COGNITIVE SCIENCE Vol 22 (1) 1998, pp. 53–105 ISSN 0364-0213


# Pragmatic Action

RICHARD ALTERMAN, ROLAND ZITO-WOLF AND
TAMITHA CARPENTER

*Brandeis University*

This paper begins with a discussion of two features of the everyday task environment. First, the everyday task environment is designed, and an important part of the design is the provision of explicit information to guide the individual in the adaptation of his activity. Second, some task environments are semi-permanent. These two features of the task environment reveal some important characteristics in the psychology of the individual. When novelty occurs, expansion in the range of behavior of the individual is guided by a process of internalization of information provided in the task environment by another. Because of the semi-permanence of home task environments, there is pay-off in organizing behavior in terms of the particulars of those environments. The body of this paper examines these ideas from the perspective of FLOABN. FLOABN is a computational cognitive model of an individual acquiring skill at using household and office devices.

## 1. INTRODUCTION

Newell and Simon (1972) define the task environment as follows (p. 55):

> The term *task environment*, as we shall use it, refers to an environment coupled with a goal, problem, or task—the one for which the motivation of the subject is assumed. It is the task that defines a point of view about an environment, and that, in fact, allows an environment to be delimited. Also, throughout the book, we shall often distinguish the two aspects of the theory of problem solving as (1) demands of the task environment and (2) psychology of the subject. These shorthand expressions should never seduce the reader into thinking that as a psychologist he should be interested only in the psychology of the subject. The two aspects are in fact like figure and ground—although which is which depends on the momentary viewpoint.

What Newell and Simon stated to be true of a theory of problem solving, we take to be true of any cognitive theory of behavior. Research in cognitive science needs to study both the task environment and the psychology of the subject. There exists a mutual dependence

Direct all correspondence to: Richard Alterman, Science Department, Center for Complex Systems, Brandeis University, Waltham, MA 02254. E-mail: alterman@cs.brandeis.edu; Roland Zito-Wolf's current address: GTE Laboratories Incorporated; Integrated Intelligent Systems Department; Waltham, MA 02254.

between these two aspects of behavior. Together they co-determine a theory of behavior. Like figure and ground sometimes a cognitive theory of behavior focuses on one, sometimes on the other.

Our interest is in the everyday world. Examples of tasks which (and environments where) the individual performs in the everyday world are: riding a subway (and the subway station), navigating to the house of a friend (and the streets, signs and maps that are used), and cooking french toast (in the kitchen at home). Even a task like solving problems in a textbook on Euclidean geometry has an everyday aspect, with the proof to generate and the student's notebook and textbook as the task and environment respectively.

One feature of the everyday task environment is that, in many cases, the task environment has been designed to help individuals achieve their goals. Labels, signs, iconographs, either spoken or written instructions, and man-made physical objects all exist in the task environment by design and they have meaning (symbolic value) that provides useful information for adapting behavior. This information is social (it is provided by an actor other than the performing one) and its symbolic value has its basis in cultural history.

The second feature of the everyday task environment is that from the perspective of the individual certain task environments are semi-permanent, they have duration across tasks over periods of time. For the individual, most behavior occurs in a short list of places: his office, the kitchen in his home, the inside of his car, the public library he uses. Over time that list gradually changes, but at any given point in time it is a short and relatively stable list. Because of the semi-permanence of certain task environments, there is payoff in organizing behavior in the terms of the regularly occurring tasks and task environments.

Given these features, a worst case scenario is that the individual actor only "sees" a task environment once and has no help in achieving her task. In theory, the individual must be capable of operating in the worst case scenario, in practice the individual accomplishes one thing at a time using the easiest methods available to her—she has the luxury of acting pragmatically. The individual is pragmatic in that she organizes behavior by means of the particulars of regularly occurring task environments. The individual is pragmatic because she uses the help of others. These strategies may not be optimal for the general case, but under the conditions of the everyday task environment it will be shown that they have payoff.

### 1.1. Overview of the Paper

The remainder of the paper comes in five parts. In the first part, an overview of the FLOABN[1] Project is given. The project resulted in two dissertations (Zito-Wolf, 1993; Carpenter, forthcoming).

The domain of the FLOABN project was reasoning about the usage of everyday home and office devices, including photocopiers, dishwashers, alarm clocks, telephones, FAX machines, vending machines, et cetera. Each dissertation resulted in a different version of FLOABN. The first emphasized activity and memory; the second the usage of instructions. The results and techniques featured in this paper are derived from this project; we will draw on results from both versions of the system.

Where models like SOAR (Newell, 1990; Laird et al., 1986) emphasize search and decision making mechanisms in a higher-order problem space, the FLOABN model deals with impasses primarily by interpreting external events and objects of the situation. Where SOAR transforms internal knowledge, FLOABN works by internalizing what is already known in the world external to FLOABN. In the second part of the paper, the FLOABN model of reading written instructions is presented. Reading instructions is treated as an everyday activity. The task is to extract relevant information. The task environment has both guidance (the structure and format of the instructional text) and semi-permanence (the instructions may be used more than once).

Within artificial intelligence the model of a memory driven reasoning system originates with Schank (1982). FLOABN shares this view. In the case of FLOABN, for a given activity, what is retained in memory functions as a map that contains references to objects and events that occurred in the external world in the course of some previous activity, which will be used to interpret, and act within, future unfolding situations. The third part of this paper examines the functioning of the memory system. Here we will show how the semi-permanence of certain task environments effects the functioning of memory. The overarching theme of FLOABN's account of memory is to focus on the dependence of memory on activity in the home task environment. The description of memory function we produce will differ from that of the MOP system of Schank. Where indexing in a MOP hierarchy is based on differences between generalizations, in FLOABN, items in memory are tagged by the features of the specific task environment in which the activity unfolds. Some of these tags come from external cues that may be available in the immediate environment. Some come from the priming of the immediately proceeding actions and events.

Because memory functioning is tied to the specifics of task environments the individual has heightened abilities in semi-permanent task environments. When the individual oper-

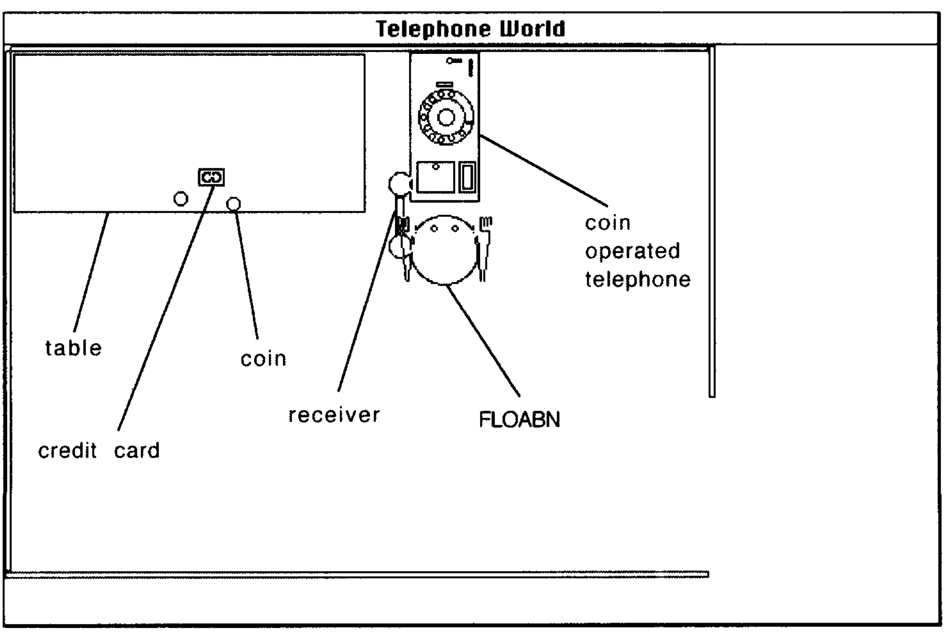



**Figure 1.** FLOABN using a coin operated phone

ates in these "islands of skill," we will refer to the actor as being "at home." The fourth part of the paper evaluates the hypothesis that incremental map-making of task environments reduces workload when the individual is "at home." We will measure both the reduction of loads for the observation and attention mechanism and the trade-off of problem-solving for memory.

The last part of the paper is a lengthy discussion section from the perspective of the literature on everyday activity and cognition.

## 2. THE FLOABN PROJECT

FLOABN is a cognitive model of an individual that acquires skill in the usage of household and office devices by acting pragmatically. Figure 1 depicts FLOABN's using a pay telephone.

### 2.1 The Everyday Task Environment

The FLOABN model assumes two features of the everyday task environment.

1. The everyday task environment provides explicit information, added to the scene of activity by another, to guide the individual in the adaptation of his behavior.
2. From the perspective of the individual, some task environments are semi-permanent; they have duration across tasks over periods of time.

These two features of the task environment reveal some important characteristics in the psychology of the individual. When novelty occurs, expansion in the range of behavior of the individual is guided by another. Because of the duration of certain "home" task environments, there is pay-off in organizing behavior in terms of the particulars of those environments.

#### 2.1.1 Help from Others

The processes by which an individual adapts her behavior to the evolution and variance in interface design for devices is guided by the direct (and indirect) instructional communication between designers and users. From the designer's point of view, building the interface is a problem of communication. From the perspective of the individual actor the task environment is a designed space (Norman, 1988); information is intentionally provided, signed, communicated, so as to guide the individual in his/her adaptive behavior. Resources for communicating to users how to use a device include written or spoken instruction, the presentation of the device, labels, diagrams, and various iconographs. As action unfolds, the user avails him/herself of the instructional information available at the scene of the activity; at each moment of the interaction such information is potentially available. An icon of a credit card conveys the "mode of payment," the design of the head piece of the telephone on the airplane affords listening, and the instructions tell the actor to "insert the credit card" before lifting the handset. Because there is no a priori analysis that can exactly predict the sequence of actions to perform, the individual must base his/her activity on the signs and instructional information provided to guide the individual's activity.

### 2.1.2 Home Task Environments

Because of the semi-permanence of certain task environments, there is payoff in organizing behavior in the terms of the regularly occurring tasks and task environments. Suppose someone is asked to cook french toast in somebody else's home. The differences between the performance of a "guest" and an "individual at home" for this kind of task illustrates the payoff of tieing behavior to the home task environment. For the guest, finding things will require the interpretation of signs available in the task environment. For example, in order for the guest to find the control for the biggest burner, she would first do a visual search to locate, on the stovetop, the relevant signs about controlling the burners and then interpret that information. For the individual at home, the situation is somewhat different; he is familiar with doing this task in this task environment. He may have internalized the location of the controller for the biggest burner, in which case he can access it directly. Even if he needs to re-interpret some of the relevant "instruction" available on the surface of the stove, his visual search is likely to be more effective, as well as the interpretation of the information he seeks. Throughout the course of activity, the bias of behavior towards task environments that have semi-permanence will also be reflected in the functioning of memory: memory for cooking french toast is tied to the features of the external world that become available during the give-and-take of activity within the individual's "kitchen at home," and not necessarily somebody else's kitchen.

## 2.2 Thinking and Behavior

FLOABN is a model of thinker as pragmatic actor. FLOABN's behavior is modeled to be tied to the specifics of its everyday task environment. The range of FLOABN's behavior is partially determined by FLOABN's personal history of activity. The tasks FLOABN performs have a tradition within a community. Adaptation of FLOABN's behavior to the demands of novel tasks and environments is mediated by the usage of instruction during activity. Interaction plays a primary role in both FLOABN's behavior and its usage of instructions.

Consider the following mix of acquisition, adaptation, maintenance, and usage of knowledge of task environment specific facts as an example of the kinds of behaviors that FLOABN is intended to replicate.

In a given actor's office, he is quite adept at calling home, or scanning the list taped to the wall above the telephone to look for the number of a friend or associate, or checking the directories on the same table as the telephone. Part of the adeptness of the actor is due to the way in which his memory is knit together with signs available in his office. It is the actor's familiarity with the requisite behavior in this particular context which guides his perceptions and selection of action. A selected set of signs available in the office are cues to aid the actor in determining the course of action. Because the "office" is a home task environment, the signs available in it facilitate the remembering of what was done in order to adapt it into "what to do;" some of the signs (e.g., numbers labeling the keys of the touchpad) will continue to work as instruction to simplify the cognitive load and make success at a given task more failsafe.

In the office next to his, the individual actor is still familiar with the kind of phone and its usage (he dials nine to get an outside line or only 4 digits to get a number within Brandeis). He would still know what the Brandeis directory looks like, and what kind of telephone number he would find there, but he would no longer know the location of the Brandeis Directory. Again, the signs facilitate the recall of relevant previous behaviors

A less familiar situation would be trying to make a telephone call from a telephone on an airplane (hereafter referred to as air telephone). Here, it is not just a matter of extending the individual's range of behavior by locating familiar types of objects. Now new procedural facts must be determined as to make it possible for the individual to determine the exact usage of the air telephone. At the scene of the activity, the task environment has been arranged so as to simplify this task for the novice user of the air telephone. The usage of the device is composed of procedures familiar to the actor (e.g., inserting a credit card or dialing a telephone number on a touch tone phone). This practice is meant to be communicated through a combination of affordances, the recognizable shapes of parts of the device (e.g., handset or touch tone pad), iconographs (that is the iconograph of a credit card on the air telephone device), and written instructions. The actor's task is to comprehend the succession of common procedures required to successfully complete a telephone call.

Some of the key characteristics of this sequence of examples are: The individual bases new behaviors on previous behaviors, including the procedural details (dialing an outside line) associated with semi-permanent task environments (Brandeis). Each of the individual's acts (in using the air telephone for the first time) is potentially informed by the input of others (instructions are accessed when needed and relevant).

### 2.3 Adaptive Planning

As a system, FLOABN begins with the simplest possible procedure for using a device, one it could have been given. Over time, it expands that procedure, its range of activity, by learning new details about the task and task environment.

The basic functioning of FLOABN is based on *adaptive planning* (Alterman,1988). The core ideas of adaptive planning are:

1. Adapt plans as-you-go-along rather than before the action begins.
2. During interaction expect the external world to provide information that will help the individual adapt his base-procedures.
3. Re-use old plans rather than always planning from scratch.
4. Old plans are tied to specific task environments.

The success of each of the basic strategies of the adaptive planner depends on the two critical features of the everyday task environment. Strategies 1 and 2 work because one of the conditions of the everyday world is that the everyday task environment is designed, providing instruction, and thus, each of the individual's acts is potentially informed by the input of others. Strategies 3 and 4 work because the individual tends to repeat activities in everyday task environments that are semi-permanent, and hence, future activities resemble the details of previous ones.

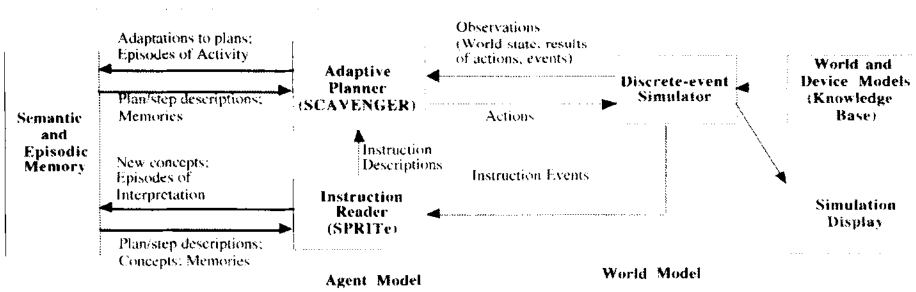



**Figure 2.** Block diagram of FLOABN (selected parts)

Each of the key features of the extended telephone-usage example described in the preceding subsection parallels the design of the adaptive planner: The individual generates new behaviors by borrowing procedural details from previous behaviors (in semi-permanent task environments). Adaptation is informed by instruction from others.

