A Schema-based Approach to Cooperative Behavior[1]

Elise H. Turner
Department of Computer Science
University of New Hampshire
Durham, NH 03824
E-mail: eht@cs.unh.edu

Roy M. Turner
Department of Computer Science &
Marine Systems Engineering Laboratory
University of New Hampshire
Durham, NH 03824
E-mail: rmt@cs.unh.edu

Abstract

Agents can rely on the patterns in the world to make their problem solving more efficient. When working with others, agents can also rely on patterns - patterns for communication and group behavior. We discuss how these patterns may be captured in schemas. We present two types of schemas: *procedural schemas* which suggest a course of action for a specific situation, and *contextual schemas* which contain knowledge about specific kinds of problem solving. Both of these types of schemas affect an agent's ability to solve problems and communicate. Both types of schemas also guide the coordination of the groups working together to solve problems. In this paper, we focus particularly on the ways in which a schema-based approach can help agents to work together by integrating their individual problem solving with the constraints of coordinated behavior.

Cooperative distributed problem solving (CDPS) (Durfee *et al.*, 1989) allows artificial intelligence systems to reap the benefits of teamwork. CDPS systems can tackle problems not readily solvable by single agents, such as simultaneous monitoring of environmental conditions in a large area or control of multiple autonomous agents moving through an area and carrying out tasks. From the standpoint of cognitive science, CDPS research sheds light on communication and cooperation between intelligent agents.

For an agent to work well as a collaborator, it must be a competent individual problem solver as well as a helpful colleague. As an individual, the agent must be able to achieve goals (some its own, some related to its shared task) efficiently in a complex, changing world. This means the agent must be able to plan quickly and flexibly, tailoring its behavior to fit its situation. As a collaborator, it must communicate effectively with others and understand its role in the organization of the CDPS system. Its interactions with others must be guided by convention so that all agents' actions can be easily understood by others.

In this paper, we discuss the role of schemas in a design for integrating problem solving, communication, and cooperation. Only parts of this design have been implemented at this time. We adopt a *schema-based* approach to cooperative distributed problem solving, based on our prior work (Turner, 1990; Turner & Cullingford, 1989b; Turner, 1989a; Turner, 1989c), which addresses both of these needs. *Procedural schemas*, much like compiled plans, allow reuse of previous planning effort, either the agent's own or another agent's (e.g., a human's). *Contextual schemas* provide knowledge about kinds of problem-solving situations, which allows an agent to automatically behave appropriately in its current context. Procedural and contextual schemas guide an individual's autonomous behavior as well as its behavior within a group.

Figure 1: AUV with goal to pick up an object.

An Example of Schema-based Cooperative Behavior

We have chosen the domain of autonomous underwater vehicles (AUVs) for this research. Ultimately, AUVs will replace humans for dangerous or expensive tasks. Many of these tasks either benefit from or require more than one AUV, for example: underwater photography, in which a long baseline is needed between camera and light (Jaffe, 1988); and large-scale environmental monitoring, which benefits from many agents each monitoring a small area. Eventually, our ideas will be tested aboard the University of New Hampshire Marine Systems Engineering Laboratory's EAVE AUVs (Blidberg & Chappell, 1986). The examples in this section are from the AUV domain.

Suppose that an AUV is at position A in Figure 1 and receives a new goal to move the rock at position R. The AUV could simply go to position R and pick up the rock. Here the AUV would use a procedural schema that tells it how to move to the new location. Part of this procedural schema would be to call its path-planner which would guide the AUV from A to R.

Unfortunately, going from place to place underwater is not always "smooth sailing". Figure 2 shows a kelp bed between the AUV and the rock. The kelp bed is an example of a *natural boundary* in the AUV domain. Natural boundaries, such as vegetation, high rocks, outcrops and strong currents, are especially difficult or dangerous for AUVs to cross. The environment can be partitioned into spaces that are surrounded by natural boundaries (Fisher, 1991). Procedural schemas that minimize the number of natural boundary crossings exist for travel among often-visited sets of areas. These schemas' predictions help to decide when and where new goals may best be achieved.

[1]This work was supported in part by NSF grant number BCS-8905888 (under agreement number 900622CR001-IT1001A with the University of Hawaii PICHTR), ONR grant N00014-89-J-3074, and a grant from the Hubbard Marine Program Endowment. The authors are listed in alphabetical order.

