The Computational Problem of Prospective Memory Retrieval

Justin Li (justinnh@umich.edu)
John E. Laird (laird@umich.edu)
Computer Science Engineering, 2260 Hayward Street
Ann Arbor, MI 48109-2121 USA

Abstract

Although the need to remember to do things in the future is important and ubiquitous in human life, this phenomenon of prospective memory has received little attention in the cognitive modeling community. One roadblock is the lack of a computational definition of the phenomenon. We propose that prospective memory is fundamentally a problem of circular knowledge dependencies. This framework allows us to explain the success of human preemptive, spontaneous retrieval, and noticing-plus-search strategies. It also provides a clear mapping of the problem onto the capabilities of cognitive architectures. We interpret previous ACT-R and Soar models of prospective memory in this light, and identify areas of architectural development that would lead to more complete models of prospective memory.

Keywords: ACT-R; cognitive architectures; long-term memory; prospective memory; Soar;

Introduction

Prospective memory tasks — tasks that require the ability to remember to do things in the future — are ubiquitous in human life. From remembering to buy milk after work, to passing a message to a co-worker, to taking medicine at a specific time, prospective memory tasks are numerous and, in the last case, could have life-or-death consequences. Despite their importance, however, prospective memory has received little attention in the cognitive modeling community. This may be due to many reasons, but one stands out: there has yet to be a crisp computational definition of prospective memory tasks.

This paper attempts to provide such a definition. After outlining some assumptions about agents, we proceed by defining the retrieval problem, which lies at the heart of prospective memory. An analysis of this problem shows that agents with prospective memory must overcome circular knowledge dependencies. We use this insight to explain how three classes of human strategies succeed in overcoming these dependencies, and conclude by examining how these strategies have been or could be modeled in the ACT-R and Soar cognitive architectures. Our goal is that the identification of existing and to-be-developed architectural capabilities will provide guidance for both cognitive architectures and the modeling of human prospective memory behavior.

Agent Assumptions

In order to make generalizations about prospective memory, some assumptions must be made about the structure of agents. These assumptions are meant to apply to both humans and existing cognitive architectures, including ACT-R and Soar.

We assume that only a small amount of knowledge directs an agent's behavior at any time; this knowledge forms the *working memory* of the agent. *Procedural knowledge* matches against working memory to provide conditional behavior by adding or removing knowledge (*memory elements*) to working memory. Separate from working memory is *long-term memory*, which stores the knowledge accumulated over the agent's lifetime. Knowledge that is removed from working memory, either deliberately or through architectural mechanisms, are *forgotten*. Although the precise mechanisms differ, we assume that the likelihood of forgetting generally increases over time. Forgotten knowledge is recovered from long-term memory through deliberate *retrievals*. To perform a retrieval, the agent must use procedural knowledge to create a *cue* that describes a subset of the features of the desired memory element; the memory system then searches and returns the "best" memory element, as determined by some memory-system-dependent *bias*. One common bias is *base-level activation*, which is a function of the recency and frequency of access of the memory element.

We additionally assume that goals are no different from other memory elements, and are subject to forgetting. This follows prior work examining the goal management process (Anderson & Douglass, 2001; Altmann & Trafton, 2002).

The Prospective Memory Retrieval Problem

Although the number of prospective memory papers have been on the rise (Marsh, Cook, & Hicks, 2006), there has yet to be a concise definition of prospective memory tasks; they have only been described as a "fuzzy set" of intuitions around "remembering to *do something* at a particular *moment (or time period) in the future*" (emphasis in original) (McDaniel & Einstein, 2007). A small set of agreed-upon definitions does exist. First, in order to classify as a prospective memory task, the agent performing the task must explicitly represent an intention. Since the intention can only be acted upon in the future, prospective memory tasks are also called *delayed intentions*; we use these terms interchangeably. Each intention contains a description of the conditions (the *target*) under which the agent must execute some behavior (the *action*). An intention is only successfully completed if the action is performed while the target is present.