## 2.4 FLOABN: The System

A block diagram of the major parts of FLOABN is shown in Figure 2. FLOABN interacts with a world defined by a discrete-event simulator. FLOABN has two components: an adaptive planner (*SCAVENGER:* Zito-Wolf, 1993) and an instruction reader (*SPRITe:* Carpenter, 1997). The two components interact with each other in a manner that improves the overall performance of FLOABN. Adaptive planning provides clues for focusing in on only the relevant instructional materials and a context for constructing the meaning of those instructions. Instructions allow the individual to adapt its behavior by getting advice—not working alone. Both components reason from a memory of previous activities. Both components reason from a memory of previous activities; details on the memory architecture of FLOABN can be found in Appendix A.

Table 1 gives a sequence of examples of the types of problems FLOABN currently handles. With each additional scenario, FLOABN extends its range of activity by acquiring

**TABLE 1**
**A Sequence of Scenarios for FLOABN**

| Scenario | Description |
|---|---|
| 1. | DIAL PHONE call from home |
| 2. | TouchTone PHONE call from Rick's office; phone numbers on the wall |
| 3. | SODA MACHINE downstairs at Ford Hall |
| 4. | COIN OPERATED PHONE/LOCAL CALL |
| 5. | COIN OPERATED PHONE/NON-LOCAL CALL |
| 6. | COIN OPERATED PHONE/LONG-DISTANCE CALL |
| 7. | COSCI DEPT COPIER (base case) |
| 8. | LIBRARY COPIER (different positioning requirements, payments) |
| 9. | ADMINISTRATION BUILDING COPIER (different controls and payment) |
| 10. | FAX MACHINE |

new task environment specific procedural facts. In scenario 7, FLOABN first learns about the office copier: in later activities (not shown), it learns how to use additional functionalities provided by the office photocopier. Later in scenario 8, FLOABN learns about the library photocopier, which with repeated use becomes a part of FLOABN's "at work" home. Still later (scenario 10) some of this knowledge is used to support the FLOABN's FAX machine activity. In a typical run, FLOABN would normally be presented with many variants of each example type.

## 3. USING INSTRUCTION

Internalization of information explicitly provided in the scene of activity is critical to the adaptive behavior of the individual. The FLOABN model focuses on one significant source of information: written instructions.

FLOABN's model of instruction reading is novel in that it models reading as an active process. There are several properties that make it an active process:

- Using goals.
- Planning the reading activity.
- Reading in the context of a larger activity.
- Improving reading skill.

Where other models of reading and the proceduralization of text have assumed the text is read once through, before the overall activity itself has begun, FLOABN reads only when necessary, what is necessary. Rather than ignoring the overall context within which the reading occurs, the reading activity is greatly affected by the interaction of individual with the external world. Rather than reading left to right all the words of the text, the reading process is controlled using goals and plans, which are interactively adapted as a part of the reading process.

A point to emphasize is that the reading module is an adaptive planner that works pragmatically, treating the text as an everyday task environment. As the reading activity unfolds, an old reading plan is used to guide the current reading activity. It is adapted using information explicitly provided in the external world for this very purpose: the structure of the text and the formatting of the text (e.g., boldface, enumerated lists, section headings) are explicitly added to the text in order to guide the reader. Over time, if FLOABN continues to re-use the same text, it builds up a set of special purpose routines and maps for reading that text.

In the remainder of this section we first motivate the architecture of the reader. We will discuss the role of instructions in activity, discuss the kinds of information available in instructions, and motivate the FLOABN design decision to only access instructions at the point of breakdown. In the second part of this section we provide some technical details; further details can be found in Carpenter and Alterman (1997ab). Included in the discussion are a summary of the communication interface between the instruction reader and the activity planner, an overview of the activity embedded reading process, examples of reading plans and the new, text-specific, reading plans that are generated in the course of activity, and a brief description of a protocol study we did. The section concludes with a discussion of the related work on instruction usage.

### 3.1. The Role of Instruction Usage in Activity

For a given actor, the scenes of activity may be relatively fixed: his home, his office, his supermarket, his bicycle et cetera. This is one of the conditions that makes working from specific knowledge of task environment (rather than general rules of operation) advantageous for the pragmatic individual. But there is also change. Although each change in required behavior to achieve a goal may be relatively minor, the cumulative effect of these changes may be rather large. With each change that impinges on an individual's environment, the individual must be able to adjust his activity in order to continue to act effectively in the world. If there was no change, the individual would have no need to adapt. If there is change and the individual does not adapt, he will quickly become outmoded.

Consider how change occurs in FLOABN's domain. Devices are community-wide solutions to the problems of individual actors. At any given point in time there usually exists several different interface designs available for a given kind of device. Device technology and interface design vary due to economics, technology, invention, social movements, and failures in the marketplace of a given design. The development of a working device and the design of its interface are in some ways independent. (In the worst case, science develops new device technology which, because of poor interface design, is not readily usable by the community of potentially interested users.) By design, each new interface can be a composite of pieces of existing interfaces for related kinds of devices. The air telephone combines the notion of a coin-operated telephone with the usage of a credit card; the FAX machine combines the practice of a telephone with the practice of a photocopier; the VCR machine is programmed in a manner that reminds one of setting an alarm clock twice.

Because procedures for using devices are composites of already know procedures, the individual can focus on learning the details of using the photocopier in the department office with the expectation that the interface for the photocopier in the library will be culturally/historically similar. When the individual uses the photocopier at the library for the first time, the task is to relate, by way of instruction, the procedure for using the device to a history of device interfaces shared between the designer and the end-user. The instructions are the means by which the designer and user communicate with one another about the usage of the device; they are also a link between the past and future of the photocopying activity. Instruction comes in several forms: in the shapes and labels of the parts of the device, pictures and icons, and text. Access to instructional information in the first (and second, and third...)-time usage of a library photocopier is a condition of success for the individual; there is no a priori analysis that can predict the sequence of actions to perform in order to photocopy at the library.

### 3.2 Content and Structure of Instructions

The following list enumerates some of the kinds of information contained in written instructions.

1. What actions to take.
2. When to take those actions.
3. Information that helps the actor visually identify parts of the device.
4. Error correcting information.

Instructions, like any other form of text, can always be viewed as a summarization. In principle, it is impossible to describe all the conditions that need to be communicated. In practice, describing them all would make the instructions very difficult to use; the actor would have problems in locating relevant guidance, and the cost of interpretation would be dishearteningly large. Which pieces of information should be excluded from a situation description? Likely ones are those that are redundant with the expected knowledge of the actor; the instructions tell you to lift the handset, not how to place it to your ear. Other likely candidates for omission are instructions for dealing with highly unlikely situations (somebody has poured glue all over the touch pad) and features of the situation that will become apparent as the activity continues. There are also dangers in leaving out too much information: if the actor does not know he is supposed to do something, and there is no information readily available in the instruction or the encompassing scene, the actor will undoubtedly fail.

In addition to the task-relevant content of the written instruction, the instructions also include structural and organizational characteristics that are a resource for the individual in using the instructions; van Dijk and Kintsch (1983) refer to these as the "superstructures of the text." While each writer has his own individual style, van Dijk and Kintsch argue that certain types of text have "specific semantic and pragmatic constraints" which aid in the understanding of text. Take a standard technical paper. Typically the purpose of the paper is given in the abstract and then expanded upon in the introduction. The substantiation of the ideas is described in several well-labeled sections and, perhaps, subsections containing easy-to-read lists, tables, and so on. The point the author is trying to make is summarized and clarified in the conclusion. Methods for designing the text, so as to aid the reading process, also exist for instructional texts. For example, a sequence of steps is often presented as a sequentially numbered list. This and other parts of text structure are easily recognizable and contribute to the readability of the instructions.

In order to simulate the ability to take advantage of the organizational aspects of the text, FLOABN works with the instructions in a LATEX (Lamport, 1986) format.

### 3.3. Interpreting Instructions at the Point of Breakdown

In FLOABN instructional information is accessed in the context of the activity only when a breakdown in the activity occurs (see list below). Thus the actor is biased towards first re-using internalized facts before seeking instructional guidance from another individual. When a breakdown occurs, a reading plan is selected for retrieving from the instructional document the relevant facts for continuing action; reading plans exploit the structural characteristics of a document. After the relevant information is extracted and proceduralized, action continues.

Accessing instructional information in the context of activity.

1. The actor is engaged in activity.
2. A breakdown occurs.
3. The reading module uses a reading plan to locate and extract relevant facts from the instructional document.
4. Action continues.

In principle, the actor could attempt to read the instructions before action begins, and in practice, this may be helpful, but in the FLOABN model instructions are used only after a breakdown has occurred. One important reason for this is that in many cases reading instructions in their entirety is redundant with what the performing individual already knows how to do. Another reason is that instructions are intended to cover all the usages of a device. For any given activity only a subset of that information is relevant. Determining in advance of action what the relevant materials are may not be possible, but at the point of breakdown the actor has a goal and a 'problem' to guide the determination of relevance.

One resource available to simplify communication of the written instructional information is the background of common procedures shared by author and reader. A second resource is the scene of the activity. A third is that instruction reading is embedded in the unfolding interaction of the actor with the device: the arrangement of the scene, the instructions, the actor's previous knowledge, and the unfolding activity all contribute to making the actor's next action discernible.

Consider the case of the instructional manual for the FAX machine in the department office. Suppose the actor is using that FAX machine for the first time: the task is to send a fax. Much of the material in the manual is not relevant to the actor's current task; the procedures for 'initially setting up the device' are not necessarily relevant to the task of sending a fax at a given point in time. Because both the actor and the activity are embedded in the same cultural history of device usage, some of the necessary material may be redundant with what the actor already knows; instructions for using the touchpad are relevant but not needed. By trying to use the device, the actor establishes the point at which he does not know what to do next.

Even when FLOABN reads the instructions, it is thinking done by a pragmatic actor. Two utilitarian principles at work that simplify the actor's usage of the instructions are:

1. Use only **when** necessary.
2. Use only **what** is necessary.

*When* a breakdown occurs the selection of *what* instruction material to use is controlled using reading plans. Reading plans allow the individual to reason about the structure and content of the instructional text in the context of the ongoing activity. Where initially the reading plans can be general, over time, special purpose plans can be developed for using documents that are regularly accessed by the individual.

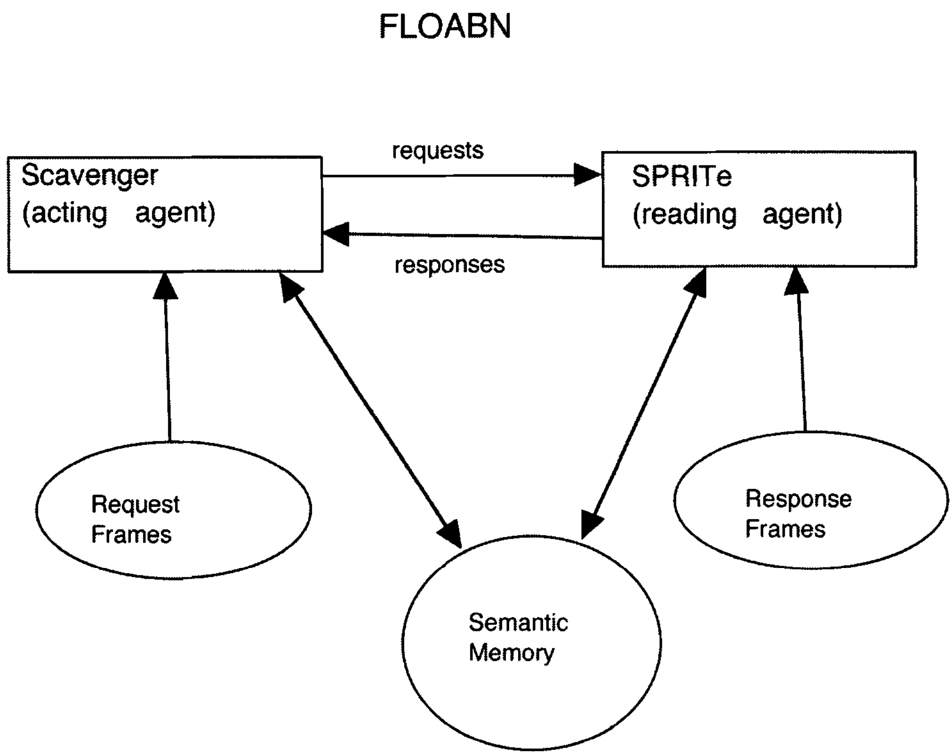



**Figure 3.** Communication between reading and action agents

### 3.4 Communication Between Activity Planner and Instruction Reader

The two main components of FLOABN, SPRITe (the reading module) and SCAVENGER (the activity planner), are independent agents (*agents:* Minsky, 1986).[2] SPRITe and SCAVENGER communicate over a specific breakdown via a defined set of request and response frames (see Figure 3); these form a short list of the kinds of information that need to be passed back and forth between the two agents within FLOABN. When the action planner requests information from SPRITe, it chooses one of about 10 request frames to fill out. Each request frame corresponds to a type of breakdown the action planner is likely to encounter.

Consider the following "Find-part-ref" request:

(request :name find-part-ref

:device fax-machine-001

:plan photocopier

:part ADF)

When this request was made, the action planner was attempting to use a FAX machine by adapting its plan for using a photocopier. When the planner encountered the unfamiliar term "ADF" (produced by an earlier access to the instructions), it asked the reader for a definition.

The reader SPRITe has a list of about 20 response frames; their types reflect the kinds of information available in instructions. Each type of request can only elicit a few of the types of response frames. For example, if the planner had made a "Find Object" request, the reader would choose one of the following:

**Object location** gives the location of object;

**Alternative object** indicates a different object can be used for the same task; and

**Missing object** indicates the object does not exist and suggests plan modifications.

The "find-part-ref" request frame has only one associated response frame: "term-definition." In order to fill out the term-definition response frame, SPRITe first reads the instructions and determines that the definition of ADF is "automatic document feeder." SPRITe then fills out the response frame as follows:

(response :name term-definition

:device fax-machine-001

:plan photocopier

:term ADF

:def automatic-document-feeder)

With this new knowledge, the action planner can use the ADF as part of its interaction with the FAX machine.

## 3.5 Overview of the Reader

SPRITe (see Figure 4) reads instructions using several methods. The overall control of SPRITe is based on the same adaptive planning model as used in FLOABN's planning agent SCAVENGER. When SPRITe first starts reading, it selects one of about 25 reading plans based on the information provided by the request frame and the observable characteristics (i.e., top-level structure) of the text. As reading takes place, the selected plan is adapted to better suit the instructions and/or reading goal of the current situation. This allows SPRITe to react to differences in organization and style between various pieces of text. This also allows the same request frame to cause different reading behavior depending on the instructions and the context.

When SPRITe has successfully read the instructions, it stores three types of information:

1. SPRITe remembers the successful reading steps that were performed as a new plan. This plan is indexed and stored in memory.
2. SPRITe remembers the characteristics of the instructions that corresponded to the successful reading steps, thus building a map of the instructions that will facilitate future usage.
3. SPRITe augments semantic memory with the knowledge it gained through reading.

