Opportunistic Reasoning: A Design Perspective

Marin D. Simina
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
marin@cc.gatech.edu

Janet L. Kolodner
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
jlk@cc.gatech.edu

Abstract

An essential component of opportunistic behavior is *opportunity recognition*, the recognition of those conditions that facilitate the pursuit of some suspended goal. Opportunity recognition is a special case of situation assessment, the process of sizing up a novel situation. The ability to recognize opportunities for reinstating suspended problem contexts (one way in which goals manifest themselves in design) is crucial to creative design. In order to deal with real world opportunity recognition, we attribute limited inferential power to relevant suspended goals. We propose that goals suspended in the working memory monitor the internal (hidden) representations of the currently recognized objects. A suspended goal is satisfied when the current internal representation and a suspended goal "match". We propose a computational model for working memory and we compare it with other relevant theories of opportunistic planning. This working memory model is implemented as part of our IMPROVISER system.

Introduction

During a mechanical engineering project a group of students were asked to design and implement a mechanical device for the quick and safe transportation of a fragile cargo (some eggs). The students went to a Home Depot (a hardware store), where they started by choosing springs for the launching component. During the design process they made the following observations:

Andy: ... hey, when I compress the spring it bends; this weakens the force of the springs ...

Mary (wrapping her hand around the spring): ... yes, we have to enclose it in a tube ...

Bill: ... the tube should be collapsible, otherwise the spring cannot be compressed ...

The students began proposing mechanisms that fit this description. One of them suggested a telescope, but it was rejected by the group because it was expected to be costly and it did not fit in the available budget. Another student proposed a collapsible camping tube, which unfortunately has a wrong shape. The designers were unable to think of where in the store they might look for a useful collapsible tube, so they moved to another part of their problem. They started thinking about load protection. Since sponges are a good way to provide cushioning, they decided to go to the store's bathroom section. During the search for sponges, one of the students saw a toilet paper holder and exclaimed:

Mary: Look! A collapsible tube!

The whole group agreed that the toilet paper holder fulfilled the requirements of their previously suspended problem.

The above example illustrates a rather mundane, but common, experience in doing design. The students started by structuring the initial problem (launching, cushioning) and then they tried to elaborate the subcomponents, one at a time. When they were stuck with one subproblem, they suspended it, and they approached another related subproblem. When they saw the toilet paper holder, however, they recognized that an opportunity to address the suspended subproblem had presented itself in the environment.

What processes are responsible for recognizing such opportunities? How can a cognitive architecture handle this kind of processing? What constraints are there, if any, on the workings of these processes? We are studying these problems in the context of developing a cognitive model for creative design. Our computer program, IMPROVISER (Wills & Kolodner 1994b), was extended in order to help us answer the above questions.

Our exploration of creative design (Kolodner & Wills 1993a) suggests that the conceptual phase, in which the problem is framed, plays a key role in designing. In this phase, which is interspersed throughout the design process, the problem situation is assessed and the given problem is reformulated and restructured. While one can organize the subgoals involved in conceptual design in a hierarchical structure, the processing of these subgoals seems far more unstructured. Designers often begin by proposing a shallow hierarchical set of subgoals as they initially formulate the way they will solve a problem (e.g., the artifact we are designing has these n parts or mechanisms; we need to design each one). They continue by addressing each of the subgoals, one at a time. It is here where the organized reasoning breaks down. When the designer fails to solve one subproblem, he/she seems to suspend it and approach another related subproblem (as in the example above). Sometimes the next subproblem is simply a not-yet-considered sibling subgoal (as when the student designers moved from designing their spring launch mechanism to the cushioning for the eggs); sometimes the opportunity to go back to a suspended subgoal is recognized (as when the toilet paper holder was seen).

When we consider the incremental and recursive nature of this reasoning process, we can identify one way of recognizing that a previously-suspended subgoal might be successfully addressed. During consideration of a new subproblem, the designer has to consider interactions with related subproblems, some of which have been suspended previously. This may provide a fresh view of the suspended problem and a new way to redescribe it. Redescription or new insights about a subproblem gained during reasoning trigger the goal scheduler to unblock the suspended subproblem, allowing already-known solutions to be recalled or new means of solving it to be recognized. This means of unblocking a suspended goal is completely under the control of the reasoner, which knows which subproblems have been part of its most recent reasoning.