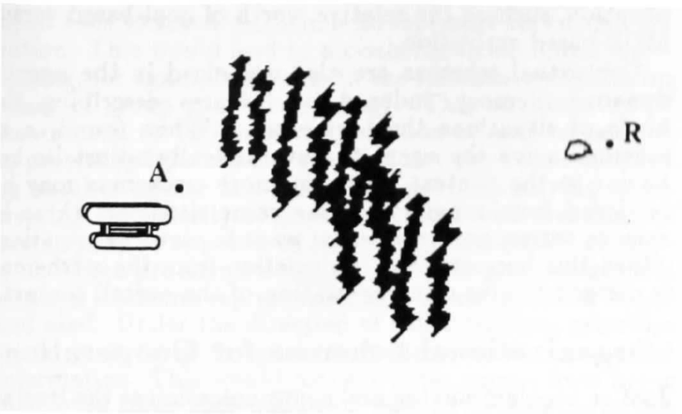

Figure 2: AUV with goal to pick up an object when there is a natural boundary present.

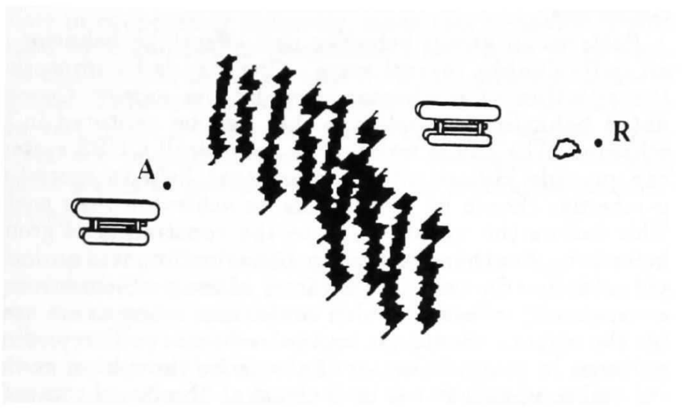

Figure 3: Two AUVs, one with goal to pick up an object, when there is a natural boundary present.

If the new goal can be satisfied in an area that will be visited as part of the schema, whether because the area contains another active location or because it is on the best path between two areas, the goal will be satisfied at that location and can be delayed until the location is reached.

If the AUV is not working alone, as shown in Figure 3, the context of working in a group gives the AUV another option. In this case, the AUV is being guided by a contextual schema representing "working with others", which allows agents to ask for help. Information about the organizational context has priority over information about the agent's individual context, so the agent commits to achieving the goal by asking another agent and the goal's priority is left the same. A procedural schema for communication tells the agent how to send a message to request the other's help.

Communication can be further affected by the organization of the agents. This is mediated by contextual schemas for different kinds of organizations. One kind of organization is a hierarchy. Suppose in this context that the AUV near A sends a message to the AUV near R requesting that the receiver adopt the goal of moving the rock. If the sender has authority over the receiver, then the receiver will raise the priority of this goal over its own goals' priorities. If the receiver has authority over the sender, then it will decide what the goal's priority should be.

The Relationship between Problem Solving, Communication and Cooperation

Cooperation, communication, and problem solving are all related. Goals related to communicating with another agent can be achieved via problem solving (e.g., (Searle, 1969)) and goals related to problem solving can often be achieved via communication (e.g., (Appelt, 1985)). Coordination between agents' problem solving is also achieved via communication, for example, by communicating partial plans (Durfee *et al.*, 1987). Cooperation, communication, and problem solving are also more deeply related by the way in which they rely on patterns in the world.

Cooperation, communication, and problem solving all benefit from the ability to make predictions about future actions. When an agent commits to a plan of action, its plan can be seen as predictions that allow it to select an order in which to achieve its goals. This allows goals to be grouped together appropriately instead of simply relying on the priority of the goals to provide an ordering. For example, in communication, goals related to the same "topic" should appear together in the conversation (Turner & Cullingford, 1989a); in problem solving, goals that can be achieved near each other should be grouped together (Fisher, 1991).

Commitments to future actions also allow agents to coordinate their behavior with others' and with the external world in a way that cannot be done by simply reacting to immediately-perceivable external conditions (cf. (Agre & Chapman, 1987; Brooks, 1986)). For example, the agent can commit to being at some location at a particular time (e.g., for a rendezvous) or to communicating with another agent at a future time.

