AI Magazine Volume 17 Number 2 (1996) (© AAAI)

AAAI–94 Presidential Address

Collaborative Systems

Barbara J. Grosz

■ The construction of computer systems that are intelligent, collaborative problem-solving partners is an important goal for both the science of AI and its application. From the scientific perspective, the development of theories and mechanisms to enable building collaborative systems presents exciting research challenges across AI subfields. From the applications perspective, the capability to collaborate with users and other systems is essential if large-scale information systems of the future are to assist users in finding the information they need and solving the problems they have. In this address, it is argued that collaboration must be designed into systems from the start; it cannot be patched on. Key features of collaborative activity are described, the scientific base provided by recent AI research is discussed, and several of the research challenges posed by collaboration are presented. It is further argued that research on, and the development of, collaborative systems should itself be a collaborative endeavor—within AI, across subfields of computer science, and with researchers in other fields.

AI has always pushed forward on the frontiers of computer science. Our efforts to understand intelligent behavior and the ways in which it could be embodied in computer systems have led both to a richer scientific understanding of various aspects of intelligence and to the development of smarter computer systems. In his keynote address at AAAI-94, Raj Reddy described several of those advances as well as challenges for the future. The Proceedings of the AAAI Innovative Applications of Artificial Intelligence conferences contain descriptions of many commercial systems that employ AI techniques to provide greater power or flexibility.

For this Presidential address, I have decided to focus on one such frontier area: the understanding of collaborative systems and the development of the foundations—the representations, theories, computational models and processes—needed to construct computer systems that are intelligent collaborative partners in solving their users' problems. In doing so, I follow the precedent set by Allen Newell in his 1980 Presidential Address (Newell 1981, p. 1) of focusing on the state of the science rather than the state of the society. I also follow a more recent precedent, that set by Daniel Bobrow in his 1990 Presidential address (Bobrow 1991, p. 65), namely, examining the issues to be faced in moving beyond what he called the "isolation assumptions" of much of AI to the design and analysis of systems of multiple agents interacting with each other and the world. I concur with his claim that a significant challenge for AI in the 1990s is "to build AI systems that can interact productively with each other, with humans, and with the physical world" (p. 65). I will argue further, however, that there is much to be gained by looking in particular at one kind of group behavior, collaboration.

My reasons for focusing on collaborative systems are two-fold. First, and most important in this setting, the development of the underlying theories and formalizations that are needed to build collaborative systems as well as the construction of such systems raises interesting questions and presents intellectual challenges across AI subfields. Second, the results of these inquiries promise to have significant impact not only on computer science, but also on the general computer-using public.

This talk was presented at the American Association for Artificial Intelligence's National Conference on Artificial Intelligence, 3 August 1994, in Seattle, Washington

Why Collaborative Systems?

There is no question that there is a need to make computer systems better at helping us

For systems to collaborate with their users in getting the information they need and solving the problems they have will require the kind of problem solutions and techniques that AI develops. But it also requires that we look beyond individual intelligent systems to groups of intelligent systems that work together.

do what we use them to do. It is common parlance that the world has become a global village. The daily news makes clear how close events in any one part of the world are to people in other parts. Use of the Internet has exploded, bringing people closer in another way. In July 1994 we had a fine demonstration of the very best use of this technology: astronomers around the world used the net heavily, collaborating to an unprecedented extent, sharing information about the comet Shoemaker–Levy 9's collision with Jupiter. As they have used the information gathered in those observations to understand better the planet's atmosphere, its inner structure, comets, and the like, they continue to employ the communications capabilities of the net. But the net could surely provide more.

There is frequent mention in the news of the information superhighway and electronic libraries, the development of technology that will rapidly get us the information we need—and, one hopes, not too much of what we don't need. But one also frequently hears, both in the media and from individuals—at least if your friends are like mine—of the frustrations of trying to get "hooked in." In a similar vein, in his invited talk at AAAI-94 addressing key problems of software in the future, Steve Ballmer from Microsoft brought up repeatedly the need to make software more "approachable." An article on the first page of the business section of the July 22, 1994 *Boston Globe* (Putzel 1994, p. 43) had the headline, "Roadblocks on Highway: Getting on the Internet can be a Real Trip." The article began, "So you want to take a ride on The Superhighway. Pack plenty of patience, and strap in tight. The entry ramp is cratered like the moon." It goes on to say, "The 'Net' offers access to huge storehouses of information … [but] ordinary people … are having a devilish time breaking down the Internet door. It's like a huge private club whose membership is guarded by a hundred secret handshakes."

The thing is, even after you get in the club, there's not much help getting what you really need. AI can play a unique and pivotal role in improving the situation, in making the Superhighway an *information* superhighway, not just a *gigabit* superhighway. The work of Etzioni, Maes, Weld and others on softbots and interface agents is an important step in this direction (Etzioni and Weld 1994; Maes 1994; Etzioni 1993).

Limitations of the means currently available for communicating with computer systems are a major stumbling block for many users. Direct manipulation interfaces are often touted as providing the advance that makes computing power accessible to one and all. Now users can just "point and click" to get what they need. They don't have to tell the computer system how to do things, just what to do. But, in reality, this is true only at a shallow level: for many applications, users no longer need to write programs or deal with obtuse command languages. But at the deeper problem-solving or task level, users still must tell the computer system how to do what's needed. Systems don't have a clue about what users are really trying to achieve. They just do a bunch of pretty low-level chores for a user, in service of some larger goal about which the systems know nothing. The system is a tool, no doubt a complex one, but it's a dumb servant rather than a helpful assistant or problem-solving partner. Users say jump, and, if they are lucky, the system figures out how high rather than replying with

usage: jump [-h] height [-v] vigor.

To have systems that help solve our problems without having to be told in laborious, boring detail each step they should take requires freeing users from "having to tell" at a different level. It wouldn't be too far-fetched to say that the mouse frees users from having to tell the system what to do at, or perhaps below, the symbol level, but it fails to provide any assistance at the knowledge level. Mice and menus may be a start, but they're far from the best that we can do to make computers more accessible, helpful, or user friendly.

For systems to collaborate with their users in getting the information they need and solving the problems they have will require the kind of problem solutions and techniques that AI develops. But it also requires that we look beyond individual intelligent systems to groups of intelligent systems that work together.

So much for politics and technology. Let's not lose sight of the fact that collaborative behavior is interesting in its own right, an important part of intelligent behavior. For at least two reasons, we should not wait until we've understood individual behavior before we confront the problems of understanding collaborative behavior. First, as I will show later, the capabilities needed for collaboration cannot be patched on, but must be designed in from the start. Second, I have a hunch that looking at some problems from the perspective of groups of agents collaborating will yield easier and not just different solutions.[1]

Examples of Collaborative Activity

That collaboration is central to intelligent behavior is clear from the ways in which it pervades daily activity. Figure 1 lists a small sampling of them. They range from the well-coordinated, pre-planned and practiced collaborations of sports teams, dancers, and musical groups to the spontaneous collaborations of people who discover they have a problem best solved together. Scientific collaborations occur on large and small scales and across the sciences and social sciences. The large scale is exemplified by the coordinated efforts of the 400 physicists who participated in finding the top quark as well as by the astronomers observing Shoemaker—Levy 9 collide with Jupiter. Archeologists attempting to produce a coherent explanation of the culture represented by the finds at a dig typically collaborate in somewhat smaller groups of specialists from different subfields.

To help illustrate some of the features of collaborative activity, I will examine in more detail two small-scale collaborations, one in health care and the other in network maintenance. The health-care example involves three people working together toward common goals; the network-maintenance example illustrates a human-computer team effort. Later, I will give an example of a collaborative group consisting solely of machines. Although I will not have time to discuss work on systems that support groups of people collaborating, this is another area to which the results of our research could contribute.

In the health care scenario portrayed in figure 2 a patient arrives at the hospital with three problems affecting his heart and lungs. Three specialists are needed, each providing different expertise needed for curing the patient. But this is not a one doctor, one disease situation. Treating the patient will require teamwork. For example, the cardiologist and pulmonary specialist must agree on a plan of action for reducing the water in the patient's lungs. And the infectious disease and pulmonary specialists must plan together the kind and amount of antibiotic to prescribe.

Notice, however, that this is a team of equals. No single doctor is the manager, telling the others who does what; there's no master-servant relationship here. The doctors need to come to a consensus about what to do and who's going to do it. Each of them has only some of the information needed to devise a plan for action; they will have to plan jointly. In doing so, each doctor will make presumptions about the others' beliefs and capabilities and must decide when to check out these assumptions. Each doctor will be counting on the others to contribute what they know to the solution and to do their share.

COLLABORATIONS IN DAILY LIFE

- sports and entertainment:
  - soccer, dance, orchestra
- science:
  - archeological digs
  - high energy physics
- design:
  - building a house
  - constructing a computer system
- health care

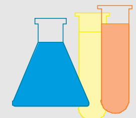


Figure 1. Collaborations in Daily Life.