As a concrete example, consider an agent that intends to buy milk after work. The target for this task is the period during which the agent is driving home; the action is for the agent to drive to the grocery store to pick up milk. Other pairs of targets and actions are possible for this task: the agent could set the target as arriving at a specific intersection, with the action of turning towards the grocery store. We do not consider the choice of targets and actions in this paper.

Incorporating the above agent assumptions, a prospective memory task can be divided into five stages (Ellis, 1996). Although we use the term "perception", the target can be any representation in working memory, which may include both external perception and internal reasoning.

Encoding The target and action are stored into the long-term memory of the agent.

Retention The agent pursues other goals while waiting for the perception of the target. The intention may be relegated to long-term memory

Initiation The agent perceives the target, and a window of opportunity arises. The agent must recognize it as the target of an intention.

Execution The agent performs the stored action.

Completion The agent must modify its memory such that the next perception of the target does not lead to action.

This paper focuses on the initiation stage, as it is the crux of the prospective memory problem. For the agent to recognize the target, the perceptions of the agent must be compared to the stored intention. Given that the retention interval may be of unknown length, it can be assumed that the intention has been forgotten and only resides in long-term memory. Thus, the intention must first be retrieved from long-term memory; this is the *retrieval problem* of prospective memory.

To limit the scope of this analysis, two additional distinctions need to be drawn regarding prospective memory tasks.

First, the retrieval of the target into working memory does not guarantee the successful completion of a prospective memory task. A target present in the environment may not be perceivable by the agent, and a target perceived may not be correctly recognized. The first *perceptual* problem is demonstrated by time-based targets (e.g. it's 5:30pm) as compared to event-based targets (e.g. leaving the office), as the agent may need to deliberately look at a clock. A similar problem exists with using *external memory* (e.g. writing down appointments in a calendar), which the agent may not have available at all times. We focus on the former in this work. The second *knowledge* problem stems from the agent's lack of knowledge. The agent may not realize that a target is present (e.g. not knowing that a convenience store sells milk), or may require additional reasoning before being match the target (e.g. not recognizing it is leaving work after a day at the coffee shop). Additionally, other errors can occur after the initiation stage (Kvavilashvili & Ellis, 1996). While these issues are relevant to modeling prospective memory as a whole, they are beyond the scope of this paper.

A final distinction is between episodic tasks and habitual tasks. *Episodic* tasks (e.g. passing on a message) are performed on an irregular basis, while *habitual* tasks (e.g. taking medicine) are performed routinely. Agents could conceivably learn from the repetition of habitual tasks, such as by acquiring procedural knowledge that leads directly to action, thus bypassing the need to form an intention. We focus only on episodic tasks to ignore such effects of learning.

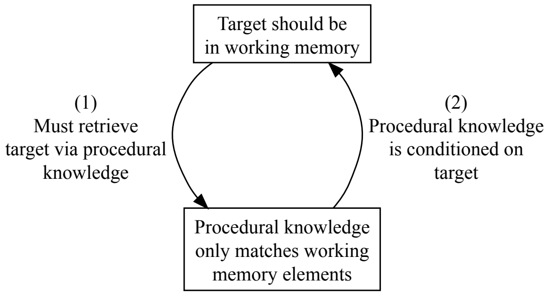
Target should be in working memory
(1)
Must retrieve target via procedural knowledge
(2)
Procedural knowledge is conditioned on target
Procedural knowledge only matches working memory elements

Figure 1: The knowledge dependency cycle of the prospective memory retrieval problem.

To summarize, we are interested in the problem of ensuring that the target of an episodic, event-based intention is in working memory during a window of opportunity.

A Computational Definition

The computational problem of prospective memory retrieval can now be defined. The goal of this analysis is to determine how an agent can ensure that the target of an intention is in working memory at a moment of opportunity.

