Cooperating with people: The Intelligent Classroom

David Franklin
Department of Computer Science
University of Chicago
franklin@cs.uchicago.edu

Abstract

A common charge leveled against computers is that it is just too hard to figure how to get them to do what you what them to: "If computers are so smart, why can't they figure out what I want?" In building the Intelligent Classroom, we plan to demonstrate that a software system can actually do exactly this. Such a system would use cameras and microphones to sense a user's actions and then use the intentions it infers from those actions to decide what to do to best cooperate with the user. In the Intelligent Classroom, the speaker need not worry about how to operate the Classroom; he may simply go about his lecture and trust the Classroom to assist him at the appropriate moments.

1 Introduction

This paper presents a technique for incorporating plan recognition into an execution system, and describes how such a system, through its cooperative behavior, allows people to interact with it in a natural and intuitive manner. In particular, the paper describes the Intelligent Classroom and the elements of its design that allow it to cooperate with the speaker. This cooperation is first evidenced in its interaction with the speaker. The Classroom will use video cameras to observe the motion and gestures of the speaker and use directional microphones to recognize a small vocabulary of utterances. Using what it observes, the Classroom will attempt to infer what the speaker is trying to do and then, based on what it believes the speaker wants it to do, will control the setting of lights, play videos, display slides, or do whatever is appropriate. Secondly, the Classroom will produce a video of the presentation, suitable for use in support of distance learning, extending learning beyond the confines of a traditional classroom.

The Classroom is in an early stage of development. However, we have implemented most of the algorithms described in the paper and we present some preliminary results, demonstrating how the Classroom works.

1.1 Why build the Intelligent Classroom?

When a speaker has a presentation to make at another facility, he is faced with a choice: play it safe and prepare overhead transparencies, or risk all sorts of technical difficulties in trying to incorporate other media into his presentation. His dilemma is this: if he attempts to present his lecture using the media that would best convey his ideas, he risks having his entire presentation be a fiasco. He may discover that his media is incompatible with what is available at the presentation site or that controlling the equipment in this facility is a complicated and daunting task. Even if the speaker is actually able to make his presentation, it is likely that his listeners will find his use of other media to be distracting as he struggles with unfamiliar equipment. Even in the familiarity of their own facilities, speakers are often reluctant to make the extra effort to prepare presentations that utilize media any more advanced than chalkboards or overhead transparencies.

The Intelligent Classroom will encourage speakers to incorporate whatever media is most appropriate to their presentations; it will support a wide range of media types (e.g. video tapes, Powerpoint or HTML slides, the display output from a laptop computer) and allow a speaker to easily control them through natural gestures and speech. One of our primary goals in designing the Classroom is for speakers to only require a brief (no more than five minute) introduction to the facility before their presentations.

A second motivation is that the Classroom will provide speakers with an easy way of producing fair quality videos of their presentations. (The cameras will pan, tilt and zoom to best capture what is happening in the lecture.) This would allow interesting lectures to be shown on cable TV, videos of entire classes to be distributed, and lectures to be broadcast in support of distance learning – allowing people who otherwise would be unable to take a course to learn the material. Typically, videos for such purposes are produced by leaving a video camera running in the back of the lecture hall; it is far too expensive to hire a camera crew and director.

1.2 An Integrated AI domain

For the Intelligent Classroom to be successful (as outlined in the previous section) there are a few things that it must do. It must:

- use real sensors (cameras and microphones) to produce information that the plan recognizer can use.
- have an extensive library describing the plans that the speaker might attempt.
- use the results of plan recognition to determine what actions to take.

The problem of how to sense what is going on in the Classroom, while very important to the success of the project, will only be discussed briefly in this paper. Some of the more interesting issues are discussed in(Franklin & Flachsbart 1998) and a more comprehensive paper is being worked on. This paper focuses on (1) how to understand what the speaker is trying to do, and (2) how to use this understanding to produce cooperative behavior.

For the Intelligent Classroom, we will look at plan recognition as the process of inferring the hidden state of other objects, based on how they interact with the world. (The actions that a person takes often reveal what it is that they are trying to do, and how a person responds to actions that you take often suggest what his goals are.) For example, if the speaker walks up to the chalkboard and picks up a piece of chalk, it is very likely that he intends to write something on the chalkboard; if the Classroom puts up a new slide and the speaker yells "No! Go back!" it is very likely that he wanted to lecture a little longer on the previous slide.