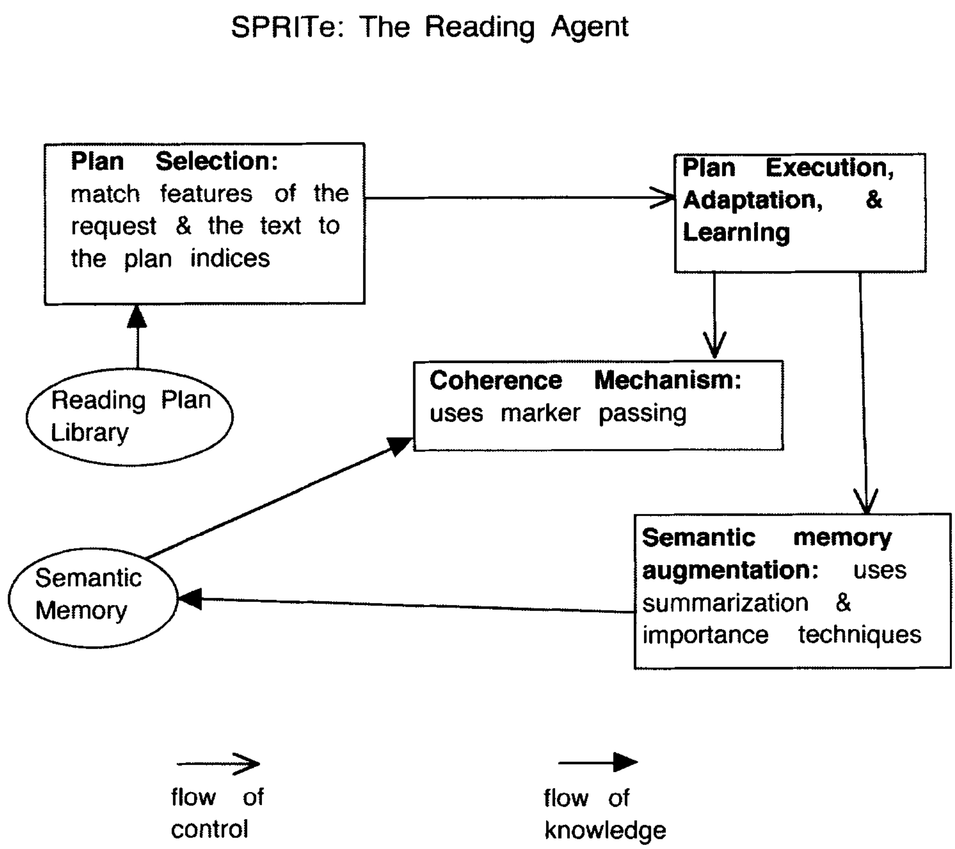



**Figure 4.** Inside the reading agent

By remembering these kinds of information, SPRITe, over time, becomes more familiar with the instruction set, incrementally converting the instructions from an unfamiliar object into one more connected and familiar to the actor. SPRITe's ability to learn in these ways increases its ability to read effectively and efficiently in the future.

During the execution of reading plans, SPRITe must 'understand' the text it is reading. Understanding is accomplished, at least in part, by establishing coherence. SPRITe establishes two types of coherence:

- textual coherence: The connection of the meaning of the text to other text being read.
- coherence with activity: The connection of the meaning of the text to the activity of the individual.

Even if the text is coherent with itself, if the connection of the text with the device and its activity cannot be established, the instructions are useless because they cannot be applied. Coherence of text with activity is not only the association of terms in the text with objects in the world (external reference), but also the association of the text with the agent's current interaction (i.e., results of plan steps taken), knowledge of how devices work, and possibilities for future action.

```
(air telephone
 ((\title (\bf (AIR TELEPHONE SYSTEM OPERATING INSTRUCTIONS)))
  (\it(The air telephone system accepts these major credit cards #\.))
  (\plain (Cain Travel Card #\, American Express #\, Carte Blanche #\,
              Diners Club International #\, Discover #\, enRoute #\,
              MasterCard #\, Visa #\.))
  (\begin (enumerate)
   (\item(Insert credit card as shown #\.)
    (Face up with card name to the right #\.))
   (\item(Lower door handle over card #\.)
    (Note #\: door will remain locked
          until handset is replaced #\.))
   (\item(Observe lighted display above for instructions #\.)
    (When instructed #\( approx #\. 10 sec #\. #\) remove phone by
          firmly grasping top of handset and pulling out #\.)
    (Return to seat to place calls #\.))
   (\item(Press green (\bf (DIAL TONE)) button and wait
              for dial tone #\.)
    (To place call #\, dial (\bf (AREA CODE))
        and (\bf (NUMBER)) #\.))
   (\item(To end call #\, press red (\bf (HANG UP)) button #\.)
    (To place additional calls repeat Step 4 #\.))
   (\item(Return handset to wall unit from which it was taken #\.)
    (Insert heel first as shown #\, then push top in firmly #\.))
   (\item(Observe lighted display above for instructions #\.)
    (Door handle will raise automatically in 10-15 seconds #\.)
    (Then remove card #\.))
   (\end(enumerate)))
  (\center
   (\large (\bf (How To Place A Call Using The Air Telephone System #\.)))))))
```

**Figure 5.** The instructions for the air telephone.

SPRITe's knowledge source is a semantic network in which basic concepts (e.g., hierarchy for types of money) are combined with detailed representations of approximately 50 devices. Coherence is established using a modification of the marker passing techniques described in Norvig (1989); see Alterman et. al (1991) and Carpenter and Alterman (1997a) for a further description of this process.

### 3.6 The Air Telephone Instructions

This section will describe in detail the process SPRITe uses to read instructions in the context of FLOABN using a simulated air telephone for the first time. Figure 5 shows the instructions for the air telephone.

To begin, FLOABN approaches the device and has the expectation that it will work like a coin operated telephone. However, when FLOABN tries to execute its plan for using a coin operated phone by lifting the receiver, it discovers it cannot lift the handset. FLOABN tries adapting its plan, but none of the steps in the coin operated phone plan are executable. So, FLOABN calls SPRITe with the following request:

(request :name next-step

:device air-telephone-001

:plan coin-operated-phone

:failed-steps

(lift-receiver

listen-for-dial-tone

insert-coin))

This request frame has four required entries:

1. the name of the request (which generally is the reading goal);
2. what device FLOABN is trying to use;
3. which action plan FLOABN is using; and
4. what plan steps have failed so far.

A fifth optional entry, the plan steps that have succeeded, is not present in this instance of the request frame.

SPRITe's algorithm is shown in Figure 6. In step 1, SPRITe notes that the request type is "next-step." In step 2, SPRITe locates the air telephone instructions (see Figure 5) and builds a representation of them, which initially contains a list of the instructions' top-level text structures: 'title,' 'italic-text,' 'plain-text,' 'enumerated-list,' and 'big-text.'

1. Identify request type.
2. If instructions for current reading task have been used before:

   **then** locate previously constructed representation
   **else** build an initial representation for the instructions

3. Select reading plan based on request frame and characteristics of the instructions.
   (a) If any of the indices of a reading plan are unmatched, anticipate that adaptation will be necessary.
4. Execute reading plan and adapt.
   (a) Sequence through steps to locate text, verify position, and extract content.
   (b) If failure occurs, adapt and either restart step 4 or continue.
   (c) If adaptation is unsuccessful restart step 3.
5. Remember successful plan steps and store in plan memory.
6. Prepare and send response frame.
7. Return control to ACTION AGENT (or to READING AGENT during a recursive call).

**Figure 6.** SPRITe's top-level control

```
(plan
:name find-steps_short-text
:goal next-step
:indices (short-text)
:plan
  (:locate
     (assoc
        '\begin(enumerate) text)
   :verify
     (if (not (next-type location
                          '\item))
         (skim (first location)
               'find-steps))
   :extract (read-enum-list location
     (getf :failed-steps request)
     (getf :successful-steps
           request))))
```

**Figure 7.** The reading plan chosen in the air telephone example

In step 3, SPRITe chooses a reading plan by matching its goal and indices to the current context; the reading goal is matched against the name of the request frame and the indices are boolean functions which evaluate either characteristics of the instructions or details from the request frame. For the current example, the plan in Figure 7 was chosen because the goal, "next-step," matches the current request frame and because the index evaluates to true (the air telephone instructions are shorter than 2 pages). Since all the indices match, no future change is anticipated.

Step 4 executes the plan. Each plan has three tasks:

**locate** a likely location for relevant text;

**verify** that the content of the selected location is going to help (optional); and

**extract** the content of the text.

The *verify* and *extract* steps require the coherence mechanism to build representations of the text at varying levels of detail, while the locate step uses SPRITe's knowledge of the structure and content of the instruction to choose the most likely portion of the text to read.

The *locate* step is responsible for finding a starting point in the instructions from which the rest of the plan can work. The locate step can achieved by one of two methods: use the representation that SPRITe has built of the instructions or use the utility function listed in the :locate portion of the plan. In the current example, SPRITe has not developed its representation of the instructions yet, so the :locate method of the "find-steps-short-text" plan is used, which locates the enumerated list in the air telephone instructions. If no enumerated list had been found, the plan would fail, and SPRITe would either attempt to adapt the plan or choose another plan.

The *verify* step uses the coherence mechanism to determine whether the "dumb" search done by the locate step actually found an appropriate piece of text. The verify portion of the "find-steps-short-text" plan looks for a description component—a piece of text that describes the purpose or content of the list. (According to SPRITe's knowledge of instruction superstructures, the description component is an optional first line of text in the list that is not preceded by the "item" formatting command.) In the air telephone example, the enumerated list does not have a description component. This is not a failing condition for this plan. Only if a description does exist but is not coherent with the reading goal would the verify step fail. If this were the case, SPRITe would go back to the locate step and find another location.

Finally, SPRITe performs the *extract* step. The call to "read-enum-list" is actually a call to a sub-plan that uses marker passing to establish coherence between the content of the enumerated list and the plan steps provided by the request frame. This instance of "read-enum-list" skims the first item (line 5) of the enumerated list and discovers the coherence between the instruction "insert credit card" and the failed-step "insert-coin."

Step 4 also includes a provision for adaptation. In the air telephone example, the reading plan was successfully executed without a need to adapt. If adaptation had been necessary, SPRITe would have first examined any unmatched indices, as well as features of the text and the request frame that were not utilized by the indices. Using this information, SPRITe would have selected either a new locate step (possibly with a related verify step) or a new extract step from a library of location and extraction techniques. The new step would be inserted into the plan and execution would proceed.

SPRITe performs step 5 of its top level reading procedure and creates a new plan. The new plan, shown below, is a copy of the original plan with a few changes. First, the text that was used to develop this plan is remembered (line 2). Although this plan could be used with any text, it will receive preferential treatment when the air telephone instructions are used again. Second, the plan has been modified to specifically find the first step of an action plan. This is reflected in the creation of a new index (line 5) which remembers that no successful-steps were provided in the request frame. Also, the :extract step (line 7) has been shortened to skim only the first item in the enumerated list. In addition, the :verify step was dropped, since it was not successfully executed.

SPRITe then prepares the following response frame:

(response :name replace-step

:device air-telephone-001

:plan coin-operated-phone

:old-step (insert-coin)

:new-step

(insert credit-card))

Control is then returned to the action planner, which proceduralizes the "insert credit-card" step into its own plan and continues interaction with the air telephone. (For discussion of the proceduralization of instructions, see Alterman et al., 1991.)

The air telephone procedure continues to deviate from the expectations provided by the coin operated telephone action plan. After inserting the credit card, the action planner is still unable to lift the receiver. Control is given to SPRITe with the following request frame:

(request :name next-step

:device air-telephone-001

:plan coin-operated-phone

:successful-steps

(insert-credit-card)

:failed-steps (lift-receiver

listen-for-dial-tone))

SPRITe selects the plan shown in Figure 7 again, because both its goal and its index match. The text-specific plan created by the previous episode (Figure 8) is not selected, even though the :text field matches the current text, because the new index (no :successful-steps field in the request frame) is not matched.

Unlike the previous episode, SPRITe does not use the :locate step provided by the plan, since SPRITe's current representation of the air telephone instructions contains the enumerated list. The :verify step is tried again, with the same results as before. Finally, the :extract step skims the list items, recognizing that the "insert credit card" item matches the successful step "insert-credit-card."

```
(plan
1 :name find-steps_short-text-001
2 :text air-telephone-instructions
3 :goal next-step
4 :indices ((short-text text)
5   (null (getf :successful-
                 steps request)))
6 :plan
     (:locate (assoc '\begin(enumerate) text)
7     :extract (skim-item (assoc '\item
                   location)
              (getf :failed-steps request)))
```

**Figure 8.** A new reading plan created during interaction with the air telephone

SPRITe then looks at the next item in the list: "lower door handle over card." This is interpreted as a "closing" action, similar to the closing action required to play a cassette in a portable radio/cassette player. A new reading plan is created (shown in Figure 9) in the same way as the plan in Figure 8. Control is then returned to the action planner with the new information.

With this information, the action planner is finally able to lift the receiver, but is immediately thwarted when the "listen for dial tone" step fails. This time, when SPRITe receives control, it selects the text-specific reading plan shown in Figure 9. The new plan is selected over the original reading plan because it better matches the current context.

Hence, the enumerated list is selected from SPRITe's representation of the instructions, and the list is perused. SPRITe matches the first two items in the enumerated list with the successful steps given in the request frame. The third item in the instructions says to "observe lighted display." SPRITe interprets this as a reading instruction. However, the display is blank since the action planner has already lifted the receiver. So, SPRITe contin-

```
(plan
:name find-steps_short-text-002
:text air-telephone-instructions
:goal next-step
:indices ((short-text text))
:plan
  (:locate
     (or (getf :enum-list inst-rep)
         (assoc '\begin(enumerate)
                text))
   :extract
     (read-enum-list location
        (getf :failed-steps request)
        (getf :successful-steps
              request))))
```

**Figure 9.** Another reading plan created during interaction with the air telephone

ues by reading the entire instruction, which indicates that the display would have said when to lift the receiver. Since the action planner already successfully lifted the receiver, SPRITe interprets this instruction as being equivalent to the step of "lift-receiver." The next item in the list, "press green DIAL TONE button," is then interpreted as an activation step. FLOABN is now able to complete interaction with the air telephone without further help from the instructions.

For further technical details on the reading process see Carpenter and Alterman (1997a).

### 3.7 A Protocol Study

Support for the SPRITe model is provided by a protocol study. Here we summarize some of the results of that study; further details can be found in Carpenter and Alterman (1997b).

Twelve subjects, students and faculty who had never before used a fax machine, were given the machine's instructions and asked to send a fax. The instructions were presented on a computer monitor one line at a time and the instruction usage was recorded. Subjects were also asked to talk aloud while sending the fax. Their comments were recorded and matched to the instructions showing on the monitor at the time each comment was made.

In spite of the length (approx. 70 pages) and difficulty of the instructions, every subject was able to use the instructions well enough to complete the requested task. No one actually read the entire set of instructions. With one exception (a subject who read none of the instructions), the subjects all used the instructions periodically during the interaction until they found text that clarified whatever part of the task was at hand. This text was then read to whatever depth was necessary to extract enough content to continue with the activity.

Four conclusions from the protocol study influence and support the SPRITe model. First, the subjects of the protocol study used the instructions in a non-linear yet systematic fashion. The subjects generally seemed to have some goal in mind which caused the subjects to skip over portions of text, backtrack to previous sections, et cetera. For example, one protocol subject, "Sara," described one of her earliest goals as:

> I have no clue how to use a fax, so I'm just trying to get an idea of what's important.