We have already established that the agent must use procedural knowledge to retrieve the intention from long-term memory. It is unclear, however, what working memory elements this procedural knowledge should be conditioned upon — that is, when should the retrieval occur? The naïve solution is to retrieve the intention only when it is needed, when the target is perceived. This solution fails, however, since without the intention in working memory, the agent could not recognize that the target is being perceived to begin with. Without this recognition, procedural knowledge has nothing to match against,oarand the attempt to retrieve the intention fails.

The fundamental computational problem of prospective memory retrieval is therefore that the necessary actions have *circular knowledge dependencies* (Figure 1). In order to retrieve a target into working memory, the agent must use procedural knowledge. This procedural knowledge, however, can only match on working memory, in which the target must exist in the first place. To provide a concrete example, in the buying milk task, recognizing that leaving work is a target requires the agent to recall that it has a delayed intention to perform when it leaves work (dependency 1). And yet, knowing to recall depends on the recognition that leaving work is a target (dependency 2). In fact, it is impossible to use working-memory-conditioned procedural knowledge to retrieve an intention at the time of target perception. Either the target is already in working memory, rendering moot the point of the retrieval, or the target is not in working memory, and the procedural knowledge for retrieval never applies.

Although we have derived the dependency cycle from the need to access the stored target, a similar dependency cycle exists for perceiving the target. One can imagine an agent which has all of its intentions in working memory, but only has partial observability of its environment. This is exactly the case for time-based intentions and external memories, where the agent must look at the clock or calendar. The agent must know when to redirect its perception to the target, but the procedural knowledge that causes this behavior cannot match unless the target is already being perceived. There is symmetry in matching the target in memory to the target in perception, in that both require the agent to overcome a knowledge dependency cycle. Of course, these two dependency cycles are not mutually exclusive — such is the case for forgetful humans with a time-based target, where the agent is not in sight of a clock and has forgotten the target time.

Although both cycles could cause the agent to fail to initiate the intended action, this paper only addresses the more important problem of memory dependencies. With the target in working memory, the agent could at least use the inefficient approach of constantly redirecting its perception, in hopes of observing the target. Without the target in working memory, however, the agent would not even realize it has an uncompleted intention, never mind trying to recognize a target in its perception

Strategies for the Retrieval Problem

In this section, we examine how prospective memory strategies observed in humans fit into and break into the knowledge dependency cycle.

Three major classes of prospective memory strategies have been identified in humans:

- *preemptive strategies* require the agent to act before the perception of the target, both to retrieve the intention and to compare the target to the agent's perceptions. These are a superset of monitoring strategies, explained below.

- *spontaneous retrieval strategies* rely on the architecture to automatically provide the agent with relevant memory elements at the right time — in this case, with the intention at the time of target perception.

- *noticing-plus-search strategies* uses metamemory judgments (explained below) to signal the agent that connections to long-term memory exist, prompting the agent to search memory for the source of that connection — in this case, that it is the target of an intention.

Since preemptive strategies requires the agent to act prior to the initiation stage, it has traditionally be considered separately from the other two strategies. This has worked well for psychologists, as it may be difficult to experimentally distinguish between automatic memory retrievals and semi-automatic metamemory judgments. In computational models, however, these distinctions are more easily made, as we have access to the internal processes of the agent. We therefore propose an alternate taxonomy of strategies, one based on the required capabilities of the agent. In both preemptive and noticing-plus-search strategies, the agent is required to perform deliberate (procedural-knowledge-controlled) memory retrievals, whether it is before the target is perceived or when a metamemory judgment is made. Spontaneous retrievals, on the other hand, do not require deliberate action on the part of the agent; rather, the architecture automatically supplies the agent with a memory element, and the agent must decide whether and how to use the knowledge provided.

This distinction between deliberate and automatic retrievals also categorizes the strategies by how they break into the knowledge dependency cycle. The spontaneous retrieval strategy does not require the use of procedural knowledge, and thus removes the first dependency in Figure 1. Preemptive and noticing-plus-search strategies, in contrast, both change the conditions of retrieval; they therefore remove the second dependency in Figure 1. In the following subsections, we examine each of the three classes of strategies in turn. For each, we ask the following questions:

- How does the strategy break into the knowledge dependency cycle?