But recognizing in the toilet paper holder the opportunity to address a suspended goal requires additional mechanisms that scan the environment and recognize when the environment is providing new insights into suspended goals. If the number of suspended goals, the complexity of the environment, or the amount of newness in the environment is high, such a mechanism could easily be overwhelmed. The mechanisms that provide this capability must be able to deal with such complexity.

Opportunity Recognition

The Problem

The prerequisite for opportunistic behavior is the existence of *suspended goals* (problems), goals that cannot be pursued in the current context and are postponed.

An essential component of opportunistic behavior is *opportunity recognition*, recognition of those conditions that facilitate the pursuit of some suspended goal. But opportunities seem to appear when they are not expected. The student designers, for example, had not previously thought about a toilet paper holder functioning as a collapsible tube. Recognizing the opportunity meant both noticing the toilet paper holder and recognizing that its mechanism (which is hidden) included a collapsible tube. More than a simple matching mechanism is needed.

Birnbaum (1986) suggests two central problems that must be addressed by a theory of opportunistic behavior: (1) how to detect opportunities and (2) how to "activate" the goals to which they pertain. An important issue here is identifying how much and what kind of processing is required in order to recognize the presence of the features that constitute an opportunity.

A Critical Review

Hayes-Roth & Hayes-Roth (1979) proposed the first significant cognitive model of opportunistic behavior. Their model of opportunistic planning was inspired by protocols of subjects planning a hypothetical day's errands. But they were most concerned with planning methods and gave little attention to recognition processes. In fact, the experimenter always mentioned opportunities to the subjects when they overlooked them, and the subjects never tried their plans in the real world, so they never really dealt with genuine opportunities and the problem of recognizing them.

Birnbaum (1986) gave more attention to recognition issues. He proposed the *mental notes* model, in which whenever a goal cannot be immediately satisfied, it is indexed in terms of the unmet preconditions that prevented its satisfaction. However, as he points out, if the goal is indexed too specifically, then there will be many cases in which it will not be recalled even though an opportunity for its satisfaction is present, and if the goal is indexed in terms of more abstract features, we cannot assume that the agent will automatically generate the abstract description that will activate the goal.

In order to solve the above dilemma within the framework of the mental notes model, Birnbaum suggests[1] spending some effort, when the goal is formed, to determine the range of situations in which it might easily be satisfied – for example, by constructing several incomplete plans for the goal in order to identify the relevant preconditions – and then indexing the goal in terms of the features that might arise in such situations. Birnbaum & Collins (1984) also suggest an active goal framework, where all the goals have the ability to examine the current situation and to initiate inference to test their own relevance.

Patalano, Seifert and Hammond (1992) criticize the use of active goals proposed by Birnbaum & Collins, claiming that this approach to opportunistic behavior is an unlikely explanation of human cognitive processes because of its computational demands. However, Patalano, Seifert and Hammond do pick up on Birnbaum's indexing scheme, calling it *predictive encoding*. Predictive encoding stresses the importance of encoding blocked goals in memory in such a way that they will be recalled by conditions favorable for their solutions. Their experimental results show evidence of this process.

However, the predictive encoding hypothesis seems incomplete, because it does not enable a cognitive agent to recognize opportunities other than those which it is able to anticipate. In particular, it does not enable an agent to recognize novel opportunities, which by their very nature, cannot be easily anticipated. Recognition of the toilet paper holder as a collapsible tube, for example, is novel in that this is not the way a toilet paper holder is generally thought of. Similar issues caused Birnbaum & Collins (1984) to conclude that if an opportunity is to be detected at all, inferential resources must be allocated to the goal recognition task.

Ram & Hunter (1992) suggested a balance between backward chaining at the time of goal suspension and forward chaining at the time of opportunity recognition. In AQUA, a set of utility metrics have been proposed in order to make a tradeoff between predictive encoding and active goals. Unfortunately, these utility metrics are very specific to story understanding.

This suggests that we need active goals in order to recognize novel opportunities, but we need to control their power and number to make them computationally feasible. We need predictive encoding, but we also need more powerful inferential capabilities. We hope that an analysis of the example presented previously can provide insight in formulating a mechanism with these properties.