She satisfied this goal by looking through the table of contents, finding a section called "Getting to know the Telecopier 7020," and browsing through it. In SPRITe, this behavior is performed by creating, modifying, and storing reading plans, which are indexed in part by the reading goal.

Second, the subjects of the study interleaved reading the instructions and interacting with the fax machine. Figure 10 shows where Sara has found text (line 1) which describes an action she has already taken (line 2). The next two lines in the instructions (line 3) give further details regarding that action. Then the text gives her some state information for the fax machine—what the LCD display on the fax machine's control panel should say (line 4). Sara immediately looked at the display, noted that the display matched the instructions (lines 5-8), and, feeling assured that she was proceeding correctly, went back to reading the instructions. Immediately she sees the next instruction (line 9) is an alternative possibility for the contents of the display. She notes that the alternative is not applicable (line 10), and proceeds without difficulty. In general, the subjects would familiarize themselves with the device, read some instructions, apply those instructions, verify their actions with the instructions, act some more, encounter difficulty, read more instructions, correct a mistake in their activity, and so on.

Third, the subjects did not read the instructions sequentially or in their entirety. This behavior took several different forms, even for the same subject. Sometimes, as mentioned above, the subjects would quickly browse through a section, skipping over material and only occasionally stopping to read. At other times, the subjects would go backwards to

```
1 TEXT a. Place the original(s) face down in the ADF.

       :

2 SARA    OK, I got that far.

3 TEXT b. Adjust the document guides to the width of the

         originals.

        {italics}Note: the copy will automatically be

         reduced to 80% anytime the document guides are

         open wider than 8.5 inches.

4 TEXT The display will indicate:

          DIAL OR PRESS COPY

          8:30 AM 8-13-86

5 ME   So you're checking the display now?

6 SARA    Right.

7 ME   Does it match?

8 SARA    It matches.  The numbers are different, but... I

         guess I'll allow that.

9 TEXT or:

          DIAL OR PRESS COPY

          WIDE ORIGINAL

10 SARA    Oop, that's not what it says.
```

*Note.* 'TEXT' refers to the line of text showing on the monitor during the current point of interaction and 'ME' indicates the tester prompting the subject to vocalize.

**Figure 10.** An excerpt from the protocol study illustrating the interleaving of reding and action

reread something more carefully. Still other times, the subjects would go back to the table of contents and select a completely new piece of text. All of these behaviors are easily accomplished using reading goals and plans.

Finally, and not surprisingly, the subjects of the protocol study became increasingly confident and skilled in their usage of the instructions over the course of the interaction. For example, early in the protocol when Sara was browsing through the "Getting to know the Telecopier 2070" section, she found a description of the parts of the fax machine. Some time later when she was examining the machine itself, she remembered exactly how to get back to that description. This shows that the subjects were building a representation of the instructions, not only of the content, but of the structure and layout as well.

SPRITe also builds a representation of the instructions as it proceeds with reading. This representation takes advantage of text structure, as well as certain type-setting conventions (e.g., boldface), in order to facilitate the navigation and usage of instructions. The representation is used by reading plans whenever the representation is available.

### 3.8 Related Work

One interesting contrasting point to FLOABN's model of instruction usage is Wilensky's model (1981) of planning and understanding, where the planning process includes a process that evaluates or assesses the situation. The FLOABN model focuses on a subset of the features that are "to be assessed," those that are presented to the actor in a way that intentionally communicates to the individual information relevant to the adaptation of his/her behavior. Thus, unlike Wilensky's model of an individual deciding how to retrieve the newspaper from the rain by ruminating over various possible scenarios (p. 211–213), the FLOABN model would focus on the recall and adaptation of previous solutions using information available in the task environment that intentionally communicates to the actor how to adapt (e.g., advice from a parent).

There are other models of instruction usage in the literature. Chapman (1990) develops a model of instruction under conditions where the instructor and instructee are both engaged in a situation, and the instructor "kibitzes" with the instructee during the engagement. Mannes and Kintsch (1991) present a model of instruction usage where the instructions are presented before the agent begins to interact with the environment, and consist of a goal and a description of the current state of the world. Badler et al. (1990) and Webber et al. (1995) describe animated simulations which are driven by natural language instructions. The work of Ross (1989) presents psychological studies that show the importance of memory in the usage of instructional material; LeFevre and Dixon (1986) show how instructions are more easily applied to future problems when presented as specific examples; Martin (1988) uses instruction as a basis of knowledge acquisition; and Kieras and Bovair (1986) discuss a psychological model of instruction usage for made-up devices where the subjects first read the instructions in their entirety and then proceduralize the 'understanding.'

In general, the view of instruction usage developed in this research differs from other models of instruction usage in its emphasis and assumptions. Instruction usage—and not analysis—is the primary method by which the individual achieves the adaptation of behavior. Instruction usage occurs in a culturally/historically structured environment. As a model of reading behavior, instruction usage places the reading activity in a larger context: FLOABN interactively interprets instructions in the context of its *existing* activity, reading only what is necessary, when it is necessary, navigating the text using the design and organization of the text that is provided by the author of the instructions.

## 4 MEMORY AND ACTIVITY

Planning is a method to guide activity. A first order planner is a planner that builds plans entirely from scratch. A second order planner is one that bases its activities on plans constructed previously for other activities; examples of second order planners are adaptive planners (Alterman, 1988), case-based planner (Hammond, 1990), and planners that work by means of derivational analogy (Carbonell, 1983). Second order planners work well in domains where there are regularities in the individual's behavior. Second order planner fit into the category of *case-based* (or memory-based) models of problem-solving and thinking (see Schank, 1982; also Kolodner, 1983ab).

Because of the semi-permance of certain task environments and tasks, patterns of behavior tend to repeat for FLOABN, and consequently the core of FLOABN is an adaptive planner. The idea of adaptive planning is to choose a plan that has worked in the past, but while executing the plan, be ready at any moment to adapt to the current situation. Thus, the system can use the information available in the external world (e.g., signs) and information available as a result of interaction to aid adaptation of its plans. (We have already seen that similar conditions exist for the task of reading text, and hence the reading process was modeled as an adaptive planner.)

In the previous section we emphasized the role of instruction in the adaptive process of a second order planner. In this section we will look at specifying some of the details about the practice of memory for a second order planner.

The overarching theme of the FLOABN's account of memory focuses on the dependence of memory on activity in home task environments. One prediction of our model is that memory outside of the context in which it was acquired does not work as well as memory retrieval within task environments that have semi-permance.

In the remainder of this section we will discuss the details of the FLOABN model of the practice of memory in support of activity. We begin with a general discussion of principles underlying FLOABN's model of memory, which is followed by a discussion of some of the details of the memory model. Memory for activity is represented as a set of *decision points* organized into a *multicase* (Zito-Wolf, 1993; Zito-Wolf and Alterman, 1993). Decision points are defined in relation to the particulars of re-occurring task environments. The multicase organization of decision points both collects together descriptors of related episodes and provides organization for the individual's behavior in future related activities. We discuss how the multicase is used during activity. We also describe how, given some goal, the particular details of the individual's interaction with the task environment are gradually added to the developing multicase structure. We conclude with a discussion of related work, specifically other Artificial Intelligence memory-based methods of representing procedural knowledge. Further details on FLOABN's model of memory for procedural knowledge, and a supporting formal analysis, can be found in Zito-Wolf (1993) and Zito-Wolf and Alterman (1993).

### 4.1 Principles of Memory in Support of Activity

The immediate situation can be viewed as an external memory of procedural information with different accessibility characteristics than internal memory (see Perkins, 1993; also Hutchins, 1995). Thus, the sum total of what is recallable from long term memory, and what is familiar to the individual and accessible in the task environment, represents the memory of the individual.

What long term memory retains about a given kind of everyday activity is not an analytic reduction of the individual's activities of a certain type but rather an enumeration, "maps" that merge the details of the individual's activity with specific features of the individual's home task environments. These "maps," step-by-step, reference the conventions at work for individual's normal routine occurring in its normal place of operation. Each link in the map may be determined by pragmatic factors and cannot be analytically determined. The map as a whole is acquired in a piecemeal fashion by the repeated performance of a given activity in a given task environment. The map, and its content, becomes most readily accessible as activity unfolds within a home task environment for the activity.

It is the tight integration of the individual's immediate task environment, activity, and memory of such, that leads to the effectiveness of memory.

- The task environment provides selection cues to aid the actor in determining a course of action. Some of the cues are provided by the objects and signs of the immediate task environment. Other cues result from the priming of the immediately proceeding actions and events. Because much of an individual's expertise is acquired during an activity within a specific task environment (home), the actor's knowledge will essentially be tagged, or indexed, by the external cues available in the task environment in which the skill was acquired. Thus, in future problem solving episodes within the same task environment, those same external cues may be readily available at a given step in the activity and will trigger the recall of the relevant materials at precisely the right time. Outside of the specific task environment these external cues may not be available.
- The task relevant facts already known by the individual guide the actor's perceptions and selection of action. The individual memory of activity in context, for example, contributes to the sorting of perceptual information available in a given situation. For unfamiliar task environments, given some task to accomplish, there are difficulties in locating the relevant details of the immediate situation out of the mass of perceptual information available. In the home task environment, an individual has greater access to the relevant details of the situation; certain perceptual features are made more salient while other less relevant ones essentially recede into the background.

1. The immediate context is part of the individual's memory for the activity.

   *Principle: Memory is distributed into the home task environment.*

2. Skill depends on the accumulation of relevant task environment specific procedural facts in long term memory.

   *(Principle: Long term memory for activity is enumerative.)*

3. Memory of relevant procedural facts is organized in terms of activity within the task environment.

   *(Principle: Activity organizes memory.)*

4. Reminding of relevant facts occurs as action unfolds (not in advance of action).

   *(Principle: Retrieval is piecemeal and interactive.)*

5. There is a tight integration of activity and task environment, i.e.,

   *(Principle: Memory function integrates particulars of regularly co-occurring pairs of tasks and task environments.)*

   (a) Reminding of procedural facts is cued by features of the task environment that have duration.

   (b) Features of the environment to attend to, and actions to be taken, are guided by what the individual has seen/done before in task environments with duration.

**Figure 11.** Principles underlying FLOABN's model of memory for activity

Figure 11 summarizes the principles underlying the FLOABN model of memory of procedural knowledge. The immediate context is part of the individual's memory for the activity. Long term memory retains particulars (rather than general principles) so as to customize the memory process towards the semi-permanent combinations of task and task environment. Procedural facts about a given activity are organized by the structure of the activity itself. Reminding of relevant facts occurs as action unfolds. All of these principles, in one way or another, are dependent on the fact that in the everyday task environment certain combinations of tasks and environments are semi-permanent.

### 4.2 Decision Points

A multicase for a given kind of activity is composed of a set of *decision points*. We define a decision point (DP) as any point in a routine interaction requiring selection among alternative *options*. A DP specifies an item for which a value is to be selected and the set of *currently known* options for that item. The set of options at each DP, as well as the set of DPs, are always open to extension; adaptation, observation, and instruction can each add new options or new DPs. Each time a possible variation is discovered in the execution of a routine in some situation, a DP must be added to the multicase or an existing DP expanded.

Each decision point contains the knowledge relevant to making a single decision, represented in terms of cases. Each option at a DP is associated with a set of *descriptors* of situations when that option was considered appropriate. Each descriptor represents one segment of some particular problem-solving episode; it describes that segment of experience by specifying a set of features and values available to the actor at that particular point in the interaction. The procedure becomes grounded in the individual's world because the descriptors are tied to specific features of the task environment.

Each pair of option and descriptor is called a *case*. There may be multiple cases for each option. To make a decision, the features of the current situation are matched against the descriptors stored with each option and the option associated with the best-matching descriptor is selected.

For example, a photocopying routine might have a decision point expressing the fact that the copier may need to be enabled, either by turning power on (in the case of the upstairs copier) or by inserting a copy card (downstairs copier). That decision would be expressed as follows:

```
(GO-TO-COPIER NEXT-STEP ->
    (CHECK-POWER ((LOCATION UPSTAIRS-OFFICE) (COPIER COPIER-001) ...))
    (GET-COPY-CARD ((LOCATION ROOM-FORD-1-002) (COPIER COPIER-004) ...)))
```

where the *item* being specified is the NEXT-STEP feature of the **go-to-copier** step and the *options* are CHECK-POWER and GET-COPY-CARD. The descriptor list associated with each option lists the values of roles, parameters, and any other features that are thought to characterize the circumstances under which that option is relevant. The DP above states that if FLOABN is in the computer science department office upstairs, it needs to turn the copier on; and if it is using the first-floor copier, it needs to insert a copy card. Typical features selected are the roles of the step and the parent step, the location of the action, the goal of the routine, and the names of the previous and parent step. Feature values can be object descriptors (e.g., the COPIER role might be filled by the value COPIER-004), symbolic expressions (e.g, the current GOAL might be (HAVE (COPY-OF PAGE-001)) ), or discrete or continuous values (e.g., an ORIENTATION-CHOICE of VERTICAL).

Although it is natural to focus on decision points as representing the branches (i.e., the NEXT-STEP features) in a procedure, decision points may be used to conditionalize any feature of any structure in FLOABN's memory. One such use is defining possible (example) fillers for the roles of steps, e.g., associating the COPIER role of the **go-to-copier** step with the set of copiers which the system has encountered thus far:

```
(COPIER-ROUTINE (ROLE COPIER) ->
    (COPIER-004 ((LOCATION ROOM-FORD-1-002) (DATE JAN 22, 1993) (STATE MA) ...))
    (COPIER-001 ((LOCATION UPSTAIRS-OFFICE) (DATE JAN 15, 1993) (STATE MA) ...))
    (COPIER-000 ((LOCATION STEVES-OFFICE) (DATE DEC 22, 1992) (STATE MA) ...)))
```

This DP stores the fact that FLOABN has thus far encountered three objects suitable to fill the COPIER role of the routine. Another example is in identifying appropriate paper orientations:

```
(PLACE-PAGE (ROLE ORIENTATION-CHOICE) ->
    (VERTICAL ((COPIER COPIER-004) (LOCATION ROOM-FORD-1-002)
        (LOCATION-CHOICE GLASS-SURFACE-004) ...))
    (HORIZONTAL ((COPIER COPIER-001) (LOCATION UPSTAIRS-OFFICE)
        (LOCATION-CHOICE GLASS-SURFACE-001)...))
    (VERTICAL ((COPIER COPIER-000) (LOCATION STEVES-OFFICE)
        (LOCATION-CHOICE GLASS-SURFACE-000) ...)))
```

This DP indicates that FLOABN has encountered two copiers requiring the page oriented VERTICALLY (paper long axis perpendicular to the copier long side-to-side axis) and one requiring it be oriented HORIZONTALLY (parallel to the copier long axis). For simplicity we have restricted the range of the ORIENTATION-CHOICE in our examples to these symbolic values. Nevertheless, it is possible for instructions to be encountered in future examples that extend or refine these choices.

