error localization and counterfactual reasoning is well integrated with existing facets of the architecture, and it depends critically on results produced by the modules for inference and problem solving. This provides further evidence that Icarus offers a unified theory of cognition.

Learning and Selecting New Skills

Having determined a sequence of primitive skills that would have achieved both of its original goals, the extended architecture must generalize these results and store them in skill memory for use in guiding future behavior. We want the acquired skills to apply in more constrained situations than those which produced the undesirable outcome, so they should mask the old skills to prevent their selection in these cases. For our driving example, the system should select the new skill only if it wants to achieve *both* `(on-right-side ?agent)` and `(at-speed ?agent)`.

To learn skills from the results of counterfactual reasoning, Icarus uses the same mechanism as during normal problem solving. As the agent works toward its top-level goal, it acquires a new skill as means-ends analysis achieves each subgoal. The use of counterfactual reasoning and mental simulation, rather than execution in the environment, has no effect on this process. However, to make effective use of this learned knowledge, we must modify Icarus' execution module. The standard mechanism selects the unsatisfied goal with highest priority, then finds a path through the skill hierarchy that should achieve it. This scheme works well when there are no goal interactions, but, as we have seen, it can lead to problems when they exist.

In response, we modified the execution module to prefer skills that, other things being equal, would let the agent achieve multiple goals. For example, it selects a skill that addresses two goals with first and second priority over one that would achieve only the first goal. However, priority still plays a key role; the system prefers a skill that tackles a first priority goal over one that would achieve goals with second and third priority. This approach lets more specific skills acquired through counterfactual reasoning mask the original, more general skills that caused the undesired behavior, while letting the older skills remain available for situations in which only they apply.

Demonstration on Urban Driving

We tested this extension to Icarus in the urban driving domain described earlier. Here we consider a single run at length to clarify the architecture's operation. Our aim is not to match human behavior in detail, but to show that the new system exhibits an important capacity of human cognition that its predecessor lacked. In this run, we placed the Icarus driving agent[3] in the leftmost lane on the right side of the road heading east. Another vehicle in the same lane was stalled in the road ahead. The agent's initial goals were `(on-right-side me)` and `(avoid-obstacle me)`. As the agent drives, it realizes that it is approaching the car ahead too rapidly and avoids colliding with it by swerving left. The agent swerves left rather than right simply because the associated skill happens to prefer that option, but this causes it to violate the high-priority goal, `(on-right-side me)`, by crossing to the left side.

At this point the agent realizes that it had ignored this high-priority goal while focusing on another one. Drawing its counterfactual reasoning abilities, the system creates a concept,

```
((on-right-side-and-avoid-obstacles ?self)
 :percepts ((self ?self))
 :relations ((on-right-side ?self)(avoid-obstacles ?self)))
```

that can serve as a new conjoined goal to direct its analysis. Next, the reasoning system backtracks through the episodic trace to the previously executed primitive skill, `(throttle-special-value me)`. Using time stamps on beliefs to reconstruct its mental state at that point, it invokes problem solving and mental simulation to search for another sequence of skills that achieves the goal `(avoid-obstacle me)` without violating the other one. In this case, the problem solver cannot find a solution that begins with this state, so the reasoner continues to backtrack through the earlier skill, `(crossing-into-left-lane me)`, which also fails to solve the problem. Eventually, after returning mentally to the state before `(wheels-straight me)`, problem solving finds a sequence of primitive skill instances,

```
(crossing-into-right-lane2 me), (wheels-straight me),
(on-right-side-lane2 me), (lane-aligned me),
(wheels-straight me)
```

that, if executed, would have achieved the conjoined goal `(on-right-side-and-avoid-obstacles me)`. Analysis of this solution using the adapted means-end problem solver leads to creation of a single new skill

```
((on-right-side-and-avoid-obstacles ?self)
 :percepts  ((self ?self))
 :start     ((on-right-side-lane1 ?self)
             (drone-ahead ?self ?drone ?dist ?angle))
 :subgoals  ((avoid-obstacles-by-right ?self)))
```

which is indexed by the new conjoined concept that, if satisfied, ensures that its component concepts are met.