A Possible Solution

Why did the students fail to remember the toilet paper holder when they were trying to decide where they might find a collapsible tube, and what allowed them to recognize it as appropriate when they saw it?

One possible reason why the toilet paper holder was not recalled and considered while thinking about collapsible tubes is that the *probe* that had been constructed (i.e., the item description used for remembering) was incompletely specified. Consequently, they retrieved items that fulfilled primary but

[1] Birnbaum credits Dehn (1989) with this idea.

not secondary characteristics of the probe (e.g., a telescope costs too much and a camping cup has a wrong shape). After every retrieval and evaluation of a new device, the probe was respecified, taking into consideration the initially ignored constraints (e.g., we want something like a telescope, but cheaper). This process was suspended, however, before the toilet paper holder was recalled.

But why was the probe inadequate for retrieving such a common object as a toilet paper holder? Our explanation is that the toilet paper holder is routinely associated with what its purpose is in the bathroom (holding toilet paper rolls) rather than with how this function is achieved (by means of a collapsible tube with a spring inside). It is not a particularly interesting device, and even though we see it every day, most of it is hidden by the roll of paper. Research shows that it is quite difficult to overcome such *functional fixedness* (Mayer 1970), which associates everyday objects with their obvious function (holding a paper roll in the case of the toilet paper holder). Routinely, we ignore other *potential* uses that can be derived from the structure and behavior of such everyday objects. Once we have specified desired criteria in a probe, it is easy to check them against a specific object. But if those criteria are different than those used to describe an object in memory, recall won't happen.

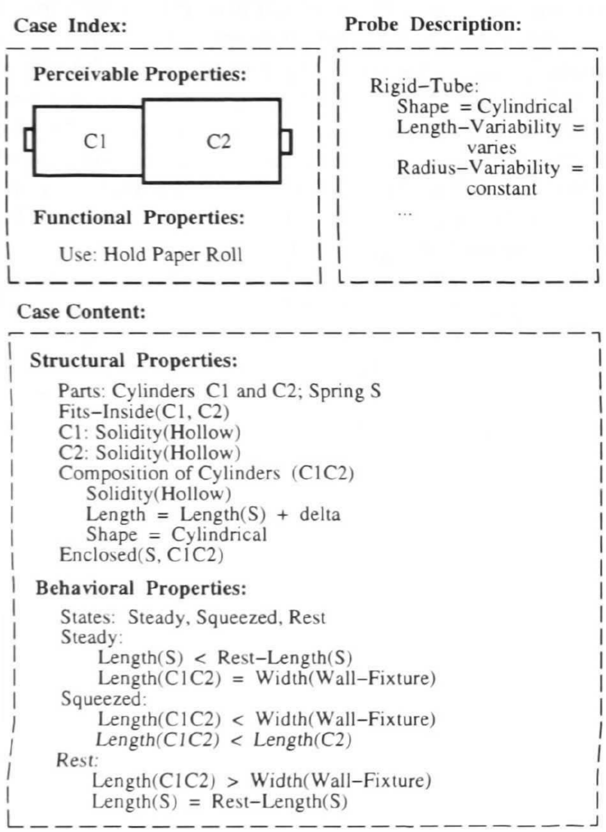

Figure 1: The many representations of a toilet paper holder

Figure 1 shows this mismatch. The collapsible tube, as described after manipulating the springs (see the PROBE DESCRIPTION in Figure 1), has the structural property that its shape is *cylindrical* and the behavioral property that its length can *vary*. The toilet paper holder, on the other hand, is indexed in memory by a combination of its *Functional Properties* and *Perceivable Properties*, shown as INDEX in Figure 1. Thus, we cannot retrieve the CASE CONTENT, namely the *Structural Properties* and *Behavioral Properties* of the toilet paper holder by using the PROBE DESCRIPTION.

What facilitates recognition of the opportunity in the environment, i.e., recognition that the toilet paper holder can fulfill the role of collapsible tube? On the store's shelf, one can see the shape of the device. Recognition procedures perceive that it is a collapsible tube, which matches the description from the retrieval probe and presumably the label that designates what needs to be encountered to unblock the suspended goal.