## 4.3 Evaluating Decision Points

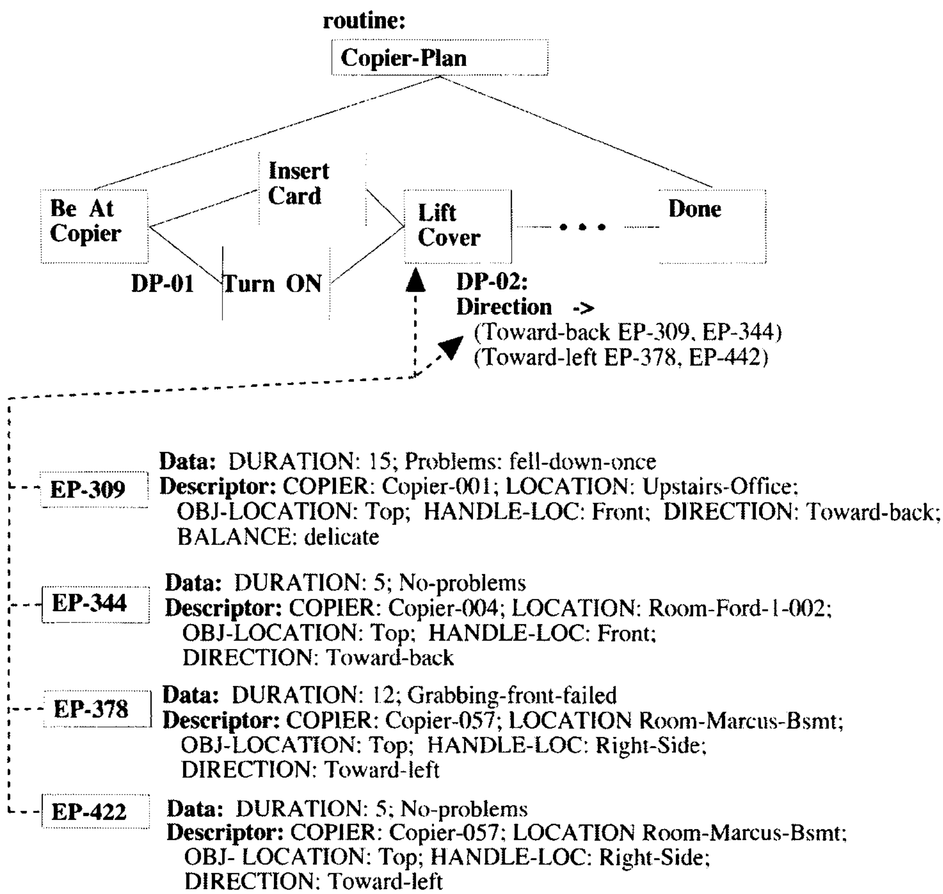
When a decision point is encountered, the alternatives are evaluated by comparing their case descriptors to the current situation; the alternative associated with the best-matched descriptor is selected. The fact that each DP lists only the options relevant at its point in the routine limits the cost of this process (Zito-Wolf and Alterman, 1993).

FLOABN selects among the options at a decision point by comparing the case descriptors for each option to the features of the current situation and selecting the option associated with the best-matching descriptor. FLOABN's matching method extends the *ratio model* (Tversky, 1977) of similarity. In Tversky's model, the items to be matched are sets of boolean features, and the match of two feature sets is defined as the ratio of the cardinality of their intersection to (some function of) their individual sizes. This normalizes similarity values to the range 0 to 1, with 0 representing disjointness and 1 representing set equality. In FLOABN, features have *values* (bindings) associated with them, so that matching also needs to take into account the similarity of the values bound to corresponding features. The match of two feature lists $x$ and *desc* is defined as:

$$\mathbf{Match}(x, desc) = \frac{\Sigma_{(f,\, v_1) \in desc}(\max_{(f,\, v_2) \in x} \mathbf{Sim}(v_1, v_2))}{|desc|}$$

where *desc* is the list of features being looked for (the feature descriptor or pattern sought), $x$ is the target being evaluated, and **Sim** is a function that computes the similarity of two values.

The match of a target $x$ to a descriptor *desc* is the normalized sum of the similarities of the values associated in $x$ and *desc* with each feature of *desc*. The match function is asymmetric in that the match ratio is defined strictly by the number of features in *desc*; this is necessary since the $x$ being tested is in general the entire current world state accessible to the actor. Including the size of $x$ in the denominator term would destroy the normalization effect. FLOABN uses *desc* as a guide to what to observe in the current situation, that is, *desc* defines the salient features of the situation with respect to a specific decision. The **Sim** function computes the similarity of two values as a weighted sum of the similarities of their types and the recursive **Match** of component parts and properties.




*Note*. This diagram presents a cross-section of the multicase and associated store episodes. It shows how instances of a single step (LIFT-COVER) from several different episodes are indexed by the multicase. The idea here is that each step collects samples of its execution in many different situations. This information is used to enrich the multi case, as in DP-02, for the DIRECTION parameter of the LIFT-COVER step.

**Figure 12.** Decision points collect together descriptors of related episodes

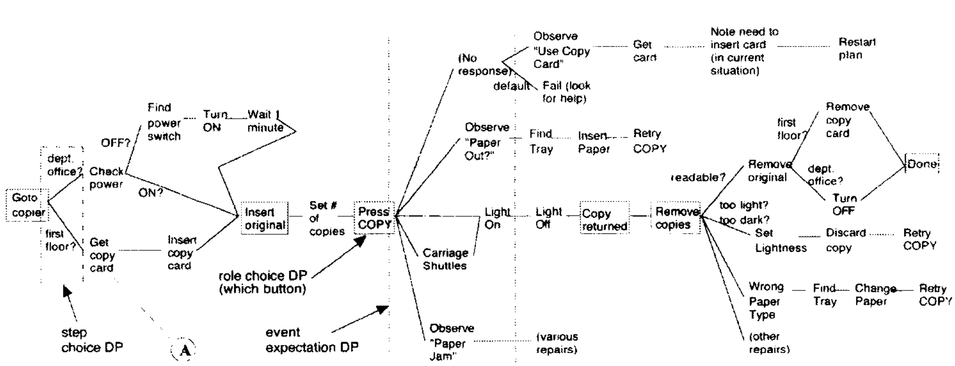
### 4.4. The Multicase

A multicase consists of a set of decision points organized into a directed graph. These decision points serve both to segment and sort the task environment specific facts acquired by the actor. Thus in Figure 12 the node having to do with lifting the cover of the photocopier has attached to it various differing episodes in the lifetime of the actor regarding lifting photocopier covers.

At a higher level, the idea is that over many experiences the copier multicase will come to approximate a generic procedure for the target task, such as shown in Figure 13. The system begins with a skeleton routine such as an actor might acquire by having the task explained or seeing it performed; the boxed steps in the figure indicate the initial routine for copying. Each additional detail arises from some specific experience. Some experiences add new paths (e.g., running out of paper), some add detail to existing paths (e.g., observing lighting and movement as copies are made), and some modify existing steps or decision criteria (e.g., learning where to look for a power switch). Most paths through the multicase access DPs contributed by several distinct episodes.

It is important to keep in mind that a diagram like Figure 13 gives a *schematic* rendering of the multicase: what the figure depicts is the overall organization of the individual's history of interactions with photocopying devices. The multicase is a way of organizing episodic memory so that it can be used as a resource for guiding future activity. By continuously accumulating and organizing the task environment specific facts that are being acquired, the multicase incrementally extends the range of situations for which the actor can effectively achieve a goal.

The partitioning of the multicase into many individual DPs makes available at each point in time those facts and actions that are believed to be relevant. That is, each decision point in the routine collects together those actions and choices that have proven useful at that point in some past execution of the routine, as shown in the DP examples above. For




*Note.* **The multicase is a way of organizing episodic memory so that it can be used as a resource for guiding future activity. This figure shows the overall organization of the individual's history of interactions with photocopying devices. While the multicase represents a routine that applies in a wide range of situations, it remains grounded in descriptors of specific episodes that contributed to it.**

**Figure 13.** A multicase organizes a history of copying episodes

example, a routine for using a change machine needs to contain knowledge about what to do if a dollar bill is rejected. The goal here is to make appropriate repairs, once identified, more readily available in future situations by linking them, in the multicase organized history, to the points of interaction where they had been useful. So when the bill is rejected, the system immediately knows to try unfolding it, or smoothing it, or brushing it off, or grabbing a different bill. This allows the multicase to become customized, for example, to the peculiarities of an individual change machine that might be part of the actor's laundromat task environment; this one particular change machine may differ significantly from other change or vending machines in its sensitivity to various factors in the preparation of bills for insertion, such as proper orientation, cleanliness, folded corners, and creases.

A related point is that one can have effective notions of what to do to repair a problem, such as bill rejection by a change machine, without necessarily understanding the reasons those actions are effective. Even for actions that are based in some theory, it is useful to

**Given:** a multicase-base and a goal

1. Select a multicase appropriate to the goal
2. Fetch the next step specified by multicase
   a. **if** no steps remain to perform, **then** return(success)
   b. **if** type(current step(s))=EVENT **then** wait for one of the events to occur
      **if** excessive time passes (based on previous experience at this step), adapt or return(event failure).
   c. **else** i. **if** multiple step alternatives exist
         **then** select one whose descriptor best matches the current situation
      ii. Project whether outcomes needed (**if** not **then** skip the step)
      iii. Check preconditions (**if** missing **then** adapt, subgoal, or fail)
      iv. Perform step
      v. Check post-conditions (either adapt, subgoal, or fail)
3. Check for unexpected events: **if** found **then**
   a. **if** adaptation limit exceeded **then** return(fail: situation too unfamiliar)
   b. **if** event is receipt of instructions relevant to current routine
      **then** interpret instructions
4. **Go to** step 2.

*Note.* This figure summarizes the process by which a multicase is used to guide activity. FLOABN's activity algorithm is based Alterman's adaptive planning (Alterman 1988).

**Figure 14.** Pseudo-code for how the multicase is used

incorporate them into our routine(s) so that we can use them without having to think about them explicitly.

In Zito-Wolf and Alterman (1992) we show this partitioning is also valuable in reducing the decision effort involved in executing a routine.

### 4.5 Using the Multicase

An adaptive planning approach to using a multicase is shown in Figure 14. FLOABN acts by selecting from the menu of options provided by the multicase at each step (or other decision) the element most appropriate to the current situation. This is done by selecting the option associated with the most similar previous situation.

Performance of a step has several parts:

**Utility Check**—A step is useful if its outcomes[3] are (a) not already asserted in the current situation, and (b) needed to achieve the goals of its parent routine. The latter is determined by looking for a chain of precondition-outcome dependencies from at least one of the outcomes of the step to the multicase's outcomes. If the step is determined not to be useful, it is skipped. This is called *projective step deletion* (cf. *projection*: Wilensky, 1983).

**Precondition Check**—The step's preconditions are checked and any missing preconditions are repaired adaptively.

**Perform Step**—This has two parts: performance of its substeps (if any), followed by invocation of its "primitive actions." The primitive actions correspond to the uninterpreted part of a step and represent the grounding of that step in low-level hand-eye motor operations. FLOABN's implementation of these operations is modeled after Firby's RAP execution-control mechanism (Firby, 1987). Steps can represent events, in which case they are treated as succeeding when their outcomes become true.

**Failure Check**—Steps can either fail explicitly or they can be judged overdue as compared to previous instances of the same step and terminated. If a step fails, FLOABN will try to adapt; if adaptation fails, it will either press on or give up, in which case the failure is propagated to the encompassing routine.[4]

**Result Check**—The step's expected outcomes are verified and repaired if necessary.

**Next Step Selection**—After a step is successfully performed, a next step is selected and the cycle repeats.

A routine begins by performing the root step of a multicase. It is complete when performance of the root step completes and its expected outcomes are valid.

### 4.6 Adding to the Multicase

The system begins with a skeleton routine such as an individual might acquire by having the task explained to it or seeing it performed. Each additional detail arises from some specific experience. Some experiences add new paths (e.g. running out of paper), some add detail to existing paths (e.g. observing lighting and movement as copies are made), and some modify existing steps or decision criteria (e.g., learning where to look for a power switch). Most paths through the multicase access elements contributed by a number of distinct experiences.

Multicases allow detail to be acquired incrementally through the overlay of old episodes with newer ones, resulting in a gradual enrichment of the multicase. As a given multicase is applied to a new situation, SCAVENGER enriches that multicase. Whenever a choice is made, the choice (if new) can either result in the addition of a new decision point or it can be added to the list of options for an existing decision point. During the test sequence we show in the empirical section of the paper, 50 new decision points are acquired and several hundred options.

This constant case acquisition has two important effects: performance requires less effort over time, and the routine becomes customized to the details of specific task environments. By "becoming a routine" we mean that one's increased familiarity with an ongoing task environment reduces certain specific costs (e.g., effort and time) of the activity; in the next section we will describe specific quantitative consequences of this process. Once the telephone-call multicase is extended to include inserting a dime in a coin operated phone, that modification does not need to be made again, nor will it be necessary to spend time receiving and interpreting messages about inserting a coin. Once the office copier is identified, it is remembered as an individual and, in the future, effort spent in identifying it is virtually eliminated.

An example of this customization to home task environments is found in the telephone domain. As time goes on task environments are differentiated by place-relevant properties. For example, we learn that the phone in an office is usually on the desk (wherever that may be) except perhaps in Roy's office, where its often fallen on the floor or buried under paper. We learn that if you need a phone number in Rick's office, you check the wall. Coin operated phones are found in halls rather than in offices. There can be no "axioms of phones" that state these facts, for they are merely regularities of the individual's experience. Nevertheless, they are reliable within the individual's "homes" and form a pragmatic basis for behavior.

### 4.7 Related Work

The memory model of FLOABN fits into the category of schema theories of memory (Bartlett, 1932; Minsky, 1975; Rummelhart, 1980). As a computational model of memory, FLOABN is an intellectual descendant of work on scripts (Schank & Abelson, 1977) and memory (*MOPs*: Schank, 1982).

Scripts (Schank & Abelson, 1977) propose that an agent's memories are organized according to specific types of situations: an eating-out script, a taxi-ride script. FLOABN's memory for activity is more specific and detailed than a script in that it retains its roots in specific episodes and situations. Where a script might describe "eating in a cafeteria" as a track in a restaurant script, FLOABN retains in memory descriptors of episodes (where pizza was bought in the student cafeteria) acquired during the course of activity within a specific locale (the Brandeis student cafeteria).

A MOP-based system conceives of memory as a hierarchy of increasingly abstract structures—Memory Organization Packets, or MOPs; the earliest example of a MOP-based memory system can be found in Kolodner (1983ab). In the context of a discussion of MOP-based systems, FLOABN has the following distinguishing features: Episodes of activity are merged into "maps" for navigating activity within task environment that are semi-permanent (home). Retrieval is tied to the features of the external world that become available during the give-and-take of activities within home task environments.

There were three early models of systems that combine memory and second order planning: *adaptive planning* (Alterman, 1988), *case-based planning* (Hammond, 1990), and *derivational analogy* (Carbonell, 1983). FLOABN is an outgrowth of the first of these. FLOABN, and adaptive planning, can be distinguished from these other models as follows: Where an adaptive planner adapts as he goes along, using information provided in the external world, Hammond's case-based planner relies on generating causal analyses (internal) either before the activity has begun or after the activity has been completed. Where the content of FLOABN's memory retains the details of previous activity within home task environment, in a derivational analogy system, memory for activity is stored using a history of the mental operations that were used to produce the "planned" activity (i.e., the plan's derivation).

Details on the comparison between FLOABN's memory and the multicase and an expanded set of artificial intelligence representation schemes for procedural knowledge can be found in Zito-Wolf (1993) and Zito-Wolf and Alterman (1993).

## 5 THE HOME TASK ENVIRONMENT

A natural by-product of FLOABN's behavior in semi-permanent task environments is that it converts what is distant and objective into the personal and subjective, making a given location in the world a 'home.'