Cooperation is facilitated by an agent's ability to predict other agents' behavior. These predictions not only make another's behavior easy to understand, but also allow an agent to coordinate its behavior with another agent's. The most obvious example of this kind of coordination is the use of shared convention in communication. Language users share knowledge of syntax and word meaning as well as knowledge of how conversations are conducted. When language users hear a question, they not only understand what they are being asked, but also that they should answer. We believe that sharing patterns or conventions for problem solving will facilitate cooperation by making it easier for agents to recognize and participate in each others' plans.

Schemas for Planning and Conversation

In our approach, an agent organizes as much of its knowledge as possible in schemas which reflect the underlying patterns present in the agent's environment, its behavior, and its relationships to other agents. This has the effect of reducing the size of the space in which the agent must search for applicable knowledge by clustering knowledge into schemas. It also reduces the size of the search space involved in planning, since each "operator" (planning schema) spans more of that search space than a single primitive operator would (e.g., (Fikes *et al.*, 1972)).

Two types of schemas are used to represent an agent's knowledge: procedural schemas and contextual schemas. Procedural schemas, or p-schemas, control the actions an agent takes to achieve goals. They are based on procedural schemas (Turner, 1989b) developed in our earlier work on reactive planning and on conversation MOPs (Turner & Cullingford, 1989c) from our work on conversation. The procedural schemas presented here are very similar in structure and function to conversation MOPs (C-MOPs). Contextual schemas, or c-schemas, represent the agent's problem-solving context and provide a background against which the agent's behavior takes place. They are based on the contextual schemas in our earlier work on reactive planning (Turner, 1989c).

Procedural Schemas

Procedural schemas can be compared to hierarchical plans. Like such plans, they describe how a goal can be accomplished by a (partially-ordered) set of steps, called *scenes*.[2] A scene in a p-schema can be either a goal, another p-schema (sub-schema), or a primitive action that the agent can directly carry out. Unlike hierarchical plans, the order in which scenes are worked on during schema application depends only partially on the order specified by the p-schema. The specified order represents the *conventional* way that the p-schema is applied. However, the execution of p-schemas is flexible enough to allow these conventions to be modified by the agent's current *intentions*, reflected in its goal priorities (Turner & Cullingford, 1989c). The scenes and their ordering provide the predictions that allow an agent to organize its goals effectively (cf. (Georgeff & Lansky, 1987; Firby, 1987)).

P-schemas are organized in a dynamic memory (Schank, 1982) where they are indexed by features of the environment. The most important feature for retrieving p-schemas is the goal that the p-schema can achieve. Once retrieved, a p-schema is instantiated to form a plan (as described in detail in (Turner, 1990)). Briefly, the schema's scenes are specialized based on the current situation using the memory, and attention is focused on a particular scene to execute based on an activation metaphor which takes into account both commitment to intention as encoded in the schema ("MOP-based activation") and the agent's current intentions as encoded in the priorities of its goals ("goal-based activation"). As attention is focused, scenes may be re-specialized to reflect the evolving problem-solving situation.

Contextual Schemas

Contextual schemas hold knowledge about kinds of problem-solving situations; they can be thought of as generalized *cases* of problem solving (e.g., (Kolodner *et al.*, 1985)). The information in c-schemas is used to control the overall character of the agent's behavior so that it is appropriate for the situation the agent is in.

C-schemas impact several areas of an agent's behavior (Turner, 1989c). They provide suggestions of goals the agent should satisfy in a particular situation and suggestions of how behavioral parameters should be set. For example, a c-schema representing the context of being in a harbor would set bounds on both an AUV's depth (e.g., don't go too near the surface nor near the bottom) and its speed (e.g., don't go too fast, since there are likely to be other vessels and obstacles present). C-schemas also provide information about how unanticipated events should be handled in a context-specific way, generally by creating a new goal in response. They also provide information about the relative importance of kinds of goals in the situation, which aids the agent's attention-focusing process. C-schemas provide information about the appropriate ways to achieve goals in a given context by suggesting which p-schemas should be used to achieve the goals. Finally, c-schemas impact the manner in which p-schemas are applied by changing parameters affecting schema application, such as the relative worth of goal-based versus MOP-based activation.

Contextual schemas are also organized in the agent's dynamic memory, indexed by features describing the kinds of situations they represent. When found, a c-schema causes the agent to automatically adjust its behavior to the context. Two or more c-schemas may be retrieved from memory for the same situation; these in essence correspond to different ways to view the situation. When this happens, the information from the c-schemas is merged to give a representation of the overall context.