But what processes direct recognition procedures to attend to the toilet paper holder on the store's shelf? And what mechanisms allow matching of something in the environment to a goal that is no longer active? We know that memory search is incremental and that when our memories can't retrieve what we are asking them for, we redescribe what we are looking for and try again. But when we aren't making headway, we postpone additional retrieval until more information is gathered and pursue other retrieval strategies or subgoals (Williams & Hollan, 1981, Norman & Bobrow, 1979, Kolodner, 1984). Similarly (and implied by predictive encoding), we suspend reasoning subgoals and subproblems that depend on postponed retrieval strategies and unmatched probes, marking them with criteria that, if encountered, predict that they should be reopened (Patalano, Seifert and Hammond 1992).

We propose that when an active subgoal (subproblem) is suspended, the subgoal and its criteria remain in working memory's working store for some limited time. We further propose that goals suspended in the working memory continuously monitor the environment, looking for matches in the environment to the specified criteria. Furthermore, we suggest that there are only a small number of these *active* goals. A computational model will provide more detail on these limitations.

A Memory Model

The Memory Architecture

The major component of our computational model (presented in Figure 2) is a working memory (WM), which communicates with both long-time memory (LTM) and perceptual processes and keeps track of recent reasoning context. As Barsalou (1992) suggests, the working memory mediates between short-term memory (STM) and the activated part of LTM. But we add significantly to Barsalou's conception. First, we give the WM a structure. Second, the structure integrates components of STM with activated portions of LTM and with perceptual mechanisms and stores. Third, this integrated component acts as a buffer for LTM. It is the place where LTM's components are manipulated and adapted. Fourth, we add a control unit (matcher), which can match (1) the current artifact being reasoning about or (2) all the suspended problems against the LTM representation of the current item presented to the Recognizer.

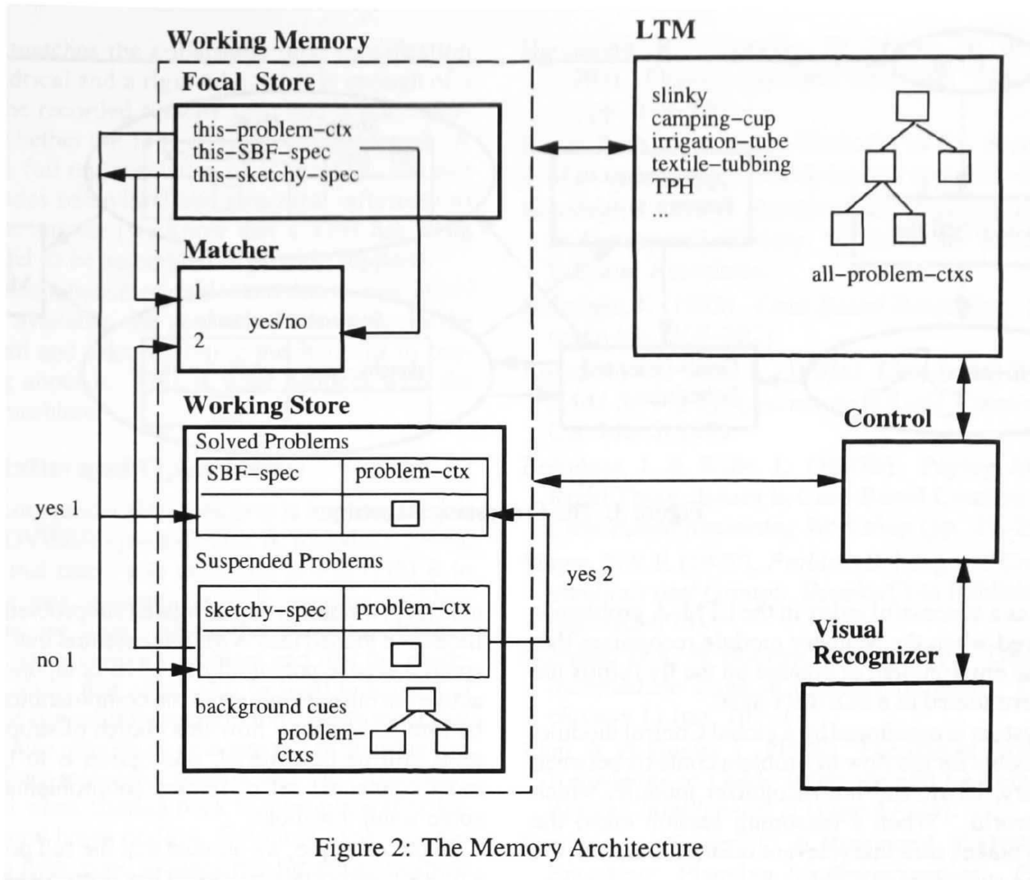

Figure 2: The Memory Architecture

The working memory that emerges has three parts: (1) Focal Store (FS), (2)WM Control unit (the only part of the control unit currently relevant is the Matcher), and (3) Working Store (WS).