- What knowledge is required for the strategy, how is that knowledge processed, and how is the result used?

- Under what circumstances is the strategy likely to fail?

Preemptive Strategies

Our usage of the term "preemptive strategies" encompasses two psychological phenomena. The first is the recalling of intentions during the retention interval, which occurs periodically for time-based targets and during context switches for event-based targets (Sellen, Louie, Harris, & Wilkins, 1997). The second phenomenon is *monitoring*, where people continually expend resources to prevent the forgetting of the intention and to compare its perceptions to the target. These resources are attentional, but in cognitive models they may also include working memory capacity. Both phenomena require the agent to act before perceiving the target (and therefore prior to the initiation stage), hence the name of these strategies. By not using the target as a condition for behavior, preemptive strategies remove the second dependency in Figure 1.

There are two phases in preemptive strategies: that of retrieving the target, and that of comparing it to perception. Both of these phases depend on procedural knowledge. In the retrieval phase, the target should be retrieved immediately prior to its perception. An earlier retrieval requires more resources to keep the target in working memory and to compare it to perception, while a later retrieval could lead the agent to miss the opportunity to act. Taking inspiration from psychology, this trade-off can be balanced by retrieving targets during *context switches*, when a significant portion of the agent's perceptions are likely to change. This minimizes the time before the possible perception of the target while remaining prior to it. During the context switch, procedural knowledge directs the agent to search long-term memory for uncompleted intentions. Any additional information about future perceptions, such as a model of the environment, narrows the number of potentially relevant intentions to search through. For example, for buying milk, the agent may be designed to always recall its goals when stepping away from its desk or when leaving work.

Once the target is retrieved, a different mechanism is required to keep the target in working memory. The goal of this phase is not to decide what targets will likely be perceived, but to prevent any retrieved targets from being forgotten. Instead of context switches, which do not contribute to the forgetting of working memory elements, a different condition for action is needed. Since working memory elements are forgotten over time, procedural knowledge that applies periodically could delay the forgetting of a target until it next applies again, thus ensuring the target remains in working memory. This is exemplified by how an agent might repeat to itself the need to buy milk at every intersection on the way home. Finally, additional procedural knowledge may be used to signal that the target is not likely to appear and thus can be forgotten, to be retrieved again at a later point.

These two phases cover the space of observed human phenomena mentioned at the beginning of this subsection. Preemptive strategies have the advantage that they only rely on domain-dependent knowledge, and do not require architectural support — that is, no additional information is needed from the architecture. They are, however, highly sensitive to the temporal dynamics of the domain; parameters such as the rate of forgetting, the layout of the environment, and even the speed of movement, can determine whether a particular preemptive strategy succeeds. As such, the context-switch detectors and interval timers of these strategies should be adjusted online by an adaptive algorithm.

Spontaneous Retrieval Strategies

Spontaneous retrieval strategies, as the name implies, rely on automatic processes to retrieve intentions into working memory. Also known as *remindings* (Hintzman, 2011), this process is entirely architectural and does not require the use of procedural knowledge; this allows spontaneous retrieval strategies to sidestep the knowledge dependency problem. As a concrete example, this is equivalent to the intention of buying milk somehow "popping" into the working memory of the agent as it drives home.

Since spontaneous retrievals are entirely automatic, the agent cannot create retrievals cues to specify what should be retrieved. At the same time, the retrieved element must be relevant to the agent's current situation. These opposing constraints are most easily simultaneously satisfied if the architecture uses some subset of the agent's working memory as the cue. While this subset may be determined in many ways (e.g. activation), for this analysis it is sufficient to note that working memory is the only knowledge required for spontaneous retrieval strategies. Whatever portion of working memory is used, the declarative long-term memory system uses it to retrieve a single memory element into working memory. In the description of spontaneous retrievals thus far, the system is not constrained to only retrieve uncompleted intentions. Other information may also be retrieved, and indeed this may true for the majority of retrievals. Thus, for prospective memory, the agent must check that the retrieved element is an intention and that the target matches its current perceptions. If this is the case, the agent may then decide to take the intended action.