Compare the efforts of a "guest" to an "individual at home." Suppose the guest is asked to play a Clifford Brown CD on the stereo. The actor "at home" knows more of the relevant particulars than the guest does, and his effort will be less than hers. She may be familiar with the types of devices instrumental to success at her task, but not the specifics of the devices that need to be used on this occasion. The CDs in his home are kept in stacks that are randomly ordered. There is more than one Clifford Brown CD in the stacks of CDs. The CD player in his living room has the interface associated with a particular design of a particular manufacturer of CD players. The on/off button is a "button" and not a "switch."

In any repeating task environment, the evolution of FLOABN's behavior is from that of "guest" to that of an "individual at home." The movement from "guest" to "individual at home" is a movement from "unskilled" to "skilled" at the activity within the task environment. Where the activity and task environment tend to repeat with some regularity for a given individual, the emergent integration of activity and environment is referred to as a "home task environment."

Some of the features associated with being in a home task environment are:

**Familiarity with Particulars**—In a home task environment, an individual knows the task environment specific facts that guide successful behavior for a given goal, and he knows precisely which facts are relevant.

**Focus and Perceptual Effort**—Activity within a home task environment is facilitated because the individual can selectively attend to the relevant features of the environment out of the mass of features potentially available. For non-home task environments there are difficulties in locating the relevant details of a situation.

**Retrieval Accuracy and Effort**—Because the individual and his memory are tightly integrated with their home task environments the information in memory is more readily accessible. At each step in the performance of a routine within a home task environment, reminding is facilitated by the cues readily accessible at that point in the interaction within the given task environment.

In the section on reading instructions, we saw how FLOABN gradually became more familiar with a given set of instructions. For a given text, FLOABN learns text-specific routines and gradually builds a map of the text. The section on memory showed how the multicase for a given kind of activity is tied to the specifics of the task environment (home); items in memory are tagged by information in the specific task environment as the situation of the activity unfolds. In this section we will evaluate, within the context of FLOABN, the hypothesis that the accumulation of features associated with an ongoing task environment over time reduces the information-processing load and the amount of perceptual effort on the part of the actor for future related activities within the home task environment.

## 5.1 Observation and Attention

FLOABN examines the world whenever it needs to access situation features or to identify an object of a specific type. Features are observed directly. Objects are identified by comparing candidate objects to exemplars stored in memory. A perceptual apparatus adequate to pick out objects in the environment (such as COIN-OPERATED-TELEPHONE-004 or COIN-93) is assumed, but the type or identity of objects must be determined by matching them to examples stored in memory. A candidate object is compared to a desired object/type using a similarity measure taking into account the types, features, and subparts of the objects.

Consider how FLOABN finds an object to satisfy the COPIER role of the **go-to-copier** step. It first looks in memory for a remembered object appropriate to the situation and finds COPIER-001. Call this the system's "expected" object for the given role. FLOABN then checks to see if that object exists in the current situation; this existence check causes one "object attention." If the expected object exists, the search terminates; if not, FLOABN compares all the objects marked as EXISTing in the situation to the expected object. Each comparison results in one "object attention" (See Figure 15).

A second part of the observation/attention model is the number of "facts" or "features" observed in the course of an episode. Features are observed in FLOABN in order to make decisions: the features are needed for comparison to the descriptors of cases stored in memory. For example, in deciding which device in a situation best qualifies as a COPIER, FLOABN will look at the features and parts of each candidate in the process of comparing them to stored exemplar(s). Feature observations also occur as a side-effect of object observations, when looking for an object of a specified type.

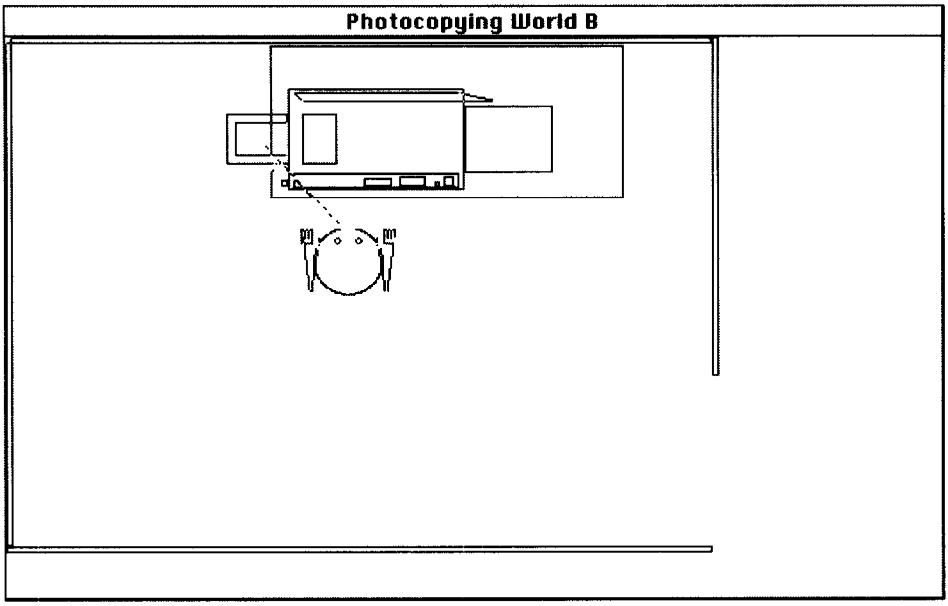



**Figure 15.** Looking for the output tray in the copier

A third aspect of observation is the number of facts committed to memory. FLOABN commits an object description or property to memory whenever it turns out to be "notable," in the sense that it turns out to be instrumental in executing the current routine. For example, in the process of locating the COPIER in a situation, FLOABN will commit to memory (a) the simple fact that each of the objects it observed EXISTs; and (b) a more detailed description of the object selected for the role, including its EXISTence, its features, and its parts list. The memorization of objects is incremental in the sense that the detailed properties of an object's parts are not committed to memory until they are determined to be instrumental for some other purpose. Once committed to memory, these properties are remembered in future episodes rather than being reobserved.

## 5.2 Quantifying Behavior

We examined the evolution of related behaviors within specific task environments over a span of episodes by presenting FLOABN with a sequence of examples including telephoning, copying, and vending machine transactions and observing the evolution of each multicase in response. FLOABN was provided initially with three skeleton multicases, one for each type of task. It was then presented with a sequence containing 15 different situations of the three types, plus 15 variant situations (one of each of the first 15 situations), plus 20 repetitions of some prior scenario (e.g., calling home from the office) for a total of 50 episodes. There were on average 25 steps per episode, yielding in excess of 1200 episodic cases. Each run of the example sequence required approximately 8 hours on an

8Mbyte Macintosh IIx under Allegro Common Lisp. For each episode we collected over 50 items of data about the evolution of the individual's activity, including such items as the size and composition of decision points, the composition of plan memory, information usage and flow, and dynamic information on the evolution of routine structure. Here we examine the change in three measures of effort involved in activity across the example sequence: the number of features observed, the amount of searching required, and overall workload. Memory usage and decision effort is discussed in more detail in Zito-Wolf and Alterman (1993).

### 5.2.1 Feature Extraction Effort

Earlier we discussed characteristics of at-home-based activity. One difference was in the number of situation features to which the actor had to attend. In non-routine situations the actor has to expend effort determining the values of the significant features of the situation and sorting through irrelevant features, having less experience to focus his search. In his home task environments, however, the actor already knows the values of many of the important features, and has a good idea of where to look for what he does not know. Figure 16 shows the number of features attended to by FLOABN during each episode. This measure counts all features—object existence, object properties, and object relations—FLOABN accessed from the situation (as opposed to from memory) in the course of activity. We expect this number to be reduced over time because as FLOABN becomes familiar with a situation, (1) relevant task environment specific features are more likely to have

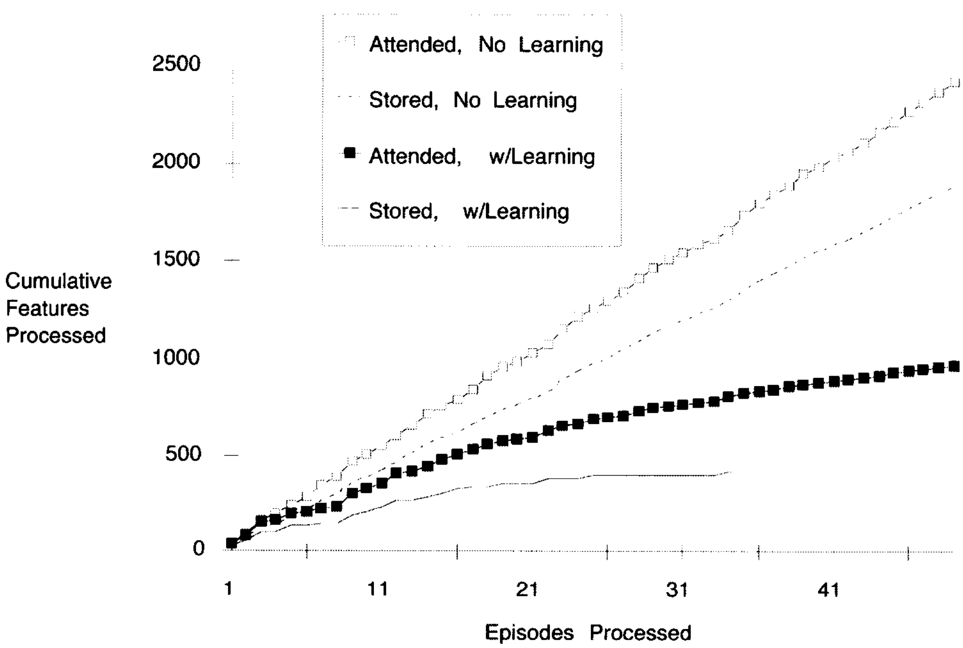



**Figure 16.** Features observed per episode

been internalized; and (2) the number of features needing to be examined at all is reduced because fewer judgments need to be made and attention is better focused. There is indeed a clear reduction in attentive effort.

### 5.2.2 Active Looking for Particular Objects

We measured the amount of active looking that FLOABN does in a given situation, that is, the number of times FLOABN had to attend to some specific object. Such attending generally occurs in the process of locating a suitable object to fill some plan role, such as COPIER, COIN-SLOT, or PHONE. This correlates somewhat with the number of features attended, since object location is typically followed by feature examination to determine how well it matches the desired object or object type. However, it differs in that object searching is a measure of how many times the actor had to select and discriminate objects rather than a count of the features extracted from a situation. An object attended in two different searches is counted twice. The familiarity afforded by at home task environments helps reduce the amount of active looking in the course of an activity (Figure 17).

### 5.2.3 Workload

To measure the overall change in effort expended in performing a task in a given situation, we created a general measure of workload that combines effort of several types. In addition to the above measures of features extracted and active looking, this measure takes into account the number of steps performed and the number of adaptations made.

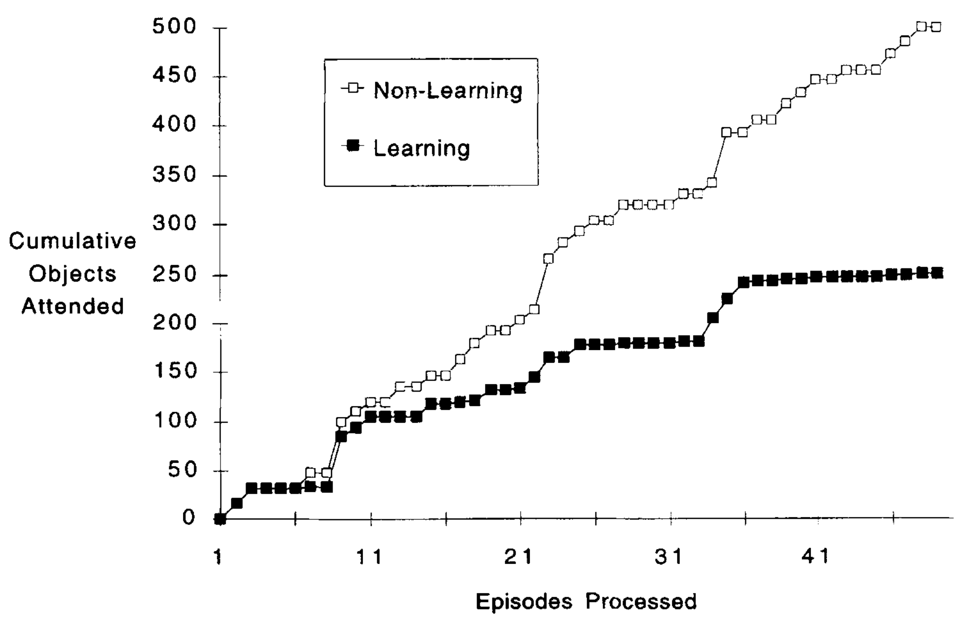



**Figure 17.** Objects examined per episode

These different components have been weighted to reflect differences in the mental effort they require:

Workload =

StepsPerformed + FeaturesAttended + 2* FeaturesStored

+5 *ActiveLooks + 10* Adaptations

Feature storage is more expensive than attending a feature because storage in FLOABN involves first attending to a feature and then either building a new DP or modifying an existing one. Active looking involves both attending and storage of features, and it also has the additional costs of visual search. Finally, adaptation typically involves a significantly larger amount of cognitive effort than the performance of a single step because, at a minimum, it involves state-space search and evaluation. We feel that this latter number is fairly conservative as some adaptations can be fairly lengthy, involving time consuming processes like experimentation and the interpretation of instructions. The results are given in Figure 18. Repetition of related behaviors within a given task environment clearly reduces the workload, and the reduced slope of the graph toward the right shows that this difference is increasing as the actor's activity become more and more tuned to the relevant task environment specific facts.

### 5.2.4 Discussion

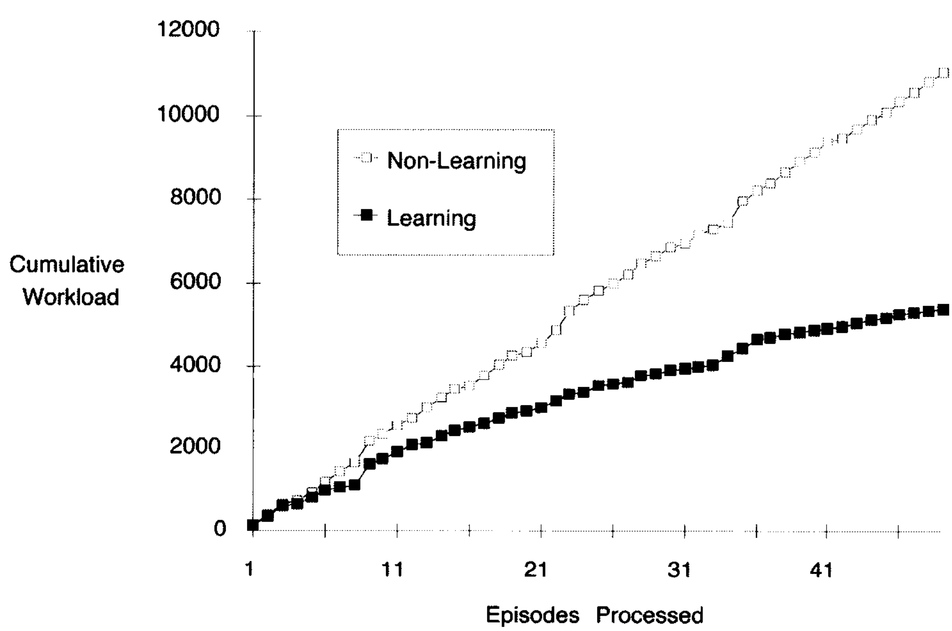



**Figure 18.** Effort required per episode

Our results suggest that the internalization of task environment specific facts and the emergence of at home task environments is effective in reducing effort in several distinct ways: active looking, features examined, memory load, and plan modification (adaptation) effort. Intuitively it is reasonable that familiarity with an activity within a given task environment reduces the effort of performing that activity. However, the study of skill acquisition in Artificial Intelligence and Cognitive Science has typically focused on the role of internal transformations of knowledge (e.g., proceduralization in ACT (Anderson, 1983) and chunking in SOAR (Laird et al., 1986)) and has largely ignored the effects of improvements in the coupling of the individual with his environment through the internalization of relevant task environment specific facts. In our view, skill acquisition is a phenomenon arising from several sources. ACT and SOAR have demonstrated that power law behavior can result from the mental transformations of knowledge such as proceduralization and chunking; Agre and Shrager (1990) have shown that power-law behavior can arise from the accumulation of local optimizations to an activity. Our work shows that similar performance improvements can be achieved through the acquisition of task environment specific facts that improve the fit between the actor and his world.