The Focal Store (FS) holds three items: (1) the current goal of the reasoner (THIS-PROBLEM-CONTEXT), (2) the current object, artifact or idea being reasoned about (THIS-SKETCHY-SPEC, which is similar to the PROBE DESCRIPTION in Figure 1), and (3) the representation of the current item presented to the Recognizer, module (THIS-SBF-SPEC), which is retrieved from LTM according to the specification generated by the Recognizer.

The working store is more interesting and has four parts.

1. A connected graph of related unsolved subproblems, represented as subgoals and the contexts in which they are applicable, called *problem contexts* (represented as small rectangles in the figure). This graph might be a subset of a problem decomposition stored in LTM when the problem was previously considered, it may have been created during the reasoning session, or it may be a combination of the two. The goal of the reasoning session is to find a solution for the whole group of related problems.
2. Background cues, which provide a history of concepts, descriptions, features, and objects that have been considered during reasoning
3. A list of Suspended Problems, each represented by a problem context that includes the relevant subgoal, the context in which it is being considered, and the still-incomplete solution description (SKETCHY-SPECS). More specifically, the representation of suspended subproblems is modeled after the content of problem contexts in design. A problem context in design, and a suspended subproblem in working memory, includes (1) a set of goals and partially ordered constraints that solutions should satisfy; (2) a set of options, or alternatives for achieving those goals[2] ; and (3) a set of relationships describing how the options satisfy the constraints. These sets are incomplete and contain as much as has been considered so far in addressing the goals.
4. A list of Solved Problems, consisting of problem contexts for which solutions (SBF-SPECs) have been found. These problem contexts have the same structure as do suspended subproblems, but their solution descriptions are complete.

This working memory structure, in effect, keeps track of the part of LTM activated during a reasoning session. At most, then, the retention time of working memory is a few hours, requiring only a limited capacity (more work is needed before speculating on how big).

The working store accommodates several subproblems (PROBLEM-CTXs), which ideally are related, at the same time. These subproblems are approached one at a time, and if the current one cannot be solved, it is transferred to the list of Suspended Problems. Solved problems are transferred to the Solved Problems queue. A suspended problem is characterized by a non elaborated specification (SKETCHY-SPEC), which

[2] In our example the options set included a telescope, a camping cup and a slinky. This initial set of options was gathered by probing the LTM with a set of indexes relevant to the context goals.