EXAMPLE 1: HEALTH CARE

- patient problems:
  - heart attack with congestive heart failure
  - pneumonia
  - emphysema
- the team:
  - cardiologist
  - infectious disease specialist
  - pulmonary specialist
- the collaborations:
  - diuretics: cardiologist & pulmonary specialist
  - antibiotics: pulmonary specialist & infectious disease specialist

Figure 2. Example 1: Health Care.

The dialogue in figure 3, taken from the work of Lochbaum (1994), illustrates that "a buddy is better than a slave": a system that works with users to solve problems jointly is better than one that takes orders. For example, in response (2), the system doesn't just report that Jupiter is down, but goes on to fix

EXAMPLE 2: NETWORK MAINTENANCE

(1) *User:*	We need to repair a connectivity problem between Mars and Saturn. Do a remote ping from Mars to Saturn.
(2) *System:*	I can't. Saturn seems to be down. I'll take care of that first.
(3)	<System reboots machine>
(4) *System:*	Okay, Saturn's back up and the remote ping was successful.
(5) *User:*	Good. Verify Mars' IP address for Saturn for me.
(6) *System:*	The entry for Saturn was wrong, but I corrected it.
(7) *User:*	Okay, good. We're done then.

Figure 3. Example 2: Network Maintenance (Lochbaum, 1994).

the problem; it acts in service of achieving the shared goal it has with the user. In (6), the system again acts to remove a barrier; it does so without requiring an additional push of a mouse button.

As in the health care example, here too different participants have different knowledge about how to proceed as well as about the conditions that pertain. To determine the appropriate course of action, they need to share information. The participants also have different capabilities for acting. The user is counting on the system to do its part in furtherance of the common goal; not to give up when something goes wrong, but to try to address the problem; and to report back on events or properties that are important to the situation and to their problem solving.

The Challenge of Simple Organisms

Our analysis of collaboration should not be restricted to complex systems that require explicit models of beliefs, desires, intentions, plans and the like. Simple organisms also work together. Ant behavior provides a compelling example. We all know—some too well—that ants leave trails for other ants to follow.[2] But this is one of the simpler ways in which ants recruit others in their colony to work toward getting food for the group. Another approach is tandem running: In his book *Insect Societies*, E.O. Wilson (1971) reports that,

> When a worker of the little African myrmicine ant *Cardiocondyla venustula* finds a food particle too large to carry, it returns to the nest and contacts another worker, which it leads from the nest. The interaction follows a stereotyped sequence. First the leader remains perfectly still until touched on the abdomen by the follower ant. Then it runs for a distance of approximately 3 to 10 mm, or about one to several times it own body length, coming again to a complete halt. The follower ant, now in an excited state apparently due to a secretion released by the leader, runs swiftly behind, makes fresh contact, and "drives" the leader forward. (p. 248)

Social insects collaborate on more than obtaining food. Weaver ants, termites, and social wasps build large complex nests that take many worker lifetimes to complete (Wilson 1971, p. 228ff), and many species collaborate on raising the next generation. In describing this behavior, Wilson makes an assertion that we should keep in mind as we attempt to develop intelligent systems. He says,

> The individual social insect, in comparison with the individual solitary insect, displays behavior patterns that are neither exceptionally ingenious nor exceptionally complex. The remarkable qualities of social life are mass phenomena that emerge from the meshing of these simple individual patterns by means of communication. In this principle lies the greatest challenge and opportunity of insect sociology. (Wilson 1971, p. 224)

This observation also poses a challenge for us in our efforts to design intelligent systems. In particular, it suggests we consider how designing systems to work together might enable us to design simpler individual systems and still accomplish complex goals. Bajcsy's work on robots that coordinate physical tasks (Kosecka and Bajcsy 1993) is a step in this direction. However, we can also look in simpler settings. A somewhat surprising theoretical result (Bender and Slonim 1994), which I'll discuss in more detail later, shows that two robots are better than one in certain map-learning situations. For example, the lighter portion of the graph fragments in figure 4 illustrate configurations that two agents can learn to distinguish, but one cannot.

One interesting issue raised by these "simple societies" is how much of their behavior is collaborative and not merely coordinated or interactive, a distinction we'll examine soon.

Another is the extent to which collaborative behavior should be hardwired into systems. An important question for us to address is the extent to which collaboration can be achieved with simple techniques and the extent to which it requires reasoning with explicit models of beliefs, intentions, and the like.

Collaboration versus Interaction

Dictionary definitions (Mish 1988) provide a crisp way of seeing the distinction between collaboration and interaction. Whereas interaction entails only acting on someone or something else, collaboration is inherently "with" others; working (labore) jointly with (co). It's the "jointly with" that distinguishes collaboration and that we need to characterize more precisely. To build collaborative systems, we need to identify the capabilities that must be added to individual agents so that they can work with other agents. Later, when I examine what's required to model collaboration, I will argue that collaboration cannot just be patched on, but must be designed in from the start.

To characterize "jointly" will require a consideration of intentions. The importance of intentions to modeling collaboration is illustrated by the clip-art maze in figure 5. From the picture alone, it is not clear whether the mice in this maze are collaborating. The picture doesn't reveal if they have some shared goal or if instead, for example, the top mouse has paid the bottom mouse to be his stepstool. We need to know something about the goals and intentions of these mice to be able to decide whether they are collaborating. Information about intentions is central to determining how they will behave in different circumstances. For example, such information is needed to figure out what the mice will do if something goes wrong: if the top mouse fails to get over the wall, will the bottom mouse help it find another plan, that is, will the mice replan together about how to reach their goal?

Another way to view the distinction between collaboration and interaction is to consider whether we want computer systems to be merely tools, or something more. The contrast among the simple natural language examples in figure 6 clearly illustrates this difference. The assistance Sandy received from Pat in writing the paper was quite different from the pen's contribution. The pen only left marks on paper, whereas Pat produced words, paragraphs, maybe even sections, not to mention ideas.

TWO ROBOTS ARE BETTER THAN ONE

Fragments from a strongly connected graph illustrating setting in which a single robot could get confused.

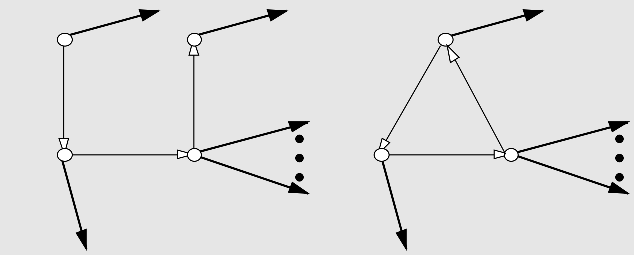
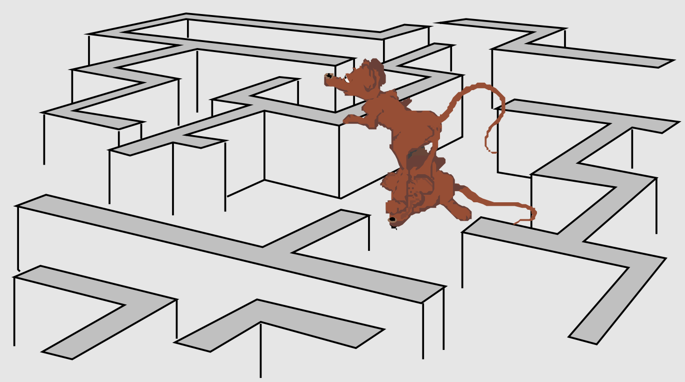


Figure 4. Two Robots Are Better Than One (Bender & Slonim, 1994).

COLLABORATION VS. INTERACTION



The two mice trying to get out of this maze are interacting, but are they collaborating?

Figure 5. Collaboration versus Interaction.

Pat and the pen form a spectrum. The question is where the computer system sits on this spectrum now. Systems are not as passive as the pen (for example, they will do spelling correction), but they're not nearly as helpful as Pat. They can't help formulate an outline or the approach to take nor identify the important points to be made, nor can

TOOLS OR COLLABORATORS?

- Sandy wrote the paper with a fountain pen.
- Sandy wrote the paper with Pat.
- Sandy wrote the paper with the computer.

Figure 6. Tools or Collaborators?

they write complete sections. More particularly, we can ask what's needed so that systems will become more like Pat than the pen in what they contribute to our efforts.[3]

Why Collaborative Systems Now?

Before moving on to look at some of the issues that modeling collaborative systems raises, I want to pause to address the question you might be asking: why is now the time to take on this challenge? There are three main reasons. First, as I mentioned earlier, modeling collaboration presents a variety of challenging research problems across AI subfields. But this would be irrelevant were it not for the second reason: progress in AI over the last decade.

AI is situated to make major advances in collaborative systems because of the substantial scientific base established by research across most of the areas involved in meeting this challenge. For example, work in both the natural language and DAI communities has provided a range of models of collaborative, cooperative, multi-agent plans. The planning community's efforts to cope with uncertainty and the pressures of resource constraints likewise provide a rich source of ideas for coping with some of the modeling problems that arise in the design of collaborative systems. Results from the uncertain and nonmonotonic reasoning communities can be drawn on to address the kinds of ascription of beliefs and intentions that arise in reasoning about what others can do as well as about what they know.