## 6 DISCUSSION

This section connects the FLOABN model of everyday activity to some more fundamental issues on the nature of behavior. There are four parts to this discussion section. The first part compares the pragmatic view of activity (FLOABN's) to a more analytic view of activity. The second part contrasts FLOABN's model of behavior to other models of interaction. The third focuses on the various ways that behavior is tied to specifics of the task environment and how each of these are reflected in FLOABN's simulated behavior. Lastly, the FLOABN model is framed from the perspective of cultural history, with the adaptation of an individual's behavior as the means by which the individual both reproduces and carries forward in time the cultural history of a given sort of activity. Although the point is not always made explicitly, throughout this discussion section, each of the particular features of FLOABN that is being discussed is traceable back to the commitments that were made at the outset of the paper regarding two features of the everyday task environment: certain task environments semi-permanent and help is provided by others to support the adaptation of an individual's behavior.

### 6.1 Pragmatic Versus Analytic View

One standard version of the everyday actor's adaptive behavior has been to view the individual as a problem-solver, part mathematician and part scientist. STRIPS (Fikes & Nilsson, 1971), as it applied to blocks world, is an example of a model of thinking and behavior that is constructed within the rational and analytic cognitive framework. STRIPS works in a neutral task environment, plans are constructed in advance of action, the plan solution always needs to be discovered, the planner always works alone without help using means-ends analysis, and plans are generated for, and proved to work in, a predicate calculus representation of the task environment.

In contrast to that vision, the FLOABN model of adaptive behavior characterizes the individual as "getting it done" with "help from others," in the "usual places," for the "daily problems" that confront him. Key to this are the two characteristics of the everyday task environment: help from others and home task environments. FLOABN was constructed assuming that it would be easier to use help than do discovery all alone and that there would be payoff in borrowing from the details of previous behaviors in home task environments.

Figure 19 contrasts idealizations of the pragmatic and analytic views of agency. The pragmatic actor is oriented towards doing and the analytic actor towards theorizing. The pragmatic actor accomplishes one particular task and the analytic actor has a theory for all tasks of the same type. For one, the performance of an activity has a history, and for the other, activities are performed without a historical basis. One assumes a solution already exists, and the other that it must be discovered. When complications arise, adaptation, for the pragmatic actor, is guided by help from another (it is mediated by instruction), while the analytic actor works alone at analysis. The pragmatic actor is directed towards creating home task environments, the analytic actor is fundamentally a guest in the world.

In our view, for the everyday world, the pragmatic actor is a more promising venue for the explanation of goal directed behavior than that of an analytic actor. Because the everyday world is conventionalized within a given community and historically rooted, it is not easily axiomatized or predicted. We may have a feel for the kinds of things that need to occur, but the actual arrangements and orderings must be pragmatically determined and cannot be derived by analysis. Because of the redundancy of everyday life, once the particulars are determined it is easier to recollect them than to re-devise them.

| Pragmatic Actor | Analytic Actor |
|---|---|
| doing | theory |
| one particular task | all tasks of the same type |
| historically based | ahistoric |
| solution exists | solution must be discovered |
| others help | individual |
| instruction | analysis |
| home task environment | guest |

**Figure 19.** Contrasting the pragmatic and analytic actors

The notion of a pragmatic action also applies to more formal kinds of tasks. An interesting comparison point here is the work of Anderson (1982) on cognitive skill acquisition as it applies to the generation of Euclidean geometry proofs. A piece of Anderson's protocol study suggests that students are reasoning about the everyday aspects of this task:

> Right off the top of my head I am going to take a guess at what I am supposed to do: \angle DCK \cong \angle ABK. There is only one of two and the side-angle-side postulates is what *they*[5] are getting to.

Our interest here is in the last sentence of this quote. Who is the "they?" Presumably it is the authors of the text. We take this as an indication that the students are reasoning in an everyday manner about the task. In other words, the student's assumptions include: the scene of the activity (the textbook) has been designed; the problem is part of a history of problem-solving episodes the student has participated in; the problem has been included to make the student practice a specific procedure; and the problem has a solution. Part of the success of the student will be attributable to the exploitation of this everyday information in order to guide adaptive behavior. Part of the success of the student will be attributable to the semi-permanence of the textbook as a task environment.

## 6.2 Interaction

FLOABN adapts its activity interactively, relying on the availability and interpretation of input from others. Both the reading and the memory component of FLOABN also depend on the interaction of the individual with the task environment. This section relates FLOABN's model of behavior to some other models of interaction.

There has been much recent interest within both Artificial Intelligence and Cognitive Science over the issues that arise from the consideration of interactionist models of activity. Suchman's work (1987) has been widely recognized for its impact on recent discussion concerning the nature of activity. Within artificial intelligence the work of Agre (1988) and Agre and Chapman (1990) has spearheaded this movement. Several sorts of arguments have been raised over the role of interaction in goal directed behavior. Some of these arguments attempted to revise the status of the term 'plan' from the notion that a plan was a 'complete' specification of an individual's actions to achieve a given set of goals.

1. The dynamic nature of the world makes the production of a complete plan infeasible for activity; plans are vague (Suchman, ibid).
2. Plans are a resource available during activity (Suchman, ibid).
3. Plans are used for communication either from one individual to another, or an individual to himself (Agre & Chapman, 1990).

These characterizations were part of a larger program to revise the role of representation and structure in mental activity, e.g.,

1. View structure not as an invariant across activity, but rather as the product of activity (Suchman, ibid; Garfinkel, 1967).
2. Structure comes into play when activities breakdown (Suchman, ibid; Winograd & Flores, 1986).

| Pragmatic Actor | Reactive Actor |
|---|---|
| instruction | perception/action loop |
| adaptation | reaction |
| others help | individual |
| historically based | by design |
| home task environment | guest |

**Figure 20.** Contrasting the pragmatic and reactive actors

3. During interaction the external world can be used to simplify and improve on mental operations that only manipulate internal representations (Lave, 1988; Kirsch & Maglio, 1994; Kirsh, 1993).

The first of these points, structure is the product of activity, could be taken to mean that structure has no role in the production of action; some might argue that for Brooks' insects (Brooks, 1991) internal representations play no role in the production of activity. In the model presented in this paper, structure has two roles: 1) it is the organization of procedural facts that makes an activity realizable in its natural setting; and 2) at the point of breakdown, it provides the basis for the interpretation of instructions.

One version of interactionist accounts has been the so-called "reactive view" of activity. Figure 20 contrasts the pragmatic and reactive view of agency. The behavior of the reactive actor is based on a perception/action loop. Both kinds of actors are interactive. The adaptation of the pragmatic actor is mediated by the usage of information intentionally added to the task environment for the purpose of guiding adaptive behavior; the reactive individual reacts to stimuli in the task environment. Success at a task, for the pragmatic individual, in part, relies on the other; the reactive actor works alone. The pragmatic actor is embedded in the history of his own activity and the traditions of his community; the reactive actor is designed for his environment and works synchronically. The pragmatic actor is directed towards creating home task environments; the reactive actor is fundamentally a guest in the world.

### 6.3 Tieing Activity to the Task Environment

FLOABN's behavior is modeled to exploit the particulars of re-occurring pairs of task and task environments. This is true for both FLOABN's overall behavior and for the reading and memory components of FLOABN. An analysis of the interdisciplinary literature on everyday activity reveals that activity is tied to the specifics of task environment in several ways, each of which is reflected in FLOABN's model of behavior.

One important theme, introduced by Vygotsky (1978), is that physical objects instrumental to the tasks you repeatedly accomplish in a given context, over time, become a part of your mental operations. Thus, the cashier does arithmetic in terms of bills and coins. The swimmer measures distance by the number of strokes to the wall. The mutual fund investor gauges his profits by the rise and fall of the Dow Jones Index. A baseball player estimates the passage of time by the runs scored and innings.

Along these lines, Scribner (1984) studied the practice of arithmetic in a milk processing plant. The key finding of this work was that the practice of arithmetic at the milk processing plant was done in terms of an instrumental physical object within the task environment, 'the milk case.' Within the plant, milk was packaged in cases of 16 quarts per case. The milk case played an important functional role for mental aspects of the pricing of delivery tickets. Rather than calculating a given order by multiplying the unit price for, say, 32 quarts of milk, the price was calculated as twice the cost of the standard 16 quart case. Thus task environment specific knowledge was used to simplify arithmetic calculation.

In the above example we see an abstract formal operation (arithmetic) being mediated by the instrumental physical objects of the task environment, the milk case. The pricers do not calculate the price of an order by multiplying single units by price, rather they convert to a milk case based method of calculation. Our argument is that what will hold for abstract mental operations involving arithmetic calculation will also hold for non arithmetic mental operations. Thus, in order to load and play a CD on the stereo in my living room, one does not resort to reasoning about the problem from the vantage point of some abstract theory about the domain, rather one reasons in the terms of the instrumental physical objects in the task environment, e.g., the on/off switch on the device, the CD case, and the tray that holds the CD in the player.

Another example from the Scribner work is the activity of preloaders. Experienced preloaders did not actually count to evaluate the size of a partial milk case, rather they seemed to be working in terms of higher level "perceptual units." The argument was that skilled workers learned the value of various configurations of containers. In this paper we have shown how this sort of effect can emerge as a function of the individual mapping out her routines for her goal directed behavior in her normal places of operation (her home). For example, we showed that over time, despite increasing complexity of task, in a given task environment the rate of the number of perceptual objects attended to accomplish a given goal was decreasing.

A related idea is recent work that extends the classical cognitive science approach to units of analysis larger than an individual (Hutchins, 1995). Variations on this idea have several names, including *person plus* (Perkins, 1993), *distributed cognition* (Cole & Engeström, 1993) and *epistemic acts* (Kirsh, 1993). In each case, the theme is to attribute cognition to the individual and his/her surround. So, cognitive science standards like representation, symbolic manipulation, and control, are distributed in varying amounts in both the mind of the individual and in the external world. The everyday task of regularly using the department photocopier admits this sort of analysis. From this perspective, FLOABN can be described as a model of an individual that evolves towards person plus. Thus, skill acquisition for FLOABN can be described as learning to redistribute the cognitive load to simplify the accomplishment of a given task in a given task environment. Early on, usage of the photocopier is mediated by instruction, over time the activity replaces instruction usage with a combination of "label interpretation" and recall.

Finally, Lave (1988: 1984) defines a notion of the *context* for an activity. Context is defined as having two parts: arena and setting. The *arena* is "the certain aspects of behavior settings that have durable and public properties." A supermarket is an arena of activity. Items sold within the supermarket arena are arranged so that an idealized shopper's physical progress through the store takes him past all the items contained in the store. The *setting* is the "personally ordered and edited version of the arena." In the supermarket only a subset of the products the store sells are considered by an individual shopper; for example, different shoppers visit different aisles in the store.

For Lave, activity is dialectically constituted in relation with setting. The shopper's personally edited version of the arena has an independent physical character with a potential realized only in relation to the shopper's activity. A shopper intends to purchase a package of enchiladas. Before she finds the enchiladas her activity is vaguely descriptive—she is looking at the enchiladas whose price keeps on going up. After she finds them, at the setting of the display of frozen Mexican dinners, the quality of her activity changes (p. 75: 1984):

> *Shopper:* Well, I don't know what brand it is. They're just enchiladas. They're put out by, I don't know. (*Discovers the display of frozen Mexican dinners.*) Here they are! (*Speaking vigorously and firmly.*) They were 65 the last time I bought them. Now they're 69. Isn't that awful?

The potential of the setting is realized in the dialectic of the individual's activity within the supermarket. Lave (p76: 1984) contends that phenomena like this, the quality difference between activity in and out of setting, is pervasive in her data, and it illustrates the tight integration of particular activities in particular settings.

This is how we interpret Lave's observation that the individual's activity within setting is enlivened: The interaction of the individual's memory of specific knowledge of the task environment and the unfolding current situation determine a set of actions which fits the current context and realizes the goal. The interaction works in both directions. It is the situation which provides selection cues to aid the actor in determining the course of action. It is the actor's knowledge of the previous acquired facts which guides his perceptions and selection of action.

### 6.4 The Cultural Historic Approach to Cognition

The generation and adaptation of new behavior is rooted in the cultural history of previous behaviors within the community. Below FLOABN is depicted as a model of an individual who reproduces and carries forward in time the cultural history of a given sort of activity. Cole and Engeström (1993) present an overview of the cultural historic view of the distribution of activity. The notion of "cultural history" emphasizes the historical aspects in the relation of individual, activity, and culturally organized environments. The individual does not work in a neutral environment, rather its design has a cultural basis. Accumulated knowledge is in part transmitted by "cultural artifacts;" tools, symbols, and texts are all cultural artifacts that can be used to mediate interaction. Cultural artifacts serve to (p. 9) "regulate interaction with one's environment and oneself." Internalization of the structure they provide for a given activity is the method by which the mental operations of an individual can transform themselves over time.

The usage of cultural history and artifacts is central to FLOABN's ability to adapt its behavior. All of the information provided in the task environment is rooted in cultural history. Standardization of part shapes, labels, and iconographs, are all cultural. The usage of each device is also rooted in cultural history. The air telephone is a combination of the practices of the telephone, the coin operated telephone, and the ATM machine. The FAX machine combines the practice of the photocopier and the telephone. Setting the VCR is like setting an alarm clock twice. The individual actor will succeed to the extent to which she has participated in the cultural history of such devices.

From the perspective of cultural history, the everyday task of using a household or office device can be described as a process of co-evolution:

1. Devices are community-wide solutions to the problems of individuals.
2. The development of a working device and the design of its interface are in someways independent.
3. At any given point in time, there usually exists several different interface designs available for a given kind of device.
4. From the designer's point of view, building the interface is a problem of communication. Resources for communicating to end users how to use a device include not only instructions, but also the presentation of the device, labels, diagrams, and various iconographs—all of these are culturally based.
5. Another resource available to a designer is that a new interface can be a composite of pieces of existing interfaces for related kinds of devices—this is also cultural historic.
6. Over extended periods of time, as technology develops, interfaces change, and designers continue to borrow from previous designs, interfaces accrue a 'cultural history'. Individuals immersed in the culture are much more likely to readily avail themselves of the usage of some device than an 'outsider.'
7. From the perspective of the end-user, during his first encounter with a interface new from his perspective, the task is to determine the usage of the device by relating it to the shared designer and end-user background. The processes by which an individual adapts to both the evolution and variance in interface design is guided by the indirect communication between designer and end-users which has its basis in culture.
8. The achievement of using a device for the first time simultaneously extends the lifetime of that design within the culture, and the range of the individual's behavior.