Organizational Schemas for Cooperation

Just as problem solving and communication at the level of the individual agent follows patterns, group behavior also follows patterns. When an agent participates in a group, its behavior will be modified by its role in that group as well as by the current phase of the group problem-solving effort. Group behavior is captured by both p-schemas and c-schemas.

Patterns in group behavior can affect the behavior of an individual in several ways. One way is by impacting the selection of p-schemas used by the agent. Coordinated behavior has patterns that can be captured in p-schemas. The characteristics of the overall CDPS system can provide indices into memory that help to specialize p-schemas chosen by individuals for achieving their goals. This tailors the agent's plan to the constraints of group behavior. Another way group behavior impacts an individual is by affecting the character of the problem solving, consequently affecting which contextual schemas are used by the agent. Some contextual schemas will represent patterns in group behavior that can be thought of as the "global context"; this is in contrast to the "local context" provided by the contextual schemas an agent would use when acting alone. At any given time, due to features of the group problem-solving situation, c-schemas representing both global and local context may be in use by an agent. In our approach, information about the global group effort has precedence over information about the local activity of an individual; consequently, c-schemas related to group activity take precedence over c-schemas related to local activity. Although powerful individuals may achieve their local goals before the goals given to them by the group, their ability to do this is derived from their power in the group—not from simply ignoring the group.

In the remainder of this section, we discuss two types of observed patterns in coordinated behavior and how they can be fit into our schema-based approach.

Example 1: Phases of coordinated problem solving. Group problem solving progresses in phases as the agents move from getting to know each other and understanding the task to dividing the task and actually performing it (Bales & Strodtbeck, 1951; Mabry, 1975). These phases can be used to guide the behavior of the agent and predict the behavior of others. Consequently, the phases are represented in a p-schema with the scenes: *information-gathering*, *task-division*, and *goal-attainment*.

However, such a p-schema does more than simply organize related actions into scenes. Each of the p-schema's scenes also influences how problem solving and communication is conducted during that scene. Consequently, the current phase of problem solving, represented by the scene in the organization p-schema that the group is currently executing, provides a kind of context for problem solving and communication, and would figure into the indices the

[2]Since p-schemas grew out of the work on C-MOPs, the terminology of MOPs (Schank, 1982) is retained.

agent uses to select c-schemas to represent its current situation. This would lead to a c-schema being found representing, for example, "being in the information-gathering phase of group problem solving". This c-schema would recommend or facilitate the selection of actions relating to providing information that has not yet been requested by another agent. It would affect problem solving by increasing the priority of goals associated with creating the overall plan as opposed to satisfying some particular subgoal. When problem solving moves to the goal-assignment phase, a c-schema representing that phase would be found and used. Under the direction of this c-schema, priorities would be lowered for goals to communicate unrequested information. This would prevent other agents from being distracted from their assigned task to process incoming messages. The c-schema might also cause agents to lower the priorities of goals not directly related to their individual tasks.

The use of p-schemas to provide indices for c-schema selection is new. We believe that the need for this mostly arises in cooperative behavior, since this behavior is governed by organization p-schemas which span large periods of time and many actions by many different kinds of agents. This means that the scenes of such a p-schema are very general in nature, and that such scenes themselves are important indicators of the agent's context, unlike scenes of p-schemas related to individual problem solving.[3]

Example 2: Chain of command of the agents. A great deal of attention in CDPS has been devoted to the way that agents can relate to each other. This is often called the *organization* of the agents and refers to the power structure between the agents (like the organization of a company) instead of to the physical connections between agents. Many organizations have been suggested ranging from a *collective*, or flat structure where all agents are equal, to *master–slave* relationships where one agent has power over another, to *hierarchical* structures where agents can have one of several levels of power (e.g., (Malone, 1987)). In hierarchical organizations, agents may be equal to some agents, in a position of power over others and subordinate to still others. Another way to organize agents is to allow them to choose the organizational structure which best fits their current situation (Gasser *et al.*, 1989) so that their relationship with others is dynamic.