Third, as I discussed at the beginning of this article, there are compelling applications needs. Again, recent progress in AI matters. AI is uniquely poised to provide many of the capabilities that are needed by virtue of the research base I've just sketched and because the kinds of problems we address and models we develop are central to providing them.

Characteristics of Collaboration

In this next section, I will characterize the phenomenon of collaboration more precisely and will describe the main properties of collaborative activity. Then I will examine the key elements that we need to model. Next I will show how these characterizations enable us to distinguish collaboration from other kinds of group activity. In the remainder of the article, I will then turn to look at some of the major research challenges presented by collaboration and the research collaborations we need to form to meet them.

Features of Collaboration

In examining the features of collaborative activity, I'll draw on what I've learned about collaboration in joint work with Candy Sidner and Sarit Kraus.[4] If we look at collaborative activity in daily life, we see several features of the participants' knowledge, capabilities, and relationships that affect the kinds of reasoning needed to support the construction of collaborative computer agents. First, as the health care and network examples illustrate, the master-servant relationship, which is currently typical of human-computer interaction, is not appropriate for collaborations. Second, as the examples also make clear, most collaborative situations involve agents who have different beliefs and capabilities. Partial knowledge is the rule, not the exception. As a result, collaborative planning requires an ability to ascribe beliefs and intentions (that is, to guess in a way that you can later retract). Third, multi-agent planning entails collaboration in both planning and acting.

These characteristics have a significant effect on the kinds of reasoning systems we can employ. We need efficient methods for nonmonotonic reasoning about beliefs and intentions of other agents, not just about facts about the world; we can't depend on once-and-for-all planning systems, nor separate planning from execution. Collaborative plans evolve, and systems must be able to interleave planning and acting. Thus, the recent moves within the planning community to cope with a range of resource-bounds issues and the like are crucial to our being able to make progress on modeling collaboration. Similarly, advances in nonmonotonic reasoning are important to handling the kinds of belief ascription needed.

The final feature of collaborative activity that I'll discuss is perhaps the most controversial one: collaborative plans are not simply the sum of individual plans. When Candy Sidner and I first investigated models of collaborative plans in the context of discourse processing (Grosz and Sidner 1990), people argued that if we would just go away and think harder we could find a way to treat them as sums of individual plans. Natural-language dialogue examples just didn't convince the planning or reasoning researchers with whom we spoke. So to convince them we were reduced to that familiar AI standby, the blocks world. The example we used is shown in figure 7. As shown at the top of the figure, the two players are building a tower of blue and red blocks; they are working together, one of them using her supply of blue blocks, the other his supply of red blocks. The question collaboration poses—and much of the rest of this article is aimed at addressing—is what their plan is. As the bottom half of the figure shows, whatever that plan is, it is not just the sum of two individual block-building plans, each with holes in the appropriate spot.

Thus, we must design collaboration into systems from the start. If they don't have mechanisms for working together—some embodiment of those mental attitudes that are needed to support the "jointly with" that characterizes collaboration—then they will not be able to form plans for joint activity. I want to turn now to investigate the mental attitudes that are required.

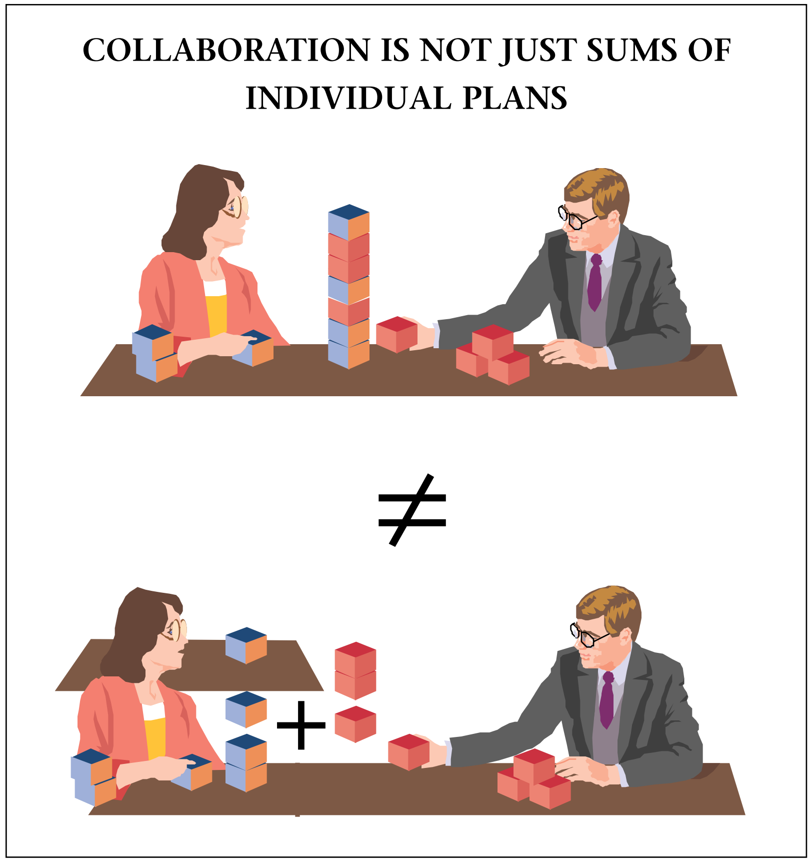


Figure 7. Collaboration Is Not Just Sums of Individual Plans.

Intending-That

First, I need to add another kind of intention to the repertoire that planning formalizations in AI have used previously. Kraus and I (Grosz and Kraus 1995) refer to this as the attitude of "intending-that."[5] Intending-that enables us to represent the commitment to others that is needed for joint activity. I will not try to define intending-that here, but rather will illustrate its role by considering several examples of collaborative activity.

To begin, I want to examine the types of responsibility that ensue when a professor and student write a paper together. For this example, we'll assume that the student and professor will each write different sections of the paper. The student intends *to* write his sections of the paper. The professor intends *that* the student will be able to write his sections. The student will do the planning required for writing his section and the subactions entailed in doing it: looking up references, producing prose and so forth. The professor's intention *that* the student be able to write his sections obliges her not to ask him to do other things at the same time: no demo crunches when there's a conference paper deadline. In addition, this intention-that may lead her to help the student when she notices doing so would further their joint goal of getting the paper written. However, the professor is not directly responsible for planning and executing the acts in writing the student's sections. And, the converse holds for the sections the professor is going to write: no last minute pleas for letters of recommendation when there's a conference paper deadline.

Meal preparation provides an example that illustrates other aspects of intending-that. Suppose two people—I'll refer to them as Phil and Betsy—agree to make dinner together. They may split up the courses: Phil making the main course, Betsy the appetizer, and the two of them making the dessert (the most important course) together. Phil intends *to* make the main course and intends *that* Betsy will be able to make the appetizer. Betsy intends *to* make the appetizer and *that* Phil will be able to make the main course. They'll

PLANS FOR COLLABORATIVE ACTION

- To have an individual plan for an act, need
  - knowledge of a recipe
  - ability to perform subacts in the recipe
  - intentions to do the subacts
- To have a group plan for an act, need
  - mutual belief of a recipe
  - individual or group plans for the subacts
  - intentions that group perform act
  - intentions that collaborators succeed

Figure 8. Plans for Collaborative Action.

each adjust their menu choices and recipes so that they won't have resource conflicts (such as both needing the same pan at the same time) and they may help each other out. For example, Phil might offer to pick up the ingredients Betsy needs when he goes to the grocery. I'll be using this example later to illustrate various elements of collaboration, but lest you misunderstand, let me say now that I'm not proposing meal-making robots as a primary application. Rather, the meals domain is a useful one for expository purposes since everyone has some intuitions about it. Thus, it requires less preliminary introduction than network management or health care. And, for some strange reason, it's less painful to discuss than paper writing.

Plans for Collaborative Action

I can now sketch what we need to add to individual plans in order to have plans for group action. I'll only have time here to lay out a piece of the picture and to do so informally, but most of what I say has formalizations, often several competing ones, behind it (see, for example, Grosz and Kraus [1995]; Kinny et al. [1994]; Levesque, Cohen, and Nunes [1990]).

Figure 8 lists the major components of group plans; to provide a base for comparison, the top of the figure lists the main components for individual plans. First, just as an individual agent must know how to do an action, agents in a group must have knowledge of how they're going to do an action. To avoid too many different uses of the word "plan," I'll follow Pollack (1990) and call this knowledge of how to do an action a *recipe* for the action. In the case of a group plan to do a joint activity, there must be mutual belief of the recipe; agents must agree on how they are going to do the action. For instance, Phil and Betsy must agree that dinner will consist of three courses of a certain sort and concur on who will make each of the courses. Second, just as individual agents must have the ability to perform the constituent actions in an individual plan and must have intentions to perform them, the participants in a group activity must have individual or group plans for each of the constituent actions in their agreed-on recipe.