Although no cognitive architecture that we know of currently supports spontaneous retrievals, a preliminary analysis can determine when such a strategy will fail. Setting aside the question of which subset of working memory is used as the cue, the result of a spontaneous retrieval is highly dependent on the biases of the memory system. The effectiveness of different biases is an active area of research (e.g. Derbinsky and Laird (2011)), which is only complicated by the lack of a deliberate cue to filter out non-intention results. The complete lack of agent control that allows the strategy to work is also its biggest weakness, as the agent cannot guarantee the utility of the retrieval. Given the large space of memory elements that could be retrieved, intentions could be a small minority. The success of this strategy, therefore, depends entirely on the retrieval bias, as well as the balance between the generality of the system and prospective memory task performance.

Noticing-Plus-Search Strategies

Noticing-plus-search (NPS) strategies incorporate components of both spontaneous retrievals and preemptive strategies. NPS works by using a second, automatic, non-retrieval channel to memory, which we call *metamemory judgments* or simply memory *metadata* (examples below). This channel signals that parts of working memory might have connections to elements in long-term memory. Thus drawing the agent's attention to (*noticing*) a memory element, the agent may choose to use it as a cue to *search* memory for the source of this connection. On retrieval, if the source of this connection is an intention, the agent then verifies that the target is present and performs the necessary action. Since procedural knowledge is not dependent on the recognition of the target but instead on metadata about the agent's perceptions, NPS strategies remove the second dependency in Figure 1. As an example of noticing, a metamemory judgment may be made when the agent sees a bottle of milk at work, prompting it to search for and retrieve the intention to buy milk.

The key knowledge that enables NPS strategies is the metadata that the agent receives from the architecture. Psychology literature suggests several kinds of metamemory judgments that may indicate an object or event has been previously stored in memory (Yonelinas, 2002). The most direct are familiarity and recognition judgments, which convey similarities between current and previous perceptions. More subtle is the noticing of discrepancies between the expected and actual processing fluencies. Regardless of the type, metamemory judgments suggest cues with which the agent could search memory. As with spontaneous retrievals, metamemory judgments do not benefit only prospective memory; the long-term memory connection may originate from some source other than an intention. Unlike spontaneous retrievals, however, the agent can decide whether to retrieve and whether to add additional cue constraints; for prospective memory, the agent could specify the result to be an intention, thus filtering out irrelevant memory elements.

The likely failure point for NPS strategies is not the number of irrelevant memory retrievals, but the number of judgments that are unrelated to prospective memory. The agent must decide which of many judgments are related to uncompleted intentions, such that the number of (potentially unfruitful) memory retrievals are minimized. This is the same problem as with spontaneous retrievals: both strategies require trading off between the generality of an architectural mechanism and the performance on prospective memory tasks. Since the strategies differ in where the architecture must provide information, they also differ in where this trade-off occurs. Our minimal assumption that goals are ordinary memory elements do not suggest a point in this trade-off; we leave the exploration of this space for future work.

Support from Cognitive Architectures

In this section, we look at how the three strategies described earlier could be implemented in two cognitive architectures, ACT-R and Soar, chosen due to their widespread use. For each architecture, we describe previous prospective memory work in the context of the dependency cycle, then explore how each strategy could be implemented.

ACT-R

ACT-R (Anderson, 2007) is a cognitive architecture designed with the goal of matching human timing data. The working memory of ACT-R consists of a fixed number of fixed-sized *buffers* to various *modules*, including *declarative memory*, which serves as the architecture's long-term memory. Procedural knowledge is in the form of *production rules*, which match on and modify working memory, and can access declarative memory through its buffer. Within declarative memory, retrievals are biased by *activation* — the higher the activation, the more likely it is for a memory element (*chunk*) to be returned, provided that it matches the cue. Activation follows the *base-level activation* formula, a function of the recency and frequency of access for that the memory element. Additionally, activation can *spread* from one element to another, although the scope of this spreading is limited for computational reasons.