The relationship of one agent in a hierarchy to others affects the nature, method, and timing of problem-solving actions and communication. The relationship can best be represented by contextual schemas, which we can think of as organization c-schemas. Although type of organization and an agent's position in it can be used as indices to the appropriate contextual schema, it is unnecessary to have a separate contextual schema for each possible combination of positions in each organization. Instead, all of the implications of the organizational structure for the agent's behavior can be captured by using a combination of c-schemas: one representing the kind of organization and others, drawn from c-schemas representing generic relationships (*equality*, *authority*, and *subordination*), capturing the relationships between the agent and others in the organization.[4] Each of these contexts affect behavior differently, but the summed contributions from the c-schemas produce behavior appropriate for the agent's role in the organization.

The most obvious example of how these contexts can affect behavior is what happens when one agent needs help in achieving a goal. When an agent makes a request of another agent, it is asking that agent to cooperatively assume some goal. For example, if an agent asks another to move a rock (like the too–distant rock in the example which opens this paper), the agent wants the other to assume the goal of moving the rock. Once we view this request as a new goal for the other agent, it is easy to see how the context will affect the problem solving of the agent that has assumed the goal. If the receiving agent's relationship to the other is captured by the c-schema representing equality, the agent may not change the priority of the goal at all. If the agent is subordinate to the other, the priority of the assumed goal will be increased, to reflect the fact that the authority's goals are more important to the overall problem solving than the subordinate's goals. If the agent has authority over the agent making the request, the assumed goal may have its priority lowered to reflect that the goals of the authority have a higher relative priority to the goals of the subordinate. By changing the priorities based on the context of relative power, the agents can automatically determine how they should respond to requests from other agents.

The relationship between the requesting agent and the agent which will provide aid can also affect the procedural schema chosen by the requesting agent to effect the request. If the requesting agent is in control of the other, it selects a p-schema that simply makes the request; in this case, the agent assumes that the request will be carried out unless it is informed otherwise. On the other hand, if the agent is in a position of subordination, it is in a situation similar to that of reasoning under uncertainty. In this case, the p-schema selected should make the request, then check its environment to see if the goal has actually been achieved (i.e., the request honored) by the other agent.

The context determined by the organization and an agent's position in it also affects the p-schemas the agent chooses to satisfy its goals. So, in our example of needing to request help from others, we can expect that each agent will have several specializations for the general p-schema for getting help from others. An agent needing help might, for example, use a p-schema which simply requests help without giving any rationale or a p-schema which makes the request, but in addition gives reasons why the help is requested or arguments for why the help should be given. The organization context of the agent, represented by a c-schema or set of c-schemas, should provide suggestions about which specialization is appropriate. If the agent is in a position of authority, it can simply send the message that requests help. If it is a subordinate or an equal, it will choose the p-schema which supplies reasons why the other should assume its goal and provide the help.

The use of c-schemas to represent organizational information has another important advantage for cooperative problem solving. It allows the agents' behavior, both individually and collectively, to adapt to situations in which the organization of the agents needs to be dynamic. The simplest way that our approach supports dynamic organization is that agents can change their view of the organi-

[3]However, in the AUV domain, missions of long duration may also be represented by p-schemas with some of the same character, possibly leading to their scenes contributing to the selection of c-schemas (e.g., for the mission phases of going to the work site, working, and returning home).

[4]These contexts could affect the behavior in different degrees, if necessary. For example, extreme authority (like the president of a company talking to the mail clerk) may have a more extreme effect on the context-based decisions.

zational context as easily as they can respond to changes in their environment (Turner, 1989c); in both cases, features of the changed situation provide indices which allow the retrieval of c-schemas representing the new situation. In addition, changes may cause the agent to notice, based on information contained in its contextual schemas, that an organizational change is needed. In this situation, the schema-based approach can allow dynamically adjusting the organization of the problem solvers as well as their individual problem-solving behavior by changing which schemas are in use.

Conclusion

In this paper, we have sketched a schema-based approach to cooperative distributed problem solving. Schemas are used to represent an agent's knowledge of how to solve problems, communicate, and interact with other agents in the CDPS system. Procedural schemas contain knowledge about sequences of actions. They are used to achieve problem-solving goals, to communicate, and to coordinate an agent's actions with those of others by providing an explicit structure for global problem solving behavior. Contextual schemas contain knowledge about problem-solving and organizational contexts. They are used to adjust the character of an agent's behavior to fit the situation it is in. Contextual schemas impact the use of procedural schemas, both by helping to select them as well as by modifying, in a context-specific manner, how they are applied. Procedural schemas, especially those at the organizational level, impact the use of contextual schemas by providing indices that help the agent select c-schemas to represent the context it is in.