On a subsequent run after learning with the same initial situation, the agent makes a different choice when

[3] The Icarus agent for this task included 23 skill clauses and 48 conceptual clauses, both organized hierarchically.

it approaches the stalled vehicle, swerving into the right lane rather than crossing over to the left side. The reason is that the architecture prefers skills that are indexed by more specific goals, thus masking the original preference for veering left over right. The result is that the agent still avoids hitting the stalled car ahead of it, satisfying the goal `(avoid-obstacle me)`, without violating the even higher-priority goal, `(on-right-side me)`.

Discussion

Counterfactual reasoning has been implicated in humans as a mechanism for establishing the cause of particular events (Roese, 1997), for identifying errors of both omission and commission (Byrne & McEleney, 2000), and for learning from errors (Roese & Olson, 1995; Wells & Gavanski, 1989). Our work with Icarus has focused on using counterfactuals to establish the cause of negative events (violations of maintenance goals) and to replace incorrect actions with proper ones. Although we have not yet shown that it can recover from errors of omission, we believe that the same mechanisms will support such learning.[4] Our account of counterfactual reasoning makes clear contact with psychological literature on the topic and, although our model makes some implausible assumptions (e.g., about memory), its main features are consistent with key theories and empirical findings.

Few computational models have made use of counterfactuals in the context of learning. One example, Mueller and Dyer's (1985) Daydreamer, uses them more broadly than Icarus but in a less directed manner. The system learns from both positive and negative experiences by postulating alternative actions and considering their consequences, but it proposes scenarios based on control goals, episodic memory contents, and emotional state. This strategy can produce a variety of outcomes, some substantially removed from reality, while Icarus pursues a single goal until achieving it. Pearson's (1996) Improv also makes use of counterfactuals to improve procedural knowledge. Like Icarus, it considers alternative action sequences starting from the last state before the error occurred, then working backward until it finds a solution. However, Improv focuses on revising skills that fail to achieve intended goals, while Icarus specializes skills that violate other goals it achieved previously. In addition, Improv revises its knowledge by modifying skill preconditions, rather than learning new skills that achieve more specific goals.

Other research on learning from errors has also focused on detecting and resolving errors, most on ones that stem from overly general rules. For example, early versions of the Swale system (Schank, 1986) adapt explanations to unanticipated situations when its expectations are violated. Similarly, Ohlsson (1996) shows how to correct errors that stem from overly general rules by comparing the actual and intended outcomes of selected actions, while Holland et al. (1986) describe a mechanism for specializing rules using counterexamples. Langley (1987) also reports an approach which compares similar situations that produce positive and negative outcomes to improve upon overly general rules. Work on analytical learning typically focuses on learning from success, but a few efforts (Carbonell et al., 1990; Laird et al., 1986) address learning from failure. These share our concern with explaining reasons for errors, but they produce control rules that specify what to avoid, while our approach instead acquires skills that mask the undesired behavior.

In addition to testing the architectural extensions in other domains that involve goal interactions, we should also improve our account of counterfactual reasoning along other dimensions. One involves increasing its psychological plausibility by placing realistic limits on the contents of Icarus' perceptual buffer and its episodic memory, which currently contain far more than their human analogs. We should also expand the generality of our counterfactual reasoning framework to learn from other types of errors, such as Ohlsson's constraint violations. In addition, we should extend the architecture's representation and its inferential abilities to let it reason about the goals and beliefs of other agents, since many of the most interesting errors that humans exhibit, and from which they are driven to learn, occur during their interpersonal interactions.

The main contribution of our work has been a computational account of skill learning through counterfactual reasoning. This involves three major steps: detecting that pursuit of one goal has violated another, reasoning backwards from the conflict to identify the choice that caused it and finding an alternative path that would have avoided it, and storing a specialized skill that produces the desired actions and masks the original behavior. Although we have embedded our account within Icarus, one could also incorporate it into other architectures, although some details would certainly differ. And although we have illustrated these mechanisms in the context of urban driving, they seem relevant to any domain in which goal conflicts can arise. We will not claim that our account covers all forms of learning through counterfactual reasoning, which may also support revision of incorrect concepts, skills, and beliefs, but we believe it advances our understanding of this complex ability, and thus our grasp of human cognition.