In the next section, we describe three examples of plan recognition that will be used in the Intelligent Classroom: recognition of what the speaker is doing, recognition of when the speaker is specifically asking the Classroom to do something, and synchronization with the speaker in his presentation script. Then we describe a system that can perform these types of recognition and look at how our early implementation works on one of the examples. Finally, we look at some relevant research in Plan Recognition and Monitoring.

2 Scenarios

The following scenarios outline a few of the sorts of cooperative behavior that will be manifest in the Intelligent Classroom. They show the sorts of sequences of actions the Classroom will be able to identify and what the Classroom will do in response.

2.1 What is the speaker doing?

> The Classroom observes the speaker walk away from the podium and over to the chalkboard. As he is walking, the camera zooms out to follow him and, once he has reached the chalkboard and stopped, the Classroom adjusts the lights and sets the camera to show the portion of the chalkboard he is likely to write on.

In this example, the Classroom uses simple plan recognition to infer that the speaker is planning to write on the chalkboard. Part of the plan for writing on the chalkboard might read something like this:

1. Move towards the chalkboard
2. Stop in front of the chalkboard
3. Pick up a piece of chalk
4. Write on the chalkboard

After observing the first two steps, the Classroom infers that the speaker wants to write on the chalkboard. At all times, the Classroom considers the perceived intentions of the speaker in deciding what it should do. At the start of this example, the Classroom believes that the speaker intends to speak at the podium for a while. In response to this, it has decided to zoom in on the speaker's face to capture his facial expressions as he speaks (or perhaps, for a particularly animated speaker, zoom out a bit to show his hand gestures.) At all times, the Classroom uses what it has inferred about the speaker's intentions to determine what to do next.

> Instead of writing on the chalkboard, the speaker looks around and then moves away from the board. The Classroom turns up the lights and says: "There is some chalk in the second drawer of the podium."

In this continuation of the previous example, the Classroom notices that the speaker is not continuing in his plan for writing on the chalkboard. The Classroom knows that people will often abort a plan when a precondition for one of the plans steps is not met. In this particular situation, the only such precondition that the Classroom can help with is knowing the location of some chalk.

2.2 What is the speaker asking me to do?

> The speaker is in the midst of lecturing from a set of slides when he notices that a member of the audience has a question. The question pertains to the previous slide. The speaker points

> to the light switch, gestures upwards, and says "Let's go back to the previous slide." The Classroom brings up the lights a bit and displays the earlier slide. The camera briefly shows the slide and then pans over to the speaker as he begins to answer the question.

In this example, the speaker is essentially telling the Classroom what he would like for it to do. In the first example, the speaker was just going about lecturing and letting the Classroom figure out how to help. But in this example, the Classroom can view the speaker's actions as direct commands. The gesture where the speaker points to the light switch and moves his hands upwards is understood as a command to bring up the lights and the utterance "go back to the previous slide" is also treated as a command.

Although the speaker's actions in this example appear to be of a very different nature than those in the first example, the Classroom treats them all the same in producing cooperative behavior. The Classroom uses the speaker's actions to recognize what the speaker is trying to do (be it lecturing at the chalkboard or getting the lights brought up) and uses these inferred intentions to determine what action to take in order to be cooperative.

2.3 Where is the speaker in his script?

> The speaker is lecturing from a set of slides and notices that he is running out of time. He says "Let's skip ahead to the video." The Classroom plays the video and, when it has finished, displays the slide that would have followed the video under normal circumstances. After discussing a few more slides, the speaker says "Let's skip ahead to the conclusion." The Classroom displays the "Conclusion" slide and the speaker finishes his presentation.

A presentation script tells the Classroom what the key events (slides, videos, desired camera shots) in a presentation are and what order they will occur in. It allows the Classroom to anticipate what will happen next and can even instruct the Classroom to take particular actions without being explicitly told to by the speaker. If the speaker has provided the Classroom with a presentation script[1], the Classroom will follow along with it while the speaker gives his presentation; if the speaker deviates from the script, the Classroom detects that and follows the speaker.

In the example above, the speaker needs to speed up his lecture and so decides to skip portions of what he had intended to present. He lets the Classroom know this by what he says. The Classroom associates the phrases "the video" and "the conclusion" with particular places in the script and is therefore able to keep up with the speaker.