One advantage of our approach is that knowledge is organized so as to capitalize on patterns in the world; this facilitates efficiently bringing the right knowledge to bear at the right time. Another advantage is that schemas allow predictions to be made about how oneself and others will behave in the future. These predictions enable agents to more easily coordinate their actions with those of others. In addition, a schema-based approach to representing and using organizational knowledge potentially allows a CDPS system's organizational structure to change as the situation demands.

This work is still in the early stages of development. As our work progresses, we will refine and evaluate our ideas in the domain of multi-AUV systems, both in simulation and during in-water tests aboard AUVs.

References

Agre, P. E. and Chapman, D. 1987. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 268–272.

Appelt, D. E. 1985. Planning English Referring Expressions. *Artificial Intelligence*, 26:1–33.

Bales, R. F. and Strodtbeck, F. L. 1951. Phases in group problem solving. *Journal of Abnormal and Social Psychology*, 46:485–495.

Blidberg, D. R. and Chappell, S. G. 1986. Guidance and control architecture for the EAVE vehicle. *IEEE Journal of Oceanic Engineering*, OE–11(4):449–461.

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA–2(1):14–23.

Durfee, E. H.; Lesser, V. R.; and Corkill, D. D. 1987. Coherent Cooperation between Communicating Problem Solvers. *IEEE Transactions on Computers*, C(36):1275–1291.

Durfee, E. H.; Lesser, V. R.; and Corkill, D. D. 1989. Cooperative Distributed Problem Solving. In Barr, A.; Cohen, P. R.; and Feigenbaum, E. A., eds, *The Handbook of Artificial Intelligence*, volume IV, chapter XVII, pages 83–147. Addison-Wesley Publishing Company, Inc., Reading, MA.

Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251–288.

Firby, R. J. 1987. An Investigation into Reactive Planning in Complex Domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 202–206, Los Altos, California. Morgan Kaufmann.

Fisher, T. L. 1991. Exploiting Natural Boundaries in the Autonomous Underwater Vehicle Domain to Limit Resource Utilization.

Gasser, L.; Rouquette, N. F.; Hill, R. W.; and Lieb, J. 1989. Representing and using organizational knowledge in distributed AI systems. In Gasser, L. and Huhns, M. N., eds, *Distributed Artificial Intelligence, Volume II*, chapter 3, pages 55–78. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Georgeff, M. P. and Lansky, A. L. 1987. Reactive reasoning and planning: An experiment with a mobile robot. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 677–682, Seattle, Washington.

Jaffe, J. 1988. Underwater Imaging System Characterization. Technical Report H–88–33, Woods Hole Oceanographic Institute.

Kolodner, J. L.; Simpson, R. L.; and Sycara-Cyranski, K. 1985. A process model of case-based reasoning in problem-solving. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Los Angeles, California.

Mabry, E. A. 1975. Exploratory analysis of a developmental model for task-oriented small groups. *Human Communication Research*, 2:66–74.

Malone, T. W. 1987. Modeling coordination in organizations and markets. *Management Science*, 33(10):1317–1332.

Schank, R. C. 1982. *Dynamic Memory*. Cambridge University Press, New York.

Searle, J. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, UK.

Turner, E. H. 1990. Integrating Intention and Convention to Organize Problem Solving Dialogues. Technical Report GIT–ICS–90/02, School of Information and Computer Science, Georgia Institute of Technology. Ph.D. thesis.

Turner, E. H. and Cullingford, R. E. 1989a. Making conversation flexible. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pages 932–939, Detroit, MI.

Turner, E. H. and Cullingford, R. E. 1989b. Using Conversation MOPs in Natural Language Interfaces. *Discourse Processes*, 12(1):63–90.

Turner, E. H. and Cullingford, R. E. 1989c. Using Conversation MOPs in Natural Language Interfaces. *Discourse Processes*, 12(1):63–91.

Turner, R. M. 1989a. *A Schema-based Model of Adaptive Problem Solving*. PhD thesis, School of Information and Computer Science, Georgia Institute of Technology. Technical report GIT–ICS–89/42.

Turner, R. M. 1989b. Using schemas for diagnosis. *Computer Methods and Programs in Biomedicine*, 30(2/3):199–208.

Turner, R. M. 1989c. When reactive planning is not enough: Using contextual schemas to react appropriately to environmental change. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pages 940–947, Detroit, MI.