FLOABN is a cognitive model of an individual who reproduces and carries forward in time the cultural history of device usage.

## 7 SUMMARY REMARKS

According to Newell and Simon the task environment and the psychology of the subject are like figure and ground. They co-define one another.

In this paper, the figure has been the psychology of the individual actor in the everyday world, and the ground has been two features of the everyday task environment. First, the everyday task environment provides explicit information, added to the scene of activity by another, to guide the individual in the adaptation of his behavior. Second, some task environments are semi-permanent. These two features of the task environment reveal some important characteristics in the psychology of the individual. When novelty occurs, expansion in the range of behavior of the individual is guided by a process of internalization of information added to the task environment by another. Because of the longevity of home task environments, there is pay-off in organizing behavior in terms of the particulars of those environments.

Examples of everyday behaviors that can be characterized by these features are ubiquitous: riding a subway, making a reservation for an airplane, solving problems in a mathematics or science textbook, reading, buying groceries, planning automobile routes to and from various places in Boston, opening a door, attending a movie or a ballgame, are all examples. In each case, both features of the everyday task environment that conditioned the design of FLOABN are in force. Each of these task environments includes information that can be used to guide the novice actor is his/her activity. From the perspective of the individual, each of these tasks occur in task environments that are semi-permanent.

The body of this paper examined these features from the perspective of the FLOABN Project.

The core of FLOABN is an adaptive planner. An adaptive planner borrows details from previous behaviors and adapts behavior by exploiting information available in the task environment. The first of the strategies is appropriate because, from the perspective of the individual, certain task environments tend to have semi-permanence; the second of the strategies is appropriate because "the other exists and she wants to help."

Information provided by another to guide the individual's adaptive behavior is potentially available at any point in the interaction with a device. This information comes in several forms: instructions, labels, iconographs, affordances, standardization of parts and shapes, and the design of the device. FLOABN models in great detail the usage of one of these sources of information: textual instructions. The design of the instruction reading component of FLOABN recursed on the architecture of the overall system. Instruction reading was treated as an activity. The core of the reader was an adaptive planner that works pragmatically. As the reading activity unfolds, old reading plans are re-used, and adapted when necessary, using information explicitly provided in the external world (the text) for this very purpose. Over time, if FLOABN continues to re-use the same text, it builds up a set of special purpose routines and maps for reading the text. As a model of reading behavior, FLOABN's model of instruction usage places the reading activity in a larger context concerning the overall role of memory and comprehension in problem-solving: FLOABN interactively interprets instructions in the context of its *existing* activity, reading only what is necessary, when it is necessary, navigating the text using the design and organization of the text that is provided by the author of the instructions.

The overarching theme of the FLOABN's account of the memory system focuses on the dependence of memory on activity in context; it grounds a functional description of memory with the particulars of the task and task environment for regularly co-occurring pairs of tasks and task environments.

What FLOABN retains in long term memory about a given kind of everyday activity is not an analytic reduction of the individual's activities of a certain type, but rather an enumeration. The task of memory is to build and use "maps" that merge the details of the individual's activity with specific features of the individual's everyday task environments. These "maps," step-by-step, reference the conventions at work for the individual's normal routine occurring in its normal place of operation. The retained procedural facts about a given activity are organized by the structure of the activity itself. Reminding and storage of relevant facts is tied to the features of the external world that become available during the give-and-take of activities; reminding and storage occurs as action unfolds. All of these factors, in one way or another, depend on the assumption that in the everyday task environments certain combinations of tasks and environments are semi-permanent.

In any repeating task environment, the evolution of FLOABN's behavior is from that of "guest" to that of an "individual at home." The movement from "guest" to "individual at home" is a movement from "unskilled" to "skilled" at the activity within the task environment; it converts what is distant and objective into the personal and subjective. Where the activity and task environment tend to repeat with some regularity for a given individual, the emergent integration of activity and environment is referred to as a "home task environment". In the section on reading instructions, we showed how FLOABN gradually became more familiar (at home) with a given set of instructions. In the section on memory, we showed how memory for a given kind of activity is tied to the specifics of the task environment (home). In the section on being "at home," we measured the movement towards home both as a reduction of loads for the observation and attention mechanism and as a trade-off of problem-solving for memory.

What emerges from the sum of all these considerations is a model of everyday behavior as pragmatic action. Other models of cognition characterize thinking as analytic problem-solving in neutral task environments, and consequently have explained changes in behavior as initiated by a formal analysis on the part of the individual actor. FLOABN is a model of an individual who reproduces, adapts, and carries forward in time activities rooted in cultural history. Thinking and acting are coupled to existing, or emerging, home task environments with help from others guiding adjustments to novelty.

## APPENDIX: ASSUMPTIONS ABOUT MEMORY

The memory architecture in FLOABN has three primary functions: semantic, procedural, and motor control.

Semantic memory provides a vocabulary for encoding information. As instructions are read, the content and structure of semantic memory supports inferencing about the internal coherence of items in the document (Alterman, 1985). Semantic memory is also used to encode facts the actor acquires about a given task. Because each action is guided by the usage of facts (either from the memory of individual or taken from the instructions) about a given task, FLOABN is implicitly building a description of the activity as it unfolds (Alterman, Zito-Wolf, & Carpenter, 1991). This ongoing representation of the situation generated by the planning agent is used as the instructions are interpreted such that the instructions can be related to events that have already occurred in the current interaction.

Multicases organize and reference the set of conventions at work for the individual's normal routine unfolding in the normal place. Multicases are encoded using the vocabulary of semantic memory. Multicases organize the acquired facts so they can be used in future episodes of pragmatic action. An example of a multicase is the one which organizes facts related to withdrawing, and depositing money from the automatic teller machine at the individual's bank in Waltham.

Motor control are clusters of hand/eye motor routines. One example of a motor control is the procedure for removing a wallet from a back pocket. The blackbox implementation of motor control is loosely based on the work of Firby (1987) on RAPS (Reactive Action Packages). The interface between multicases and motor control is that maps "bottom out" into pointers to clusters of hand/eye motor routines. We have no special allegiance to the implemented model of motor control; we included it only as a convenience for modeling purposes.

For the examples we discuss in this paper, the initial locus of control begins functioning at the procedural level, at the level of multicases and procedural facts. After an action is determined from a map, FLOABN invokes a cluster of RAPS to achieve the action. When failures occur, and instructions are read, FLOABN is primarily operating at the level of semantic memory.

An issue within several communities interested in goal directed behavior is that plans do not, and cannot completely specify action. The world is dynamic and there are too many uncertainties to exactly predict what one should do next. In order to carry out a plan for an activity, an actor is going to need to be able to adapt and improvise. In the case of FLOABN some of the improvisation is explained as a function of the semantic and pragmatic, but other parts of it, like fitting a key in a lock, are left to be covered by "motor control." Because the focus of this research was elsewhere, many technical issues were left unresolved in the interface between motor control and other parts of the system.

**Acknowledgments:** The first author would like to acknowledge the following people for their comments on earlier drafts of this paper: Phil Agre, Bill Clancey, David Kirsh, Suzanne Mannes, Tom Trabasso, David Waltz, and an anonymous reviewer. Special thanks to Jim Greeno and Bill Clancey for seeing this paper through the review process.

This work was supported in part by ONR (N00014-96-1-0440). Additional funding was provided by NSF (ISI-9634102).

## NOTES

1. For Lack of a Better Name.
2. In this paper, the term *agent* is used to refer to one of the two goal-directed components of FLOABN, and the term *actor* is used to refer to the system, FLOABN, as a whole.
3. Each step in FLOABN has associated with it preconditions and outcomes. These conditions are neither necessary nor sufficient, but rather indicate the appropriateness of a given step. The preconditions and outcomes are part of the initial multicase and step definitions. When a new step is added to a multicase, for example the step of inserting a coin into a coin operated telephone, precondition and outcomes are transferred from other insert coin operations that the actor already knows about.
4. FLOABN maintains a running confidence estimate that is increased by successful steps and adaptations and decreased by problems, overdue events, and failures. Routine performance is abandoned when it falls below a set threshold.
5. We added the emphasis.

## REFERENCES


Agre, P., & Shrager, J. (1990). Routine evolution as the microgenetic basis of skill acquisition. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, (pp. 694–701), Hillsdale, NJ: Lawrence Erlbaum Associates.

Agre, P. E. (1988). The dynamic structure of everday life. Technical Report AI-TR 1085, MIT Artificial Intelligence Laboratory.

Agre, P. E., & Chapman, D. (1990). What are plans for? In P. Maes (Ed.), *Designing autonomous agents: Theory and practice from biology to engineering and back*. (pp. 17–34), Cambridge, MA: MIT Press.

Alterman, R. (1985). A dictionary based on concept coherence. *Artificial Intelligence*, 25, 153–286.

Alterman, R. (1991). Adaptive planning. *Cognitive Science*, 12, 393–421.

Alterman, R., Zito-Wolf, R., & Carpenter, T. (1991). Interaction, comprehension, and instruction usage. *Journal of the Learning Sciences*, 1(4).

Anderson, J. R. (1983). Acquisition of cognitive skill. *Psychological Review*, 89, 369–406.

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

Badler, N., Webber, B., Kalita, J., & Esakov, J. (1990). Animation from instructions. In N. Badler, B. Barsky, & D. Zeltzer (Eds.), *Making them move: Mechanics, control and animation of articulated figures*. (pp. 51–93). Los Altos, CA: Morgan Kaufmann.

Bartlett, F. (1932). *Remembering: A study in experimental and social psychology*. Cambridge, MA: Cambridge University Press.

Brroks, R. (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139–159.

Carbonell, J. (1983). Derivational analogy and its role in problem solving. In *Proceedings of the Third National Conference on Artificial Intelligence*.

Carpenter, T. (1997). *Reading to plan; planning to read*. Ph. D. thesis, Brandeis University.

Carpenter, T., & Alterman, R. (1996a). Reading to plan: Planning to read. Technical Report CS-96-188, Brandeis University.

Carpenter, T., & Alterman, R. (1996b). A protocol study of reading and activity. Technical Report CS-96-189. Brandeis University.

Chapman, D. (1990). Vision, instruction, and action. Technical Report AI-TR-1024, MIT Artificial Intelligence Laboratory.

Cole, M., & Engestrom, Y. (1993). A cultural historic approach to distributed cognition. In G. Salomon (Ed.), *Distributed cognitions*, (pp. 1–46). Cambridge, MA: Cambridge University Press.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3), 189–208.

Firby, R. J. (1987). An investigation into reactive planning in complex domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, (pp. 202–206).

Garfinkle, H. (1967). Studies in ethnomethodology. Polity. Cambridge, UK: Priority Press.

Hammond, K. J. (1990). Case-based planning: A framework for planning from experience. *Cognitive Science*, 14, 385–443.

Hutchins, E. (1995). How a cockpit remembers its speed. *Cognitive Science*, 19, 265–288.

Firby, R. J. (1987). An investigation into reactive planning in complex domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, (pp. 202–206).

Garfinkle, H. (1967). Studies in ethnomethodology. Polity. Cambridge, UK: Priority Press.

Hammond, K. J. (1990). Case-based planning: A framework for planning from experience. *Cognitive Science*, 14, 385–443.

Hutchins, E. (1995). How a cockpit remembers its speed. *Cognitive Science*, 19, 265–288.

Kieras, D. E., & Bovair, S. (1986). The acquisition of procedures from text: A production-system analysis of transfer of training. *Journal of Memory and Language*, 25, 507–524.

Kirsh, D. (1995). The intelligent use of space. *Artificial Intelligence*, 73, 31–68.

Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18(4).

Kolodner, J. (1983a). Maintaining organization in a dynamic long-term memory. *Cognitive Science*, 7, 243–280.

Kolodner, J. (1983b). Reconstructive memory: A computer model. *Cognitive Science*, 7, 281–328.

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

Lamport, L. (1986). *Latex*. Reading, MA: Addison-Wesley Publishing Company.

Lave, J. (1984). The dialectic of arithmetic in grocery shopping. In B. Rogoff & J. Lave (Eds.), *Everyday cognition*. (9–40). Cambridge, MA: Harvard University Press.

Lave, J. (1988). *Cognition in practice*. Cambridge, MA: Cambridge University Press.

LeFevre, J-A., & Dixon, P. (1986). Do Written instruction need example? *Cognition and Instruction*, 3(1), 1–30.

Mannes, S. M., & Kintsch, W. (1991). Routine computing tasks: Planning as understanding. *Cognitive Science*, 15, 305–342.

Martin, J. (1988). *A computational theory of metaphor*. Ph. D. thesis, University of California at Berkeley.

Minsky, M. (1975). A framework for representing knowledge. In P. W. Winston (Ed.), *The psychology of computer vision*. (pp. 211–277).

Minsky, M. (1986). *The society of mind*. New York, NY: Simon and Schuster.

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

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

Norman, D. A. (1988). *The psychology of everday things*. Basic Books.

Norvig. P. (1989). Marker passing as a weak method for text inferencing. *Cognitive Science*, 13(4), 569–620.

Perkins, D. N. (1993). Person-plus: A distributed view of thinking and learning. In G. Salomon (Ed.), *Distributed cognitions*, (pp. 88–110). Cambridge, MA: Cambridge University Press.

Ross, B. H. (1989). Remindings in learning and instruction. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning*. (pp. 438–469). Cambridge, MA: Cambridge University Press.

Rumelhart, D. E. (1980). Schemata: The building blocks of cognition. In R. Spiro, B. Bruce, & W. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R., & Abelson, R. (1977). Scripts, plans, goals, and understanding. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge, MA: Cambridge University Press.

Scribner, S. (1984). Studying working intelligence. In B. Rogoff & J. Lave (Eds.), *Everyday cognition*. (pp. 9–40). Cambridge, MA: Harvard University Press.

Suchman, L. A. (1987). *Plans and situated actions*. Cambridge, MA: Cambridge University Press.

Tversky, A. (1997). Features of similarity. *Psychological Review*, 84(4), 327–352.

van Dijk, T., & Kintsch, W. (1983). Strategies of discourse comprehension. New York, NY: Academic Press.

Vygotsky, L. S. (1978). Mind in society. Cambridge, MA: Harvard University Press.

Webber, B., Badler, N., Di Eugenio, B., Geib, C., Levison, L., & Moore, M. (1995). Instructions, intentions and expectations. *Artificial Intelligence*, 73, 253–269.

Wilensky, R. (1981). Meta-planning: Representing and using knowledge about planning in problem solving and natural language understanding. *Cognitive Science*, 5, 197–233.

Wilensky, R. (1983). *Planning and understanding*. Reading, MA: Addison-Wesley.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition*. Reading, MA: Addison-Wesley.

Zito-Wolf, R. (1993). *Case-based representation for procedural knowledge*. Ph. D. thesis, Brandeis University.

Zito-Wolf, R., & Alterman, R. (1992). Multicases: A case-based representation for prodedural knowledge. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. (pp. 331–336). Hillsdale, NJ: Lawrence Erlbaum Associates.

Zito-Wolf, R., & Alterman, R. (1993). A framework and an analysis of current proposals for the case-based representation for prodedural knowledge.In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. Hillsdale, NJ: Lawrence Erlbaum Associates.