[1]Presentation scripts allow the speaker greater control of how the Intelligent Classroom will respond during his presentation. The Classroom will have a large library of default behaviors that will be sufficient for typical presentations. Scripts give an outline of the presentation and can tell the Classroom how to respond to particular events.

3 Design issues

In order to produce cooperative behavior, an agent must not only perform the appropriate actions, but perform them at the right time. As a result, even when the Intelligent Classroom knows what the speaker is trying to do, it still must carefully synchronize its actions with the speaker's. For example, when the speaker goes to the chalkboard to write, there are two very different camera techniques that the Classroom must use: one for when he walks and the other for when he writes. If, when the speaker was walking, the Classroom were to frame him as if he were writing, the resulting footage would appear ludicrous. Because of this need for synchronization, the Intelligent Classroom produces unique challenges that cannot be easily solved using existing plan representation. We will now describe a new way of representing plans that facilitates synchronization and show how cooperative behavior can be produced using existing plan recognition techniques.

3.1 Plan representation

In order to be clear as to what our plan representation means, we define some terms:

- An *agent* is anything (person, machine or other object) that can be viewed as causing change in the environment.
- A *process* is a sequence of actions that will be executed by a single agent. For example, a STRIPS plan for stacking blocks would be considered a process. We have developed a language for describing processes that have simultaneous actions and conditional branching.
- A *plan* is a set of processes (often to be executed by a number of different agents) that, when executed together correctly, accomplish a particular goal. For example, in the Intelligent Classroom, many plans have some processes executed by the speaker and other processes executed by the Classroom. It is important to note that this is not really a new definition of plan – any plan that has a step of the form "wait for this event to happen" is implicitly representing processes external to its main actor. This definition simply makes these external processes explicit.

Figure 1: Plan and process definitions for the Intelligent Classroom

Figure 1 shows a simple plan and two of its processes that represent a speaker's plan for going to the chalkboard and writing. The plan definition details who the actors are, what the plan accomplishes, what processes need to be executed, and how they should be synchronized. The first item under `:processes` states that a process labeled `_p1`, executed by `?lector` and named `lector-move-to-chalkboard-and-lecture` is a part of this plan. The first item under `:synchronization` indicates that step `_1` of process `_p1` needs to start at the same moment as step `_1` of process `_p2`.

The process definitions describe who the actors are (the `:main-actor` is the actor who executes the process), what steps will be executed in the running of the process, and how long different parts of the process should take to run. For the process `lector-move-to-chalkboard-and-lecture`, the first item under `:steps` states that the first step (labeled `_1`) in running this process is to achieve the goal of being at the chalkboard, and once this goal is achieved, to go on to the step labeled `_2`. For the same process, the first item under `:time-constraints` states that the process must take between 30 and 3000 seconds to run, and will typically take 300 seconds. (The "typical" values can be used to predict how long a process should take.) The other two items indicate how long the steps `_1` and `_2` must take to execute.

3.2 Process manager

At the center of the system is the process manager, which is responsible for monitoring the progress of all the processes going on in the Classroom. For example, the Classroom views the presentation script provided by the speaker as a process; as the speaker goes through his lecture, the process manager tries to synchronize the script process with what the speaker is doing.

Figure 2 shows the important structures in the process manager. At the highest level, there is a set of all active processes. In each process, the steps are treated as the states of a Finite Automaton; to monitor a process, the process manager keeps track of which step the process is in. The process manager will advance a process to a different state when it observes one of the events that the process is waiting for. For example, the process is currently in step 2 and if the process designated by `_track` signals failure, the process will go on to step 1.

When a process moves to a new step, the process manager spawns processes for each of the actions (propositions to `achieve` or processes to `do`) given in the step definition. Then, while these subprocesses run, the original process waits for any of the events given in the step definition to occur. Finally, when such an event occurs, the process manager halts all of the subprocesses and advances the process based on which event occurred.

There are two types of events that a process may wait on: a memory proposition becoming true and a process sending a particular signal. Processes send signals when they start, when they are done, and sometimes to communicate important events they have observed. In addition, the Intelligent Classroom is always performing some basic sensing to keep track of where the speaker is; this basic sensing will also signal events.