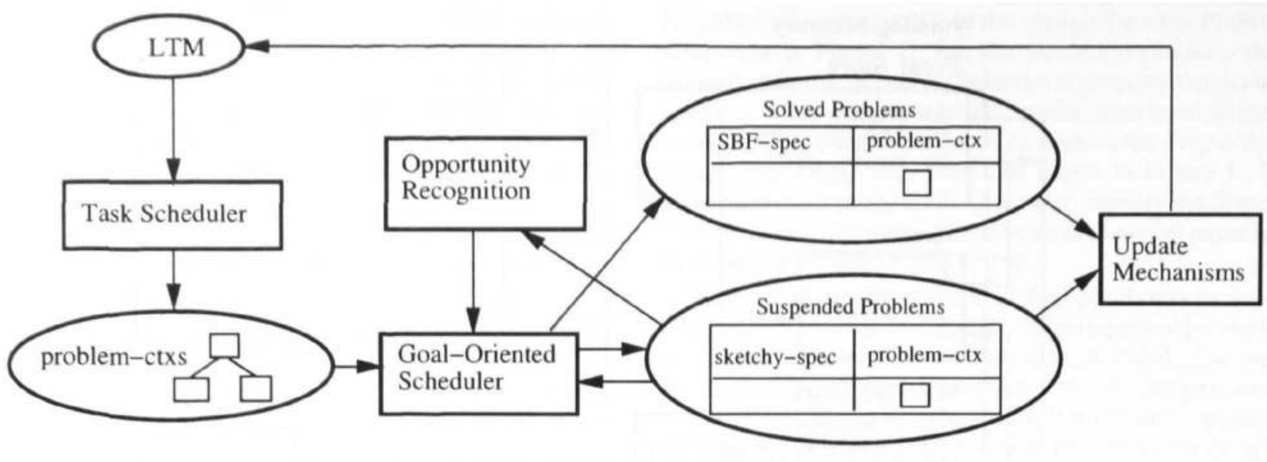

Figure 3: The Processing Algorithm

cannot be used as a successful index in the LTM. A problem is considered solved when the Matcher module recognizes that something in the environment or created on the fly fulfills the requirements formulated in a SKETCHY-SPEC.

The whole system is monitored by a global Control module, which is responsible for the flow of problem contexts between working memory, LTM, and the recognizer module, which perceives the world. When a reasoning session ends, the control module makes sure that relevant information from the WM updates the structures in LTM.

The Processing Algorithm

Mediation between working memory, long-term memory, and perceptual processes are key to working memory's functioning. Four control components (see Figure 3) are important to using working memory well: (1) The task scheduler loads a graph structure (a set of related subproblems) in the Working Store. (2) The goal-oriented scheduler uses the graph structure and the sets of suspended and pending problems to choose what to do next. Among other things, it suspends subproblems when no headway is being made; it reinstates them when their indexes (specs) are matched. (3) Opportunity recognition procedures notice opportunities to reinstate suspended goals and send messages to that effect to the scheduler. This is accomplished by having perceptual functions (the object recognizer is the only one of these in the scheme presented) focus their attention based on the sketchy specs recorded in suspended subproblems. For example, the object recognizer seeks to identify objects whose descriptions *partially match* the sketchy specs associated with suspended subproblems. When such an object is seen, the recognizer asks inference procedures if they can *quickly* determine if the object has other properties specified in the sketchy spec. If so, the opportunistic component notifies the goal scheduler that a suspended goal ought to be reinstated. (4) Update mechanisms update the structures in LTM based on recordings in WM.

When a new problem is approached, a hierarchical structure is proposed for it. Sometimes the structure is already recorded in memory; sometimes it is on paper; sometimes it must be constructed – we don't consider that issue right now. As a next step, a small group of related subproblems is brought into focus and loaded into WM. It is essential that this group is kept small, because potentially all of its components may become active during reasoning and the computational demand should be limited. Exactly how this choice of subproblems is made must still be discovered; one option is to bring in only the most connected set of related subproblems and only up to some small threshold.

In our example, we assume that the full problem (design of a quick transportation device) has been considered previously and that there are a set of subproblems recorded in memory. In the session we focus on, two subproblems are brought to attention and loaded in WM: the launching device problem and the cushioning material problem. The graph structure in WM has the full problem at the top and these two subproblems hanging off of it as sibling subproblems.

One of the subproblems is chosen for focus, and it is loaded into the Focal Store as the current problem (THIS-PROBLEM-CTX). Here, the launching device problem is chosen first. Reasoning procedures work on this problem until it is solved, in which case it is put into the solved problems queue, or until no progress can be easily made, in which case it is added to the queue of suspended problems. When the need for a collapsible tube emerged in solving the launching device problem, no useful device was recalled from LTM, nor was one seem immediately on the shelves of the store. Thus, this subproblem is suspended. The description created of the collapsible tube (the probe in Figure 1) is used as the SKETCHY-SPEC for this suspended problem. When a subproblem is suspended, a new problem is chosen to work on. Here, the cushioning subproblem is selected, and reasoning procedures begin working on it.

At the same time, perceptual functions are scanning the environment, looking particularly for things that partially match sketchy specs of suspended problems[3] In our case, the object recognizer notices the toilet paper holder on the shelf of the

[3]In fact, the sketchy specs are related with the visual images of the options considered so far (telescope, camping cup and slinky in our example). Priming processes recognize some of these options when the visual recognizer is scanning the environment (sometimes spuriously). Indirectly, the associated sketchy specs are remembered.

store. The TPH matches the collapsible tube specification because it is cylindrical and a rigid tube. This is enough of a partial match to the recorded sketchy spec that it asks inference procedures whether the TPH also has variable length. A simple scan of the full representation of a TPH (i.e., the one in LTM that includes behavioral and structural information) supplies a positive answer (we know that a TPH has to be compressed in order to be assembled to provide support).