Although previous researchers have looked at several prospective memory related problems, few have looked at the problem of retrieval as a whole. Elio (2006) compared two schemes of prospective memory retrieval on how they match human timing data. An "intention monitoring" strategy first retrieves uncompleted intentions, then directs the agent to test specific aspects of the environment to determine if the target matches. In contrast, an "intention cueing" strategy first elaborates on the agent's current perceptions, then relies on spreading activation to bias which intention gets retrieved. Although these strategies are superficially similar to a preemptive strategy and a spontaneous retrieval strategy, they do not in fact tackle the retrieval problem. Crucially, the agent in this work is given knowledge as to when it is appropriate to retrieve an intention; indeed, the only challenge for the agent is to determine which intention is the correct one to act on. A similar issue can be found in Lebiere and Lee (2002), which looked at the *intention superiority effect*, where uncompleted intentions are more easily recalled than completed intentions.

The most complete account of prospective memory in ACT-R comes from Altmann and Trafton (2002). In that work, the authors looked at how super-goals could be retrieved and resumed after a sub-goal has been completed in the Tower of Hanoi puzzle. To ensure that the super-goal remains retrievable from declarative memory, the agent must boost its activation sufficiently before beginning the sub-goal — a form of one-time preemptive strategy. Once the sub-goal is completed, the agent then retrieves and resumes the super-goal. In fact, this work presents a special case of preemptive strategy, where the target — the completion of the sub-goal — is itself a context switch, and therefore perfectly predicts the need to act. Since the agent immediately resumes the super-goal, there is no need to prevent the intention from being forgotten while waiting for the target. The Tower of Hanoi agent therefore provides a full solution to the prospective memory retrieval problem. The strategy used, however, is a very domain-specific; in addition to the context-switch target, the recursive nature of the puzzle also allows the agent to determine the necessary amount of activation boost. Such information may not be available in other tasks.

Despite the lack of prospective memory agents in ACT-R, the architecture has the potential to support each of the three strategies. Although ACT-R does not support spontaneous retrievals, spreading activation allows working memory elements to influence deliberate retrievals. Spontaneous retrievals could disrupt background reasoning if the declarative memory buffer is overwritten, but this is easily allayed by adding a buffer specific for this purpose. For preemptive strategies, the work of Altmann and Trafton (2002) provides a starting point, although generalization is necessary for the agent to be usable in other domains. Finally, for noticing-plus-search strategies, the approach taken with Soar (Li, Derbinsky, & Laird, 2012) may allow metamemory judgments to be computed efficiently. In particular, the chunk merging capability of ACT-R already requires connecting new percepts with previous memory elements; this provides an entry point for recognition metadata. Implementing other types of metamemory judgments may require additional theoretical advances in both psychological and ACT-R theory.

Soar

The Soar cognitive architecture (Laird, 2012) focuses on real-time use of large amounts of knowledge. Soar's procedural knowledge is also formulated as production rules, which operates on the directed graph that forms Soar's working memory. There are also specialized buffers that allow access to Soar's long-term memories. Although there is no limit on the size of working memory, the need for real-time reactivity constrains how large working memory can be. Soar contains two long-term memories: semantic memory, which stores knowledge about the world, and episodic memory, which contains the past experiences of the agent.

There has only been one study of prospective memory in Soar, done by Li and Laird (2011). In that work, the authors used rules to bring relevant intentions into working memory at the right time. An intention is encoded into a rule, which fires when the target is perceived and brings the intention into working memory. Since it is difficult to modify procedural memory, such a strategy is more suited for habitual tasks; as such, this strategy does not neatly fit into any of the three classes above. Although the resulting agent solves the prospective memory retrieval problem, the authors noted that the approach is not scalable and that the agent loses reactivity as more intentions are formed and completed.