When the process manager receives a signal, it attempts to explain it through one of the processes it is monitoring (find a process that is waiting for that signal). If no explanation can be found, the system will hypothesize new processes that could explain the signal. When new processes are hypothesized, it is likely that most of them are actually incorrect[2]. So, the pro-

[2] Were the system to use a measure of likelihood, many of the hypothesized processes could be eliminated immediately. We have not yet committed to any particular technique for determining likelihood. In the short term, we will provide "reasonableness" tests for every process: memory

Figure 2: Important structures in the process manager

cess manager needs to deal with sets of processes, and be able to reject processes as future sensory data contradicts their presence. One way that a process can be rejected is if an assertion is made to memory that directly contradicts one of the assertions that is connected to it. Another cause for rejection is the violation of temporal constraints. Each step in a process has a shortest and longest completion time associated with it. Using this information, the process manager computes the interval that any given event should occur in. If the system fails to sense this event (and was trying to sense it) in the interval, a temporal constraint has been violated and the process can be removed from consideration.

When a process set is reduced to a single process, the process manager will accept that process as being a process that an agent is actually running. After it accepts a process, the process manager may ascribe certain beliefs to the agent that are suggested by his executing that process. However, if either a set of hypotheses is reduced to nothing or sensor data causes the rejection of an accepted process, then the system must acknowledge that some of the information it has ascribed to other agents is incorrect.

3.3 Monitoring other agent's processes

There may be many processes making up a given plan, but the Intelligent Classroom itself will be running only some of them. In order for the Classroom to stay synchronized with the other processes, it must monitor these as well as its own. For example, if the speaker's process is "walk to the chalkboard and write something," the Classroom will need to observe him as he walks, notice when he stops, and perhaps detect that he is actually writing.

To accomplish this, the Classroom must run observation steps that parallel the action steps in other agent's processes. The Classroom executes steps in processes where it is the main actor and executes observation steps in processes where it is not. For steps that are unobservable (for example, if the lecturer is to mentally choose which color chalk to use), it uses the step's typical completion time to determine when to try to observe the next step.

3.4 Plan recognition

Most of the work of plan recognition is accomplished by the process manager as it recognizes what processes the agents in the Intelligent Classroom are executing. In fact, by most definitions, the process manager is actually doing traditional plan recognition. However, given how we define "plan", the system must perform one additional step to recognize plans rather than processes. As mentioned previously, a plan consists of a number of processes, being executed by at least one agent. Plan recognition, given a set of active processes, is accomplished by finding a plan that explains (and is consistent with) the processes. A process is explained by a plan if the process matches one of the processes in the plan definition and is consistent if none of the synchronization or time constraints are violated.

Due to the linear nature of lectures, we hope to be able to avoid the problem of having to consider sets of plans (this would render Plan Recognition intractible.) Instead we will address two ways that we expect a speaker to go from one plan to another: through interruptions and through abandonment. With interruptions, the speaker temporarily suspends his current plan to pursue another, while with abandonment, he never resumes the old one. The Classroom will maintain a stack of suspended plans which will aid it in following along as the speaker interrupts and then resumes or abandons his plans.

3.5 Producing cooperative behavior

Most plans that will be recognized in the Intelligent Classroom will have a process that has the Classroom

---

queries that, when satisfiable, indicate that it would be reasonable for an agent to run this process in the current situation.

Figure 3: The Intelligent Classroom's Architecture

as its main actor. This means that a speaker, in executing his part of a plan, expects the Classroom to do its part of the plan: to cooperate. Cooperation, as might be expected, is achieved through being aware of what the agents around you are trying to do and then seeing how you fit into their plans. For now, because the Intelligent Classroom is purely a cooperative agent, there are only two situations where it needs to take action:

- It recognizes that the speaker is executing a plan that requires the Classroom to run some process (one of the processes in the plan has the Classroom as its main actor.)
- It recognizes that something is going wrong with a plan that it has been monitoring processes for. In these situations, the Classroom must either find a remedy for the problem or ask for help.

In both of these instances, for the Intelligent Classroom to first recognize what is going on and then to take the appropriate action, it must know what plans and processes the speaker is executing and exactly how far each of the processes has progressed. Otherwise, the Classroom will frequently perform the right actions, but at the wrong time and hinder rather than help the speaker. The synchronization declarations in the plan definitions tell the Classroom how to synchronize its actions with those of other agents.