When a subproblem becomes unblocked due to new information becoming available, the goal scheduler unblocks the suspended problem and asks reasoning mechanisms to proceed in reasoning about it. This is what happens with the launching device problem.

Status and Open Issues

The working memory model discussed here is implemented as part of the IMPROVISER system (Wills & Kolodner 1994a, 1994b). Our original intent was to extend IMPROVISER to allow it to handle and maintain multiple pending problem contexts. However, we soon realized that handling multiple problem contexts was a memory problem and that the mechanism that would allow that could also be used to explain at least some cases of opportunity recognition. We suspect that this approach will also provide us with ways of explaining forgetting during a long reasoning session and the "freshness" that reasoners feel when coming back to a problem after letting it rest for several hours or days. But more exploration is needed before we have good explanation for either of these phenomena. Indeed, we don't yet have a full explanation of the constraints on memory in handling multiple contexts and in maintaining control of the active goals involved in opportunistic recognition. We do believe, however, that we have proposed a framework within which these questions can be answered quite nicely. We look forward both to continued computational modeling and continued experimentation on people to answer these questions.

Acknowledgements

This research was was funded in part by NSF Grant No. IRI-8921256 and in part by ONR Grant No. N00014-92-J-1234. Special thanks to Linda Wills for helpful discussions about the IMPROVISER system. We thank Ashwin Ram, Hari Narayan, Linda Wills and our anonymous reviewers for their comments on this research.

References

Barsalou, L. (1992). *Cognitive Psychology.* Lawrence Erlbaum Associates.

Birnbaum, L. (1986). *Integrated Processing in Planning and Understanding.* PhD thesis, Yale University.

Birnbaum, L. & Collins, G. (1984). Opportunistic Planning and Freudian Slips. *Proceedings of the Sixth Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates.

Dehn, N. (1989). *Computer Story-Writing: The role of Reconstructive and Dynamic Memory,* PhD thesis, Yale University.

Grimson, W.E.L. (1990). *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press.

Hammond, K., Converse, T., Marks, M. & Seifert, C.M. (1993). Opportunism and Learning. *Machine Learning,* 10 (pp. 279–309).

Hayes-Roth, B. & Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science* 3 (pp. 275–310).

Kolodner, J. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Lawrence Erlbaum Associates.

Kolodner, J. (1993). *Case-Based Reasoning.* Morgan Kaufmann (pp. 369–382).

Kolodner, J. & Wills, L. (1993a). Case-based Creative Design. *AAAI Spring Symposium on AI and Creativity.* Stanford, CA. March 1993.

Kolodner, J. & Wills, L. (1993b). Paying Attention to the Right Thing: Issues in Case-Based Creative Design. *AAAI Case-Based Reasoning Workshop* (pp. 19–25).

Mayer, N.R.F. (1970). *Problem Solving and Creativity: In individuals and Groups.* Brooks/Cole Publishing Company, (pp. 162–175).

Norman, D.A. & Bobrow, D.G. (1979). Descriptions: An intermediate stage in memory retrieval. *Cognitive Psychology* 11 (pp. 107–123).

Ram, A. & Hunter, L. (1992). The Use of Explicit Goals for Knowledge to Guide Inference and Learning. *Journal of Applied Intelligence,* 2(1) (pp.47–73).

Patalano, A., Seifert, C. & Hammond, K.(1993). Predictive Encoding: Planning for Opportunities. *Proceedings of the Fifteenth Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates (pp. 800–805).

Smith, S.M. & Blankenship, S.E. (1991). Incubation and the persistence of fixation in problem solving. *American Journal of Psychology,* Vol.104, No. 1 (pp. 61–87).

Williams, M. & Hollan, J. (1981). The process of retrieval from very long time memory. *Cognitive Science* 5 (pp. 87–119).

Wills, L., Kolodner, J.(1994a). Towards More Creative Case-based Design Systems. *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94).* Seattle, WA.

Wills, L., Kolodner, J.(1994b). Explaining Serendipitous Recognition in Design. *Proceedings of the Sixteenth Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates (pp. 940–945).