As with ACT-R, Soar has the potential to support all three strategies. To support spontaneous retrievals, additional buffers have to be added to working memory to avoid overwriting deliberate retrievals. Unlike ACT-R, Soar does not currently have a mechanism for the contents of working memory to automatically influence the retrieval process. Although it is possible to use the entire working memory as a cue, this is computationally expensive (Laird, 2012). These are engineering problems, however, which do not rule out the possibility of a solution. For preemptive strategies, since they involve additional procedural knowledge and do not require architectural support, there are no roadblocks in creating a Soar agent which uses them. Indeed, these strategies were explored in Li and Laird (2013), which demonstrated the dependence of the strategy on the temporal dynamics of the domain. Finally, for noticing-plus-search strategies, there has been work in conveying recognition and other metamemory judgments to Soar agents (Li et al., 2012). Although the mechanism was not designed for prospective memory, the availability of such metadata is a step towards a functional agent.

Concluding Remarks

Describing prospective memory as a computational problem is a key step towards modeling human strategies. We submit that the dependency cycles are critical features of prospective memory tasks, and they must be embedded into the domain for any complete model of prospective memory. These dependency cycles allow us to explain three classes of human strategies, and furthermore, they provide a mapping of these strategies onto the capabilities of cognitive architectures. This mapping proposes areas of development for cognitive architectures, which may be a path towards better modeling of human prospective memory.

Acknowledgments

The authors acknowledge the funding support of the Office of Naval Research under grant number N00014-08-1-0099.

References

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science*, *26*(1).

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

Anderson, J. R., & Douglass, S. (2001). Tower of hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*(6).

Derbinsky, N., & Laird, J. E. (2011). A functional analysis of historical memory retrieval bias in the word sense disambiguation task. In *Proceedings of the 25th AAAI conference on artificial intelligence.*

Elio, R. (2006). On modeling intentions for prospective memory performance. In *Proceedings of the 28th annual conference of the Cognitive Science Society.*

Ellis, J. (1996). Prospective memory or the realization of delayed intentions: A conceptual framework for research. In M. Brandimonte, G. O. Einstein, & M. A. McDaniel (Eds.), *Prospective memory: Theory and applications.* Lawrence Erlbaum.

Hintzman, D. L. (2011). Research strategy in the study of memory: Fads, fallacies, and the search for the "coordinates of truth". *Perspectives on Psychological Science*, *6*(3).

Kvavilashvili, L., & Ellis, J. (1996). Varieties of intention: Some distinctions and classifications. In M. Brandimonte, G. O. Einstein, & M. A. McDaniel (Eds.), *Prospective memory: Theory and applications.* Lawrence Erlbaum.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT Press.

Lebiere, C., & Lee, F. J. (2002). Intention superiority effect: A context-switching account. *Cognitive Systems Research*, *3*(1).

Li, J., Derbinsky, N., & Laird, J. E. (2012). Functional interactions between memory and recognition judgments. In *Proceedings of the 26th AAAI conference on artificial intelligence.*

Li, J., & Laird, J. E. (2011). Preliminary evaluation of long-term memories for fulfilling delayed intentions. In *AAAI 2011 fall symposium on advances in cognitive systems.*

Li, J., & Laird, J. E. (2013). Preemptive strategies for overcoming the forgetting of goals. In *Proceedings of the 27th AAAI conference on artificial intelligence.* (in press)

Marsh, R. L., Cook, G. I., & Hicks, J. L. (2006). An analysis of prospective memory. *Psychology of Learning and Motivation*, *46*.

McDaniel, M. A., & Einstein, G. O. (2007). *Prospective memory: An overview and synthesis of an emerging field*. Sage.

Sellen, A. J., Louie, G., Harris, J. E., & Wilkins, A. J. (1997). What brings intentions to mind? An *in situ* study of prospective memory. *Memory*, *5*(4).

Yonelinas, A. P. (2002). The nature of recollection and familiarity: A review of 30 years of research. *Journal of Memory and Language*, *46*(3).