3.6 Dealing with the physical world

The first several parts of this section describe only the high-level operation of the Intelligent Classroom. But, like any other physically embodied system, the Classroom must be able to sense what is going on around it and to physically take action. To accomplish this, the Classroom architecture consists of two components: the high-level execution system described in this paper, and a control system that links together reactive skills and vision modules to form tight control loops(Firby *et al.* 1995). We use the Gargoyle modular visual system(Prokopowicz *et al.* 1996) and have implemented Perseus' tracking algorithms(Kahn & Swain 1995)(Kahn *et al.* 1996) with it(Flachsbart 1997), allowing the Classroom to reliably locate and track the speaker's head, hands and feet.

Special "primitive" processes define how to build control loops for specific tasks, and then observe them. When necessary, these processes can modify the control loops' behavior by setting parameters or even swapping in new skills or vision modules. (Gargoyle was designed to facilitate building and adapting these sorts of dynamic control loops.) This allows the high-level system to aid the sensing system by giving it useful contextual information. For example, if the Classroom dims the lights, it can adjust the parameters of the appropriate vision modules so that the vision system does not suddenly go blind. Figure 3 shows the system architecture we use for the Intelligent Classroom. (Franklin & Flachsbart 1998) discusses the architecture in greater detail and outlines how contextual information can be used to dynamically configure the control loops.

In addition to aiding computer vision, situational context is needed to help with speech recognition. The speech recognizer can use contextual information to restrict what it listens for. Also, to prevent the Classroom from acting on speech that is simply part of the presentation, the Classroom will only "listen" when the speaker is addressing it[3].

4 Scenarios (revisited)

Having described many details of the design of the Intelligent Classroom, we will now demonstrate how the current implementation works on the first scenario. Initially, the speaker is standing to the side of the chalkboard and lecturing. The Classroom is running the process `observe-lector-stand-and-lecture`, which instructs the camera to frame the head and hands of the speaker. Figure 4a shows the input to the vision system and the Classroom's resulting representations of what it saw and what will be shown on

[3] We are considering two possible techniques that the speaker may use to indicate that he is addressing the Classroom: (1) preceding all voice commands with "Computer" or "Classroom" and (2) facing a computer monitor that will serve as the "embodiment" of the Classroom.)

Figure 4: Experiment results: (a) speaker lecturing (b) speaker walking towards the chalkboard (c) speaker writing at the chalkboard (d) partial transcript of the system

the video.

Then, the speaker picks up a piece of chalk and moves towards the chalkboard. At this point, the Classroom begins to run the process `observe-lector-move-to-chalkboard-and-lecture` (see Figure 1), which instructs the camera to zoom out and track the speaker as he walks. Figure 4b shows the speaker picking up a piece of chalk.

Finally, the speaker stops at the chalkboard and begins to write. At this point, the Classroom moves on to the second step of the observation process, which instructs the camera to frame the chalkboard and the speaker's head. Figure 4c shows what happens as the speaker writes and Figure 4d shows a transcript of the system's run. The transcript shows when processes start and stop, what events they waited for and what events actually occurred.

5 Ambiguity in plan recognition

Any system that uses the results of Plan Recognition to affect what actions it takes must deal with the problem of ambiguity: the system often needs to take action before it has determined exactly what the other agent is doing. Because the Intelligent Classroom always needs to be producing video footage, this is always the case. For example, consider what really should happen when the speaker walks towards the chalkboard. There are at least two processes that the speaker might be executing: he could be going to the chalkboard to write, or he could be going there to lecture. While he is walking, the Classroom cannot tell which of these processes he is running. But, at the same time, the Classroom must continue to film him.

In situations like this, the Classroom will have to act on what it can infer: in this example, the Classroom could only determine that the speaker simply plans to go somewhere to do something. The Classroom's process in this plan involves filming the speaker as he moves and finally stops somewhere. Then, as the Classroom narrows down the speaker's possible plans, it is able to take more specific actions. At any moment, the Intelligent Classroom takes the most specific action that it can justify.

To support this technique of further specifying processes as it is executing them, the Classroom will rely on a hierarchy of plans and processes(Kautz & Allen 1986). The processes will be defined so that, when the Classroom is executing a general process, it can switch to a more specific version of the process, starting the specific process at the point where the general process left off. For example, when the speaker begins writing on the board, the Classroom will immediately start framing him as someone writing on the chalkboard. Our representation of processes (with its labeling of steps) supports this kind of adaptive behavior.

Plan ambiguity occurs in the Classroom when it hypothesizes a set of processes for the speaker (based on an observed action). This ambiguity is resolved by repeatedly substituting a process (for which each process in the set is a specialization) for the whole set. This process will be the one that the Classroom acts on. As processes from the process set are eliminated, the Classroom is able to select increasingly specific processes to represent the set until, finally, it knows the precise process that the speaker is executing.