Concluding Remarks

In this paper, we presented a set of interacting computational mechanisms that support learning from undesirable outcomes via counterfactual reasoning. We embedded this account within Icarus, a theory of the cognitive architecture that placed strong constraints on our

[4] Ginsberg (1986) discusses the use of counterfactual reasoning in identifying subgoals during problem solving, which we have not addressed here.

approach to the problem. After reviewing the structures and processes that Icarus assumes, we presented new mechanisms that identify the violation of previously satisfied goals, localize the cause of this event by inspecting an episodic trace, invoke problem solving to find alternative steps that would have avoided the error, and learn specialized skills from this analysis that generate the desired behavior in the future.

We demonstrated these interacting mechanisms in the context of a simulated urban driving environment, showing that they behave as intended in a complex scenario that requires multi-step reasoning. We also considered earlier work on error-driven learning that bears similarities to our own, but that has addressed different issues, and promising directions for extending our approach. It seems clear that counterfactual reasoning plays an important role in human learning and, although our current model explains only certain forms of such cognitive behavior, it nevertheless provides a novel account of the mechanisms that underlie support this complex ability.

Acknowledgements

This material is based on research sponsored by ONR under agreement N00014-09-1-0123. The U. S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are the authors' and should not be interpreted as representing the official policies or endorsements, either expressed on implied, of ONR or the U. S. Government.

References

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum.

Anzai, Y., & Simon, H. A. (1979). The theory of learning by doing. *Psychological Review*, *86*, 124–140.

Byrne, R., & McEleney, A. (2000). Counterfactual thinking about actions and failures to act. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*, 1318–1331.

Carbonell, J., Knoblock, C., & Minton, S. (1990). Prodigy: An integrated architecture for planning and learning. In K. VanLehn (Ed.), *Architectures for intelligence.* Hillsdale, NJ: Lawrence Erlbaum.

Epstude, K., & Roese, N. J. (2008). The functional theory of counterfactual thinking. *Personality and Social Psychology Review*, *12*, 168–192.

Ginsberg, M. (1986). Counterfactuals. *Artificial Intelligence*, *30*, 35–79.

Holland, J., Holyoak, K., Nisbett, R., & Thagard, P. (1986). *Induction: The process of inference, learning, and discovery.* Cambridge, MA: MIT Press.

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

Langley, P. (1987). A general theory of discrimination learning. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development* (pp. 99–161). Cambridge, MA: MIT Press.

Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (pp. 1469–1474). Boston: AAAI Press.

Li, N., Stracuzzi, D. J., Cleveland, G., Könik, T., Shapiro, D., Molineaux, M., et al. (2009). Constructing game agents from video of human behavior. In *Proceedings of the Fifth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (pp. 64–69). Stanford, CA: AAAI Press.

Mueller, E. T., & Dyer, M. G. (1985). Towards a computational theory of human daydreaming. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society* (pp. 120–129). Irvine, CA.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1961). GPS: A program that simulates human thought. In H. Billing (Ed.), *Lernende automaten.* Munich: Oldenbourg KG. (Reprinted in E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York: McGraw-Hill.)

Ohlsson, S. (1996). Learning from performance errors. *Psychological Review*, *103*, 241–262.

Pearson, D. J. (1996). *Learning procedural planning knowledge in complex environments.* Unpublished doctoral dissertation, Computer Science and Engineering Division, University of Michigan, Ann Arbor, MI.

Roese, N. (1997). Counterfactual thinking. *Psychological Bulletin*, *121*, 133–148.

Roese, N., Hur, T., & Pennington, G. (1999). Counterfactual thinking and regulatory focus: Implications for action versus inaction and sufficiency versus necessity. *Journal of Personality and Social Psychology*, *77*, 1109–1120.

Roese, N., & Olson, J. (1995). *What might have been: The social psychology of counterfactual thinking.* Hillsdale, NJ: Erlbaum.

Schank, R. (1986). *Explanation patterns: Understanding mechanically and creatively.* Hillsdale, NJ: Erlbaum.

Stracuzzi, D. J., Li, N., Cleveland, G., & Langley, P. (2009). Representing and reasoning over time in a symbolic cognitive architecture. In *Proceedings of the 31st Annual Meeting of the Cognitive Science Society* (pp. 2986–2991). Amsterdam.

Wells, G., & Gavanski, I. (1989). Mental simulation of causality. *Journal of Personality and Social Psychology*, *55*, 161–169.