Plans for group actions include two major constituents that do not have correlates in the individual plan. First, the agents must have a commitment to the group activity; they must all intend-that the group will do the action. For example, both Betsy and Phil need to have intentions that they make dinner. Among other things, these intentions will keep them both working on the dinner until the meal is on the table. Second, the participants must have some commitment to the other agents being able to do their actions. Phil must have an intention that Betsy be able to make the appetizer. This intention will prevent him from using a pan he knows she needs; it might also lead him to offer to buy ingredients for her when he goes grocery shopping.

Modeling Challenges

With this sketch of the requisite mental states of the participating agents in hand, I want to turn now to consider what we need to model to be able to build computer systems that can be collaborative partners. In this section, I'll look at three elements that have been identified as central by people working on modeling collaboration: commitment to the joint activity, the process of reaching agreement on a recipe for the group action, and commitment to the constituent actions. As we examine each of these elements, we will see there is a pervasive need for communication and a range of questions concerning the information that needs to be communicated at any given time. In addition, a significant part of the modeling challenge arises from the need for any formalization to provide for systems to cope with a general background situation in which information is incomplete, the world changes, and failures are to be expected.

Commitment to the Joint Activity

Bratman (1987) has argued that intentions play three major roles in rational action: (1) they constrain an agent's choice of what else it can intend; in particular, agents cannot hold conflicting intentions; (2) they provide the context for an agent's replanning when something goes wrong; (3) they guide "means-ends reasoning."

The commitment of participating agents to their joint activity plays analogous roles in collaborative activities. For example, Phil cannot plan to study at the library until 6 p.m. if he's going to provide a main course for dinner at 6 and make dessert with Betsy; that is, Phil cannot hold intentions that conflict with his commitment to the collaborative dinner plan. In addition, if Phil is unable to make the main course he originally planned, then his replanning needs to take into account the rest of the plan he has with Betsy. He must make a main course that will fit with the rest of the menu they've planned. In both cases, agents must weigh intentions rooted in their individual goals and desires with those that arise from the joint activity. Furthermore, Phil's commitment to the dinner-making plan will lead him to undertake means-ends reasoning not only in service of the constituent actions for which he takes responsibility, but also possibly in support of Betsy's actions. For instance, if he offers to help Betsy by buying ingredients for her while he's out shopping, he'll need to reason about how to get those ingredients.

Commitment to a joint activity also engenders certain responsibilities toward the other participants' actions. For instance, agents must be aware of the potential for resource conflicts and take actions to avoid them. Phil cannot intend to use a pot he knows Betsy needs during a particular time span; at a minimum, he must communicate with her about the conflict and negotiate with her about use of this resource.

Communication is needed not only to help decide on resource conflicts, but also when something goes wrong while carrying out a joint activity. We saw an example of both the communications requirement and replanning in the network maintenance example discussed earlier. The system did not give up just because Saturn was down; instead, it both reported the problem and went on to try to fix it.

The commitment to joint activity leads to a need to communicate in other ways as well. For example, agents may not be able to complete a task they've undertaken. If an agent runs into trouble, it needs to assess the impact of its difficulty on the larger plan, and decide what information to communicate to its partners. Although in some cases, an individual agent will be able to repair the plan on its own, in other situations the replanning will involve other group members. In some cases, the group activity may need to be abandoned. When this happens, all the group members need to be notified so no one continues working on a constituent activity that is no longer useful.[6]

Reaching Consensus on the Recipe

When a group of agents agrees to work together on some joint activity, they also need to agree about how they are going to carry out that activity. Because the agents typically have different recipe libraries (that is, they have different knowledge about how to do an action), they need a means of reconciling their different ideas of how to proceed. For example, if Phil thinks a dinner can consist of a pasta course alone and Betsy thinks no meal is complete without dessert, they need to negotiate on the courses to be included in the dinner they're making together. Furthermore, agents may need to combine their knowledge to come up with a complete recipe; they may need to take pieces from different recipe libraries and assemble them into a new recipe. So, this part of collaboration may also entail learning.

Reaching agreement on a recipe encompasses more than choosing constituent actions. Once agents divide up the constituent tasks (a topic we'll return to shortly), they need to ensure that their individual plans for these subtasks mesh. They'll need to know enough about the individual subplans to be able to ensure the pieces fit together. Betsy needs to make sure the appetizer she plans will go with Phil's main course, and that in preparing it she won't run into resource conflicts.[7] It's important to note, however, that agents do not need to know the complete individual plans of their collaborators. How much they do need to know is an open question. The two cookbook recipes in the sidebar (taken from Prince [1981]) illustrate a spectrum of detail. Notice, however, that the longer recipe contains less detail than an agent performing this action must know (for instance, it does not tell how to scrape the pig), and the shorter recipe may contain more detail than a dinner-making partner responsible for dessert needs to know. Some characteristics of the recipe-agreement process are not evident in this simple meals example. In general, agents will not be able to decide on a full recipe in advance. They'll

Communication is needed not only to help decide on resource conflicts, but also when something goes wrong while carrying out a joint activity.

Recipes: Level of Detail?

From *Joy of Cooking*
Roast Suckling Pig 10 servings
Preheat oven to 450 degrees.
Dress by drawing, scraping, and cleaning: a suckling pig.
Fill it with: Onion Dressing . . .
It takes 2 1/2 quarts of dressing to stuff a pig of this size.
Multiply all your ingredients but not the seasonings . . .
(Rombauer & Becker, 1931)

From *Répertoire de la Cuisine*
Cochon de lait Anglaise:
—Farcir farce à l'anglaise. Rôtir.
(Gringoire & Saulnier, 1914)

[English suckling pig: Stuff with English stuffing. Roast.]

need to plan a little, do some actions, gather some information, and then do more planning. Just like in the case of individual plans, they'll need to be able to modify and adapt various parts of the plan as they work their way along. They'll need to communicate about the adaptations. Typically agents will be working with partial recipes and they'll be interleaving planning and acting.

It's evident then that recipe selection entails a significant amount of decision making. One major open issue in modeling collaboration is the kinds of group decision-making processes that are needed, and ways to implement them.

Commitment to Constituent Actions

In addition to deciding what recipe they're going to use, the participants in a collaborative activity need to decide who's going to do what. They have to agree on an assignment of constituent actions to subgroups or individuals. This, again, requires a group decision-making process. It raises the question of whether to have centralized control and a team leader who decides, or a more equal partnership, as in the health care example, that requires negotiation. Different choices vary in flexibility, robustness, and communication costs. Research in the DAI community has addressed the question of task allocation; proposed solutions range from market mechanisms like contract nets (Davis and Smith 1983)—to voting mechanisms and negotiation protocols (Rosenschein and Zlotkin 1994). However, most of the techniques proposed to date have assumed individual agents were largely self-interested; there was no shared goal nor commitment to the success of the whole group. Thus, there is a need to determine the effect of these features of collaboration on the design and choice of decision making processes.

To decide how to divvy up the constituent actions needed to complete a task, agents must reason about the capabilities of other agents; for systems to be able to do so will undoubtedly require techniques for belief attribution and nonmonotonic reasoning. Agents also need means of balancing their own previous obligations—from either individual desires or other group plans—with the new obligations that agreeing to perform a constituent action for the group activity would entail. They need to reconcile intentions from private acts with those from group activities. Again, communication needs abound.

Another View of Collaborative Activity

Having completed the short overview of the modeling needs posed by collaboration, I want to turn to look briefly at another characterization of collaborative activity. Doing so will let us examine the extent to which various features of collaboration must be explicitly modeled.

Figure 9 lists Bratman's four criteria for shared cooperative activity (Bratman 1992). Only one of these criteria, commitment to the joint activity, was explicit in the framework I just laid out. The others are implicitly encoded in various constructs I've discussed (Grosz and Kraus 1995). Mutual responsiveness and mutual support can be derived from commitment to joint activity and the specification of what it means for one agent to intend-that another be able to do an action. Meshing subplans can be derived from ways in which agents come to agree on recipes in the context of intentions-that the full group succeed. The point of looking at collaboration from Bratman's perspective is not only good scientific hygiene (it's always good to cover the philosophical bases), but also because we can see that the criteria do not need to be explicit in a formalization.

Although explicit models may enable more flexibility, sometimes they're not needed. The map-learning result of Bender and Slonim (1994) I mentioned earlier provides an example of a problem domain in which collaboration can be "built-in" to an algorithm without explicit models of belief and intention. The model that Bender and Slonim examined is that of strongly connected, directed graphs with indistinguishable nodes; the edges out of each node are labeled only with respect to the node: they are numbered, but do not have names that continue along their length. Learning in this setting means being able to output an exact copy of the graph.

Now this model might not seem close to reality, but first-time visitors to Boston will tell you it's a good approximation to the challenge they've faced driving around, trying to figure out what's connected to what, or learning the street map. There are lots of one way streets and no street signs to speak of. Many of the intersections look the same: crowded with buildings, pedestrians, and cars. Learning the graph corresponds to learning all of the streets with no error; that is, the robot can remember and connect the intersections correctly.