6 Related work

The design of the Intelligent Classroom has been influenced by the work of researchers in several AI disciplines: plan recognition, process-based execution monitoring, knowledge representation and robotics (discussed earlier). In this section we describe the research that has most strongly influenced our design.

Our process manager and how it functions bears a strong resemblance to the inner workings of PAM(Wilensky 1981). Both build representations of what is going on incrementally: first trying to explain new input based on current hypotheses of what is happening and then, only when that fails, proposing new hypotheses. And both represent what the different agents are doing and why. Our representation of plans and processes has been influenced in many ways by Schank's representation of scripts(Schank & Abelson 1977).

In deciding what to infer about agents, given the plans and processes they are involved in, the Classroom will use many of the techniques suggested by (Konolige & Pollack 1989) and (Allen & Perrault 1986).

The idea of viewing the operation of the world as the interaction of a set of concurrent processes is adapted from (Earl & Firby 1997). They have also looked at ways to learn what events should be observed (and when) through the repeated execution of processes.

The view of processes as being sequences of steps where each step has a number of conditions that it waits for (signals from other processes or memory propositions becoming true) is taken from Firby's work on RAPs(Firby 1994).

7 Conclusion

This paper outlines a technique for effectively cooperating with people and describes how it is being implemented in the Intelligent Classroom. We discuss how, in order to cooperate in the way needed in the Classroom, a system must explicitly represent the processes the speaker executes, and follow along while they execute. Further, we describe a system that will produce the cooperative behavior we desire, and show some test results from an early implementation.

In the immediate future, we will work on techniques for reducing the number of plans and processes that are being considered (through some sort of "reasonableness" measure.) We will try to get the Intelligent Classroom to make the sorts of inferences discussed by Konolige and Pollack, and Allen and Perrault. Finally, we will look at how the contextual knowledge the system maintains can be used to make the sensing tasks (computer vision and speech recognition) much easier.

Beyond the Intelligent Classroom, we hope that our techniques will prove useful in many other domains where computer-human interaction is involved: from games (where the system may attempt to thwart a player's intentions) to providing better help for desktop computer applications.

References

Allen, J. F., and Perrault, C. R. 1986. Analyzing intention in utterances. In *Readings in Natural Language Processing*. Morgan Kaufmann Publishers, Inc.

Earl, C., and Firby, R. J. 1997. Combined execution and monitoring for control of autonomous agents. In *Proceedings of the 1st International Conference on Autonomous Agents*.

Firby, R. J.; Kahn, R. E.; Prokopowicz, P. N.; and Swain, M. J. 1995. An architecture for vision and action. In *Fourteenth International Joint Conference on Artificial Intelligence*.

Firby, R. J. 1994. Task networks for controlling continuous processes. *Second International Conference on AI Planning Systems*.

Flachsbart, J. 1997. Gargoyle: Vision in the intelligent classroom. Master's thesis, University of Chicago.

Franklin, D., and Flachsbart, J. 1998. All gadget and no representation makes jack a dull environment. In *AAAI Spring Symposium on Intelligent Environments*.

Kahn, R. E., and Swain, M. J. 1995. Understanding people pointing: The perseus system. *International Symposium on Computer Vision*.

Kahn, R. E.; Swain, M. J.; Prokopowicz, P. N.; and Firby, R. J. 1996. Gesture recognition using the perseus architecture. *Computer Vision and Pattern Recognition*.

Kautz, H. A., and Allen, J. F. 1986. Generalized plan recognition. In *Proceedings of the Fourth National Conference on Artificial Intelligence*.

Konolige, K., and Pollack, M. E. 1989. Ascribing plans to agents: Preliminary report. In *Eleventh International Joint Conference on Artificial Intelligence*.

Prokopowicz, P. N.; Kahn, R. E.; Firby, R. J.; and Swain, M. J. 1996. Gargoyle: Context-sensitive active vision for mobile robots. *American Association for Artificial Intelligence*.

Schank, R., and Abelson, R. 1977. *Scripts Plans Goals and Understanding*. Lawrence Erlbaum. chapter 3.

Wilensky, R. 1981. *Inside Computer Understanding*. Lawrence Erlbaum Associates, Inc. chapter 7.