The graphs at the bottom of figure 4 are fragments taken from a strongly connected graph that illustrate the kinds of settings in which a single robot could get confused and be unable to learn correctly. In particular, a single robot will not be able to distinguish between the different highlighted configurations. It won't know after three steps whether it's gotten back to the starting node as in the right graph or is headed off somewhere else as in the left one. A pebble might help a robot differentiate the two situations, but if the robot were in the situation portrayed on the left it would lose a pebble dropped at the top left node. In contrast, two robots can learn the graph structure, by using an algorithm that resembles in spirit the behavior of the tandem-running ants Wilson described. The second robot can periodically catch up to the first robot. Specifically, Bender and Slonim show that one robot cannot learn these graphs in polynomial time, even with a constant number of pebbles, and that two robots can learn in polynomial time with no pebbles needed. The main difference is that the second robot can move on its own (thereby catching up) whereas a pebble cannot. The second robot can collaborate, the pebble cannot.

It's interesting to look at the assumptions of the Bender and Slonim algorithm. Their two robots start together. They must be able to recognize one another; they can't be anonymous—they need to know each other. They also need to communicate, for instance by radio or by touching. Collaboration is built into the algorithm. The robots do not explicitly reason or negotiate; instead, the designers (or, in this case, the algorithm writers) directly encode collaboration.

SHARED COOPERATIVE ACTIVITY

- Mutual responsiveness
- Commitment to the joint activity
- Commitment to mutual support
- Meshing subplans

Figure 9. Shared Cooperative Activity.

Distinguishing Collaboration from Other Group Activity

We might stop now to ask whether the criteria we've given for collaboration really distinguish it from other group behaviors. Do they rule out anything? I will examine two other types of behavior to show they do. First, I'll look at the contrast between collaborating with someone and contracting to another agent. Second, I'll revisit, this time in more detail, the differences between collaboration and interaction, looking to see how the criteria we've developed distinguish between them. There are other kinds of behavior one might examine; for example, Bratman (1992) suggests we need to rule out coercion: if two people go to New York together as a result of one kidnapping the other, that's hardly collaborative. However, the formalization of individual will is more complex than the properties needed to distinguish contracting and interaction from collaboration; for now, I'll stick with cases we have some chance of understanding formally.

The two painting scenarios given in figure 10 lead to different answers to the question posed at the bottom of the figure. The contrast makes evident a key distinction between contracting and collaborating. In both scenarios, Leslie is going to do the prep work for the paint job. (She's the helper we'd all like around when we're painting.) She'll need to

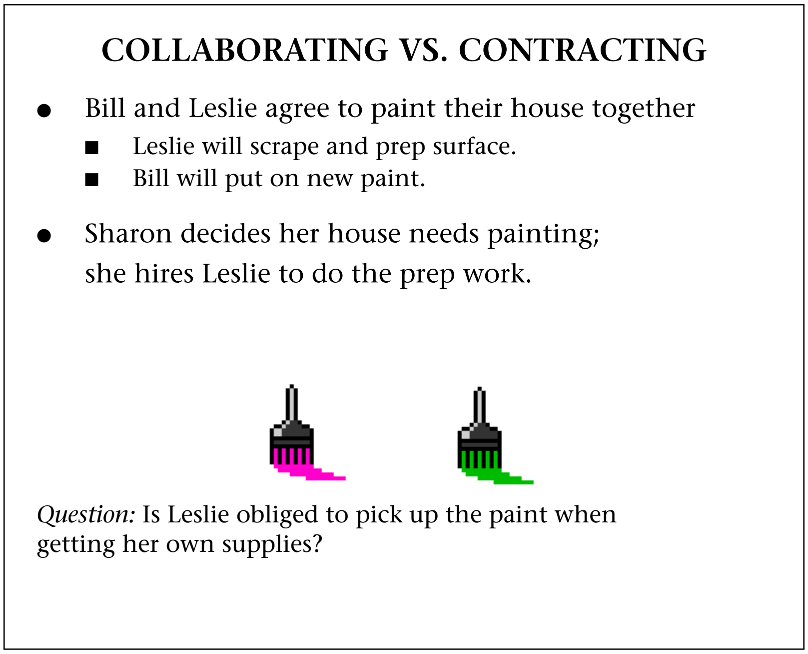


Figure 10. Collaborating versus Contracting.

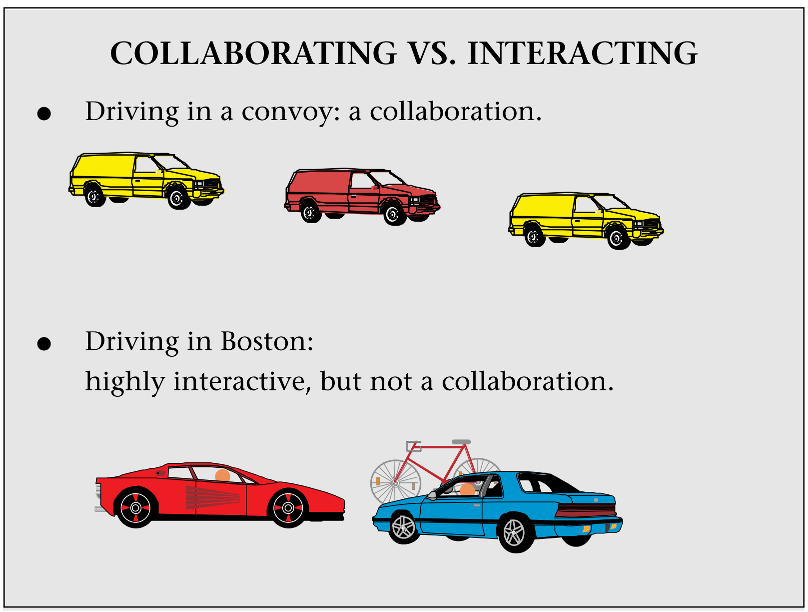


Figure 11. Collaborating versus Interacting.

acquire the supplies required and get them to the house. The question is whether there's any obligation for her to help the other painter; for example, while she's at the hardware store getting supplies for her part of the job, does she need to consider picking up the paint? In the contracting scenario in which Sharon has hired Leslie, there's no obligation whatsoever to help the other person; all Leslie is obliged to do is the prep work. In contrast, when Leslie is collaborating with Bill, her commitment to their joint activity and commitment to Bill's being able to do his part along with a recognition that Bill needs the paint from the hardware store oblige her at least to consider picking up the paint. She may in the end decide she cannot help Bill out (for example, if she has gone on her bike and is thus unable to transport the paint), but she must consider the tradeoff between doing so and not. That is, she needs to weigh her obligation to the group activity against other commitments in light of her abilities and make a decision.

To illustrate how the framework I've presented enables us to distinguish collaboration from interaction, I'll use the driving example in figure 11. Driving in a convoy is a paradigmatic example of joint action by a team, as Levesque, Cohen and Nunes present so clearly in their paper on Teamwork (Levesque, Cohen, and Nunes 1990). In contrast, ordinary driving is not—even if drivers act simultaneously, even if they follow certain traffic laws, even if the laws are those in a motor vehicle code and not just conventions that result from evolutionary choice as in Boston.

The contrast is most clearly seen by considering what happens when something goes wrong. Typically, convoy drivers agree on a recipe they will follow, one that includes checkpoints. They have a set of agreed-upon signals, or CB radios, or car phones, some method of communication they can use to help maintain mutual belief of how they are carrying out their joint activity. If one of them needs help, the others will assist. So we have commitment to the joint activity, an agreed-upon recipe, commitment to the success of others' actions, and a means of communication to establish and maintain the requisite mutual beliefs. In short, we have collaboration.

The scene in Boston is different. There's no common goal, although the drivers may have a goal in common, namely filling the blank space in front of them. (The bicyclist having seen this has wisely fled the scene.) An individual driver has no commitment to the success of actions of other drivers. There is no joint activity so no commitment to one; there's no agreed upon recipe; and there's no commitment to the ability of other agents to successfully carry out their actions. There may be communication between drivers, but it's not in service of any shared goal. In short, there's no collaboration.

Collaborating Computer Systems

I've now characterized the phenomena we need to model and described some of the research challenges they pose for AI. Before turning to research problems, I want to look at the ways in which the issues I've discussed arise in and affect settings in which several computer systems need to coordinate their activities. The Superhighway goal has led people to ask many questions about this setting recently, most of them about handshakes, protocols, and means of charging for services.

I'd like to ask a different kind of question: What capabilities could we provide if systems were to collaborate rather than merely interact? What difference would they make to system performance? To examine these issues, I will focus on the types of systems settings that much research in DAI has considered. Figure 12 illustrates a typical scenario.[8] To meet the needs of the user in the situation portrayed in this figure will take the coordinated efforts of three systems, one that has access to NASA images, one that has access to orbit information, and a compute server. Systems B and C will need to collaborate to determine the appropriate coordinates for the picture, and then will need to communicate the results to System A, which can access and display the appropriate pictures.

The advantages of collaboration, as opposed to mere interaction, again may be illustrated by considering what happens if something goes wrong. For example, if System B discovers network problems and it's collaborating with A, it might communicate this to A to save A wasted effort. If System A can't get to its information source, it needs to report back to the other systems, providing as much information as possible so the whole group can explore alternative recipes. If System C is unable to do a particular calculation, it needs to communicate with the other systems, again providing as much information as possible, so that the whole group can formulate a revised plan; just sending back a simple "can't compute" error message will not suffice. Collaboration can also play a role when no problems arise; for instance, Systems A and B might share rather than compete for network resources.

None of this is to say that systems must reason with explicit models of belief, intention and the like in such settings. Remember the ants. However we implement the ideas of commitment to joint activity and to the success of others as well as the ability to formulate a joint plan of attack, we need them in our systems.

COMPUTER AGENTS

- User's need:
  - Obtain photographs of Jupiter at anticipated impact sites of Shoemaker-Levy 9
  - Wants best grain-size, largest area available
- The team:
  - System A: network agent with access to NASA images
  - System B: access to information on orbits of Jupiter and Shoemaker-Levy 9
  - System C: compute server

Figure 12. Example 3: Computer Agents.

RESEARCH PROBLEM AREAS

- Recipe (plan) construction
- Multi-agent learning
- Agent-action assignment
- Modelling commitment
- Communication requirements, constraints, tradeoffs
- Negotiation
- Intention-conflict resolution

Figure 13. Research Problem Areas.

Research Problems

I want to turn now to ask what problems need to be solved for us to be able to construct systems that can collaborate with each other or with people or both. Figure 13 gives only a small sampling. The problems listed cross all areas of AI. I clearly won't be able to examine them all in detail and so will focus on two—negotiation and intention-conflict

AGENT CHARACTERISTICS:

- Individually motivated or 'benevolent'?
- Central design or different designers?
- Agents anonymous to one another?
- Can you break a promise or deceive?
- Any long-term commitments?
- Do agents have complete information?
- Limits on reasoning power?

Costs: communication, negotiation time

Figure 14. Agent Characteristics.

resolution. I want to use these problems to illustrate the difference between what we know how to do from work on individual, individually-motivated agents and what's needed for collaboration.

Negotiation

The term negotiation has been used within the DAI community to refer to many different types of activities. They may all have in common only one thing, the aspect of negotiation that's mentioned prominently in the dictionary; that is, "to confer with another so as to arrive at the settlement of some matter" (Mish 1988). The realization in their systems of "confer" and "arrive at settlement" are among the problems facing designers of systems.

Although some work has been done on systems that support negotiations among people (Sycara 1988), I will consider work that addresses how multiple computer agents can reach consensus or accommodate each other's needs when there are conflicts in their goals or among their resource needs.[9] Such systems will need abilities to communicate, resolve conflicts, and coordinate activities. Researchers working on these problems have considered agents with widely varying characteristics; some of the dimensions of agent characteristics are listed in figure 14.

The first dimension listed obviously affects all the others. One might ask under what conditions people or systems move from self-interest to concern with the good of the group, that is move to benevolence or collaboration. Huberman and his colleagues (Glance and Huberman 1994) have considered factors that contribute to cooperative behavior emerging (see also Axelrod [1984]). They have looked at situations in which people can choose between acting selfishly or cooperating for the common good and have asked how global cooperation can be obtained among individuals confronted with conflicting choices. They examine situations in which the individual rational strategy of weighing costs against benefits leads to all members defecting, no common good, and everyone being less well off. The situation changes, however, if the players know they will repeat the game with the same group. Each individual must consider the repercussions of a decision to cooperate or defect. The issue of expectations then comes to the fore. Individuals do not simply react to their perceptions of the world; they choose among alternatives based on their plans, goals and beliefs" (Glance and Huberman 1994, p. 78). Their modeling work shows that cooperative behavior arises spontaneously if groups are small and diverse and participants have long-term concerns. If we are going to build systems that will interact on more than one-shot deals, it might be wise to build them with capabilities for collaboration.

The other dimensions listed in the figure allow for variations in how much control is exerted both over the design of individual agents and over the social organization of the agents. A significant question is when to design agents with built-in cooperation mechanisms (Durfee and Lesser 1989), thus making them essentially benevolent. Other dimensions of variability concern whether agents have any history of interactions or memory of one another. Breaking a promise matters much more if your identity is known and you'll meet the person again. How much agents know about one another, and in particular about each other's preferences, also affect negotiation strategies.

Much of the work in this area has drawn on research in game theory, but has extended it to consider questions of how one designs agents to behave in certain ways, and what kinds of performance different strategies will deliver. A range of results have been obtained. For instance Rosenschein and Zlotkin (1994) have specified attributes of problem domains that affect the choice of an appropriate negotiation strategy and have shown that the domain as well as the deal-type affects whether or not agents have any incentive to lie. Kraus, Wilkenfeld, and Zlotkin (1994) have derived strategies that take into account the time spent in negotiating. Alas, as the U.S. Congress has proved repeatedly, merely stalling is a great strategy for keeping the status quo.

In collaborative settings, negotiation has different parameters and raises additional issues. There is a shared goal and typically at least some history: agents collaborate over the full time period required to complete their joint activity. Agents may also collaborate more than once. How does this affect inclinations to tell the truth, intention-reconciliation decisions, and commitment to promises (for example, commitments to completing subtasks)? In reconciling individual needs with those of the group, agents need to weigh the importance to them of the group succeeding against other demands. How might we take this into account? How do individual goals interact with shared objectives? There is some game theory literature (Aumann and Maschler 1995) that deals with repetitive interactions: the success of the "tit for tat" strategy in playing the iterated prisoner's dilemma is one example. But this work too, at least so far as I have been able to determine, focuses on self-motivated agents, and does not account for the role of joint commitment to some shared goal in determining choices.

Finally, although protocol design can help ensure certain behaviors among computer systems, any negotiation that includes human participants will need to accommodate to human preferences and styles. Systems that collaborate with humans may have to cope with the nonrational as well as the rational, as anyone who's ever been at a faculty meeting will know.

Intention-Conflict Resolution

Intention-conflict resolution is an inherent problem given agents with limited resources. Each of the collaborating agents has multiple desires not all of which can be satisfied. It will have plans for actions it is carrying out to meet some desires, plans that include various intentions. It must ensure that its intentions don't conflict (a property that distinguishes intentions from desires). In considering whether it will do something, an agent must weigh current intentions with potential intentions. There are significant computational resource constraints. As those who have worked on real-time decision making know, all the while the world is changing—agents can't think forever. They must choose what to think about and whether to stop thinking and start acting. That heart-attack patient would surely prefer the doctors to start working on a good plan rather than waiting until they've proved they've got the best plan. As Voltaire would have it, "the best is the enemy of the good." Collaborating agents need to reconcile competing intentions while respecting time constraints.

The planning community has been actively considering this question in the context of a single agent. Three of the major approaches they've taken are to design techniques for decision compilation into finite state machines (Brooks 1991; Kaelbling and Rosenschein 1991), to design deliberation scheduling algorithms (Boddy and Dean 1989; Horvitz 1988), and to develop architectures that include rational heuristic policies for managing deliberation (Pollack and Ringuette 1990).

Decision compilation techniques have been used mostly for simple agents, although as we saw with the ants, simple agents can sometimes do complex things. Deliberation scheduling algorithms have been applied both to traditional object-level algorithms and to anytime algorithms. Both on-line preference computations and off-line ones (compiling decision-theoretic utilities into performance profiles) have been investigated. In contrast with this decision-theoretic scheduling approach, the work on architectures has explored the development of satisficing (heuristic) policies that can be subjected to experimental investigation.

These approaches provide general frameworks for managing reasoning in dynamic environments, but none of them have yet been applied to the problem of intention-conflict resolution in multi-agent, collaborative settings. Again, collaboration changes the parameters of the model; it affects the questions we ask and the kinds of answers we need to find. For example, we need methods for balancing individual obligations with group obligations and for weighing the costs of helping other agents with the benefits; remember Leslie's dilemma in the house-painting example. Agents need to be able to determine when it's best to ask others in the group for help. Finally, we need techniques that apply in states of partial knowledge, for example agents typically will not know all the utility functions a priori.

Research Collaborations

In using the word "systems" in the title, I intended to pun. In this last section, I want to shift to another meaning. "Systems" can refer not only to the systems we build, but also to the systems that those systems are part of, the network in which the users of computer systems—including us, as researchers—and the systems themselves participate. So, I want to shift from collaboration as an object of study to collaboration as something we do.

I hope the need for collaborative efforts across AI research areas is evident from the problems I've discussed. Not only are such efforts needed to solve these problems, but also collaboration provides a good testbed for research in many of the areas I've mentioned: nonmonotonic reasoning, planning, reasoning under uncertainty, and reasoning about beliefs as well as natural language, vision, and robotics. The 1993 Fall Symposium on Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice (Terveen 1995) brought together researchers from several different areas. As the title suggests, the symposium emphasized those situations in which the collaboration is between a human and a computer. In the future, I hope to see many more symposia and workshops focused on particular technical issues that cross areas.

A major lesson of the 1980s was that AI could not stand alone. Rather, AI capabilities need to be designed as parts of systems built collaboratively with others.

I want now to mention another kind of collaboration within the field, that of working together to define new directions and to explain to those outside the field what it is all about. In the spring of 1994, following a suggestion of some program managers at ARPA,[10] AAAI organized a small workshop to define an agenda for foundational AI research at ARPA. Just prior to AAAI-94, at the suggestion of various National Science Foundation division and program directors, we held a somewhat larger workshop to discuss and delineate the ways in which AI could contribute to the National Information Infrastructure and in particular to the Information Infrastructure Technology and Applications program within it.

Each workshop brought together people from across the spectrum of AI research and applications. The participants were asked to become familiar enough with work outside their own individual research interests to be able to explain it to funders and to justify funding of work on key problems. Participants worked for the common good, even if their individual cost was higher. People had to invest time to learn about areas in which they were not already expert. I was struck by the enthusiasm everyone exhibited, and by the perseverance with which they stuck to the task. I expected our job to be hard, but it turned out to be even harder than I'd thought. We all learned a lot. I hope, and have reason to expect, that the reports (Weld 1995; Grosz and Davis 1994) will benefit the field. Given the current climate for research funding, I expect AAAI will be called on to do more of this, and AAAI will in turn need to call on you, our members, to help.

So, we've come full circle: we need to do what we study.

Let me return now to research issues and consider the need for collaboration with other areas of computer science. A major lesson of the 1980s was that AI could not stand alone. Rather, AI capabilities need to be designed as parts of systems built collaboratively with others. We need to work with people in other areas of computer science, getting them to build the kinds of platforms we need and working together to integrate AI capabilities into the full range of computer systems that they design. This is especially true as we turn to work on large networks of interconnected machines with different architectures. It should be obvious that collaboration with other computer scientists will be needed to build collaborative systems; these systems will use networks and access large data bases; they'll depend on high-performance operating systems. The development and testing of experimental systems will surely be a cooperative endeavor.

At the research level as well, there are overlapping interests, questions in common and benefits to be gained from coordinated research. For example, research in distributed systems asks many questions about coordination and conflict avoidance that are similar to those asked in DAI. Typically, this work has been able to constrain the environments in which it is applied so that simpler solutions are possible. For example, communication protocols can often depend on a small set of simple signals. Two-way cross fertilization can be expected: AI researchers can examine the extent to which the protocols and solutions developed for distributed systems can be adapted to less restricted environments; conversely, by identifying limitations to these approaches and ways to overcome them, we will propose new solutions that may lead to more powerful distributed computer systems.

The July 1994 issue of the *Communications of the ACM* was a special issue on intelligent agents. It included articles from a range of computer science fields, including several on AI projects. I was quite pleased to see AI mixed in with all the rest. However, what struck me most was how much the rest of computer science could benefit from knowing what we know how to do. In particular, I was surprised by the (non-AI) articles that still view a user's job as programming, albeit with more and more user-friendly languages. Why not have an intelligent agent that's really intelligent? Why not build an agent that considers what the user is trying to do and what the user needs, rather than demanding to be told how to do what the user needs done, or a system that learns. In her article, Irene Greif from Lotus claims that the "next wave of innovation in work-group computing [will result in] products that encourage collaboration in the application domain" (Grief 1994). If so, the work we've done in various fields—natural-language processing, representation and reasoning to name a few—could help save development time and effort and make better products. But for this to happen, we in AI have to work on the problems of modeling collaboration.

In this article, I have not discussed the question of computer systems that assist people in collaborating or groupware. That, of course, is the focus of those people in the field of `computer supported cooperative work.' Much of the work in that field has focused on current applications. The kind of foundational work I've proposed we take up could provide a solid theoretical, computational foundation for work in this area and the basis for significant increase in the capabilities of such systems.

Finally, many of the problems I've described will benefit from interdisciplinary work. Since the 1970s, various areas of AI research have drawn on work in psychology and linguistics, and we can expect these interdisciplinary efforts to continue. More recently, DAI and planning research has drawn on game theory and decision theory. As we aim to understand collaboration and to build systems that work together in groups, we will need also to consider work in those social sciences that study group behavior (such as anthropology and sociology) and to look into mathematical modeling of group processes.

Conclusion

As the acknowledgments make evident, in preparing the Presidential Address and this article, I have had the help of many people. Our interactions were often collaborations. Each of my collaborators, in this work and in my research, has shown me the benefits of collaboration, and my experience collaborating makes me sure that I'd rather have a computer that collaborated than one that was merely a tool. It's time we generalized John Donne's claim (figure 15). Designing systems to collaborate with us will make a difference to AI; it will make a difference to computer science, and, by enabling qualitatively different kinds of systems to be built, it will make a difference to the general population. At the very least, it will lead to a decrease in the number of times people say "stupid computer." Besides, working on collaboration is fun. I hope more of you will join in the game.

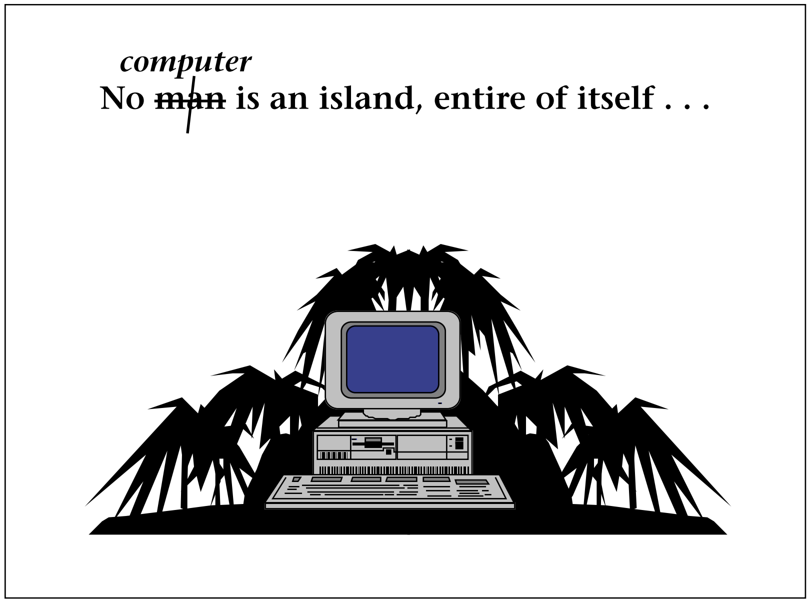


Figure 15. No Man Is an Island, Entire of Itself….

Acknowledgments

Candy Sidner, who was a partner in my earliest explorations of collaboration, and Sarit Kraus, who has been a partner in extending and formalizing this early work, have been the best of research collaborators. This article, and the talk on which it is based, simply would not have been possible without them. Karen Lochbaum applied our model of collaboration to discourse processing and in doing so asked critical questions, pointed out flaws and gaps, and helped improve the formalization. Michael Bender, Andy Kehler, Steven Ketchpel, Christine Nakatani, Dena Weinstein, and the other students in my graduate planning course have taught me a lot by the questions they've asked. Ray Perrault, Martha Pollack, Jane Robinson, Jeff Rosenschein, Stuart Shieber, and Loren Terveen provided helpful comments and advice. The participants in the AAAI/ARPA and AAAI/NSF workshops taught me a lot about current research in areas of AI outside my normal purview. Dan Bobrow and Pat Hayes convinced me I'd survive being President of AAAI and giving a Presidential Address. I thank them for their advice and support. My youngest collaborator, Shira Fischer, prepared the slides for the talk, some of which have become figures in this paper, under tight time constraints. She likes pointing, clicking, and clip art much more than I do. My most longstanding collaborator, Dick Gross, not only joined me in escaping playpens, but also taught me enough about medicine that I could construct a reasonable health care example.

Notes

1. This is just a hunch, one I won't try to prove. However, the fact that a team of three robots from Georgia Tech won the office-cleanup part of the 1994 Robot Competition (Balch et al. 1995) suggests my hunch may be right.

2. A crew tromped through my kitchen and overtook the honey pot while I was preparing this Presidential Address.

3. This example was chosen for its linguistic, not its practical, features. Alas, I don't think paper-writing assistance is an application we should aim for anytime soon.

4. I'm a long-time fan of collaboration. My first research collaboration was while I was still a graduate student: the speech group in the AI Center at SRI functioned as a team. However, my affection for collaboration precedes these research collaborations by several decades. It seems to have taken root when my twin brother and I discovered that if we put our heads together and formulated a joint plan, we could get out of play pens, apartments, and back yards. I better not say anything about the trouble we also got *into* this way.

5. Various philosophers (Vermazen 1993; Bratman 1992) have also argued for types of intentions other than intending to do an action.

6. Some formalizations of collaboration (Levesque, Cohen, and Nunes 1990) force an agent to communicate when it drops an intention related to the group activity; for example, the formalization explicitly encodes an obligation to communicate if something goes wrong. However, in certain systems settings, it may be more efficient for other agents to detect a collaborator's dropped intention. Blackwell et al. (1994) take this approach in a mobile computing application. The extremely dynamic nature of this joint activity led the system designers to place the burden for maintaining mutual belief about commitment to the joint activity on the host (requiring it to check for the mobile system) rather than on the mobile system (requiring it to report a change).

7. An important open question is how agents can detect resource conflicts given incomplete knowledge about each other's recipes (Grosz and Kraus 1995).

8. I thank Steven Ketchpel for suggesting this example.

9. Another approach to multi-agent coordination bypasses the need for negotiation by designing agents to follow social laws (Shoham and Tennenholtz 1995). This kind of approach entails more effort at design time, but less run-time overhead; however, it requires either centralized control of design or collaborative designers.

10. The Department of Defense Advanced Research Projects Agency; the acronym has recently reverted to DARPA.

References

Aumann, R. J., and Maschler, M. B. 1995. *Repeated Games with Incomplete Information*. Cambridge, Mass.: MIT Press.

Axelrod, R.M. 1984. *The Evolution of Cooperation*. New York: Basic Books.

Balch, T.; Boone, G.; Collins, T.; Forbes, H.; MacKenzie, D.; and Santama´ria. IO, GANYMEDE, and CALLISTO: A Multiagent Robot Trash-Collecting Team. *AI Magazine* 16(2): 39–52.

Bender, M. A., and Slonim, D. K. 1994. The Power of Team Exploration: Two Robots Can Learn Unlabeled Directed Graphs. In *Proceedings of the Thirty-Fifth Annual Symposium on Foundations of Computer Science*. Los Alamitos, Calif.: IEEE Computer Society Press.

Blackwell, T.; Chan, Kee; Chang, Koling; Charuhas, T.; Gwertzman, J.; Karp, B.; Kung, H. T.; Li, W. D.; Lin, Dong; Morris, R.; Polansky, R.; Tang, D.; Young, C.; and Zao, J. 1994. Secure Short-Cut Routing for Mobile IP. In Proceedings of the USENIX Summer 1994 Technical Conference. June.

Bobrow, D. 1991. Dimensions of Interaction: AAAI-90 Presidential Address. *AI Magazine* 12(3): 64–80.

Boddy, D. M., and Dean, T. 1989. Solving Time-Dependent Planning Problems. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 979–984. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Bratman, M. 1992. Shared Cooperative Activity. *Philosophical Review* 101:327–341.

Bratman, M. 1987. *Intention, Plans, and Practical Reason*. Cambridge, Mass.: Harvard University Press.

Brooks, R. 1991. Intelligence without Representation. *Artificial Intelligence* 47:139–159.

Davis, R., and Smith, R. 1983. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence* 20:63–109.

Davis, R., and Smith, R. 1983. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence* 20:63–109.

Durfee, E. H., and Lesser, V. R. 1989. Negotiating Task Decomposition and Allocation Using Partial Global Planning. In *Distributed Artificial Intelligence, Volume 2,* eds. L. Gasser and M. N. Huhns, 229–244. San Francisco, Calif.: Morgan Kaufmann.

Etzioni, O. 1993. Intelligence without Robots (A

Reply to Brooks). *AI Magazine* 14(4): 7–13.

Etzioni, O., and Weld, D. 1994. A Softbot-Based Interface to the Internet. *Communications of the ACM* 37(7): 72–76.

Glance, N. S., and Huberman, B. A. 1994. The Dynamics of Social Dilemmas. *Scientific American,* 270(3): March, 76–81.

Greif, I. 1994. Desktop Agents in Group-Enabled Products. *Communications of the ACM* 37(7): 100–105.

Gringoire, T., and Saulnier, L. 1914. *Le Re'pertoire de la Cuisine*. Paris: Dupont et Malgat-Gu'ering, 81.

Grosz, B., and Davis, R., eds. 1994. A Report to ARPA on Twenty-First–Century Intelligent Systems. *AI Magazine* 15(3): 10–20.

Grosz, B., and Kraus, S. 1995. Collaborative Plans for Complex Group Action. *Artificial Intelligence* 86(1).

Grosz, B., and Sidner, C. 1990. Plans for Discourse. In *Intentions in Communication,* 417–444. Cambridge, Mass.: Bradford Books.

Horvitz, E. J. 1988. Reasoning under Varying and Uncertain Resource Constraints. In Proceedings of the Seventh National Conference on Artificial Intelligence, 111–116. Menlo Park, Calif.: American Association for Artificial Intelligence.

Kaelbling, L. P., and Rosenschein, S. J. 1991. Action and Planning in Embedded Agents. *Robotics and Autonomous Systems* 6(1).

Kinny, D.; Ljungberg, M.; Rao, A. S.; Sonenberg, E. A.; Tidhar, G.; and Werner, E. 1994. Planned Team Activity. In *Artificial Social Systems: Lecture Notes in Artificial Intelligence,* eds. C. Castelfranchi and E. Werner. Amsterdam: Springer Verlag.

Kosecka, J., and Bajcsy, R. 1993. Cooperation of Visually Guided Behaviors. In Proceedings of the International Conference on Computer Vision (ICCV-93). Los Alamitos, Calif.: IEEE Computer Society Press.

Kraus, S.; Wilkenfeld, J.; and Zlotkin, G. 1994. Multiagent Negotiation under Time Constraints. *Artificial Intelligence*. 75(2): 297–345.

Levesque, H.; Cohen, P.; and Nunes, J. 1990. On Acting Together. In Proceedings of the Eighth National Conference on Artificial Intelligence, 94–99. Menlo Park, Calif.: American Association for Artificial Intelligence.

Lochbaum, K. 1994. Using Collaborative Plans to Model the Intentional Structure of Discourse, Technical Report, TR-25-94, Harvard University.

Maes, P. 1994. Agents That Reduce Work and Information Overload. *Communications of the ACM* 37(7): 31–40.

Mish, F. C., ed.1988.*Webster's Ninth New Collegiate Dictionary*. Springfield, Mass.: Mirriam-Webster.

Newell, A. 1981. The Knowledge Level: Presidential Address. *AI Magazine* 2(2): 1–20, 33.

Pollack, M. E. 1990. Plans as Complex Mental Attitudes. In *Intentions in Communication,* eds. P. N. Cohen, J. L. Morgan, and M. E. Pollack. Cambridge, Mass.: MIT Press.

Pollack, M. E., and Ringuette, M. 1990. Introducing the TILEWORLD: Experimentally Evaluating Agent Architectures. In Proceedings of the Eighth National Conference on Artificial Intelligence, 183–189. Menlo Park, Calif.: American Association for Artificial Intelligence.

Prince, E. F. 1981. Toward a Taxonomy of Given-New Information. In *Radical Pragmatics,* ed. Peter Cole, 223–255. San Diego, Calif.: Academic.

Putzel, M. 1994. Roadblocks on Highway: Getting on the Internet Can Be a Real Trip. *The Boston Globe,* 22 July 22, p. 43.

Rombauer, I. S. and Becker, M. R. . 1931. The Joy of Cooking. St. Louis, Mo.: A. C. Clayton Printing Company, 408.

Rosenschein, J., and Zlotkin, G. 1994. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. Cambridge, Mass.: The MIT Press.

Shoham, Y., and Tennenholtz, M. 1995. On Social Laws for Artificial Agent Societies: Off-Line Design. *Artificial Intelligence* 73:231–252.

Sycara, K. P. 1988. Resolving Goal Conflicts via Negotiation. In Proceedings of the Seventh National Conference on Artificial Intelligence, 245–250. Menlo Park, Calif.: American Association for Artificial Intelligence.

Terveen, L., ed. 1995. *Knowledge-Based Systems* (Special Issue on Human-Computer Collaboration) 8(2–3).

Vermazen, B. 1993. Objects of Intention. *Philosophical Studies* 71:223–265.

Weld, D. 1995. The Role of Intelligent Systems in the National Information Infrastructure. *AI Magazine* 16(3): 45–64.

Wilson, E. 1971. *The Insect Societies*. Cambridge, Mass.: Belknap Press/Harvard University Press.

Barbara J. Grosz is Gordon McKay Professor of Computer Science at Harvard University. Professor Grosz pioneered research in computational models of discourse (including models of focusing of attention, reference, and discourse structure). She has also published papers on collaborative planning and on natural-language interfaces to databases. Her current research interests include the development of models of collaborative planning for joint human-computer problem-solving, coordinating natural language and graphics for human-computer communication, and determining the effects of discourse structure on intonation. A Fellow of the American Association for the Advancement of Science and the American Association for Artificial Intelligence (AAAI), she is also the Past-President of the AAAI, a Member and former Chair of the Board of Trustees of the International Joint Conferences on Artificial Intelligence, Inc., and a member of the Computer Science and Telecommunications Board of the National Research Council.