Socially Situated Planning

Jonathan Gratch

University of Southern California Information Sciences Institute
4676 Admiralty Way, Suite 1000, Marina del Rey, CA 90292
gratch@isi.edu

Introduction

Virtual environments such as training simulators and video games do an impressive job at modeling the physical dynamics of synthetic worlds but fall short when modeling the social dynamics of anything but the most impoverished human encounters. Yet the social dimension is at least as important as good graphics for creating an engaging game or effective training tool. Commercial flight simulators accurately model the technical aspects of flight but many aviation disasters arise from social breakdowns: poor management skills in the cockpit, or the effects of stress and emotion. Perhaps the biggest consumer of simulation technology, the U.S. military, identifies unrealistic human and organizational behavior as a major limitation of existing simulation technology (NRC, 1998). And of course the entertainment industry has long recognized the importance of good character, emotional attachment and rich social interactions to "put butts in seats."

This article describes a research effort to endow virtual training environments with richer models of social behavior. We have been developing autonomous and semi-autonomous software agents that plan and act while situated in a social network of other entities, human and synthetic (Hill et. al, 1997; Tambe, 1997; Gratch and Hill, 1999). My work has focused on making agents act in an organization and obey social constraints, coordinate their behavior, negotiate conflicts, but also obey their own self-interest and show a range of individual differences in their behavior and willingness to violate social norms, albeit within the relatively narrow context of a specific training exercise.

Socially-Situated Planning

There are many approaches to modeling social behavior. My approach, which I term *socially-situated planning*, draws inspiration from the shared-plans work of Grosz and Kraus (1996), but has some important differences. Socially-situated planning adds a layer of social reasoning atop a general purpose planning system such as an artificial intelligence planning system. The planning handles task-level behaviors whereas the social layer manages communication and biases plan generation and execution in accordance with the social context (as assessed within this social layer). In this sense, social reasoning is formalized as a form of meta-reasoning.

Social Assessment: To support a variety of social interactions, the social reasoning layer must provide a rich model of the social context. The social situation is described in terms of a number of static and dynamic features from a particular agent's perspective. Static features include innate properties of the character being modeled (social role and a small set of "personality" variables). Dynamic features are derived from a set of domain-independent inference procedures that operate on the current mental state of the agent. These include the set of current communicative obligations, a variety of relations between the plans in memory (your plans threaten my plans), and a model of the emotional state of the agent (important for its communicative role).

Planning: One novel aspect of this work is how the social layer alters the planning process. Grosz and Kraus show how meta-level constructs like social commitments can act as constraints that limit the planning process in support of collaboration (for example, by preventing a planner from unilaterally altering an agreed upon joint plan). I go beyond this to show how to model a variety of "social stances" one can take towards other individuals based on one's role in an organization and other dispositional factors. Thus, the social layer can bias planning to be more or less considerate to the goals of other participants, act to boss agents around, or to meekly avoid conflicts.

Communication: Another key aspect of social reasoning is the ability to communicate socially appropriate information to other agents in the virtual environment. As with many approaches to social reasoning, the social layer provides a set of speech acts that an agent can use to convey or request information. Just as plan generation should differ depending on the social situation, the use of speech acts must be similarly biased. A commanding officer in a military operation would communicate differently and under different contexts than her subordinates.

Social Control Programs: Rather than attempting to formalize some generic rules of social behavior, I've adopted the approach of providing what is essentially a programming language for encoding the reasoning of the social layer. This language provides a set of inference procedures and data structures for representing an agent's social state, and it provides a set of control primitives that initiate communicative acts and alter the behavior of the task-level planning system. A simulation developer has a great deal of latitude in how they write "social control programs" that inform an agent's social-level reasoning. The strong constraint imposed by this language is that social reasoning is forced to operate at a meta-level. The control primitives treat plans as an indivisible unit. An agent can have multiple plans "in mind" and these can be communicated and treated differently by the planner, but the social-layer cannot manipulate or refer to the contents of these plans directly. This concept will be made clearer in the discussion below. These social control programs can be viewed as defining a finite state machine that changes the state of the set of control primitives based on features of the social context. In the examples in this article this state machine is defined in terms of a set of condition action rules, although in one application these state transitions have been formalized in terms of STRIPS-style planning operators and the social-program actually synthesized by the planning system (Gratch and Hill, 1999).

Illustration

This approach has been used to model the behavior of military organizations in the context of a training system (Gratch and Hill, 1999) but the following contrived example provides a clear view of the capabilities of the system. In this example, two synthetic characters, Jack and Steve, interact with each other in the service of their own conflicting goals. The interaction is determined dynamically as the agents interact with each other, but is also informed by static information such as the social stance they take towards one another.

These agents are embodied in a distributed virtual environment developed by Rickel and Johnson (1999) that provides a set of perceptual, communicative and motor processes as well as a model of cognition. In this example, I've replaced their model of cognition with my situated-planning approach. The agents share certain task level knowledge encoded as STRIPS-style operators. They know how to drive vehicles to different locations, how to surf, and how to buy lottery tickets. They also have individual differences. They have differing goals, have varying social status and view their relationship with each other somewhat differently.

Jack's goal is to make money, he views Steve as a friend, and treats him fairly. Steve wants to surf, views Jack as a friend, but is rude. All of these terms have a specific technical definition discussed below. Both agents develop different plans but have to contend with a shared resource. Besides performing task level actions, the agents engage in speech acts and generate gestures, facial expressions, and affective speech modulation based on properties of their social state.

What follows are annotated traces of two separate runs of the system where the only difference is a change in the "personality" of the Steve agent. In the first trace he treats Jack rudely, in the second he treats him fairly. The text generation is extremely simplistic and agents actually communicate with each other through a more stylized plan-communication language.

Rude Interaction:

Jack: I want to make-some-big-money. *[Creates a new plan containing this goal. Looks concerned, scratches his head, then, after devising a plan looks hopeful.]*

Steve: I want to catch-some-waves. *[Creates a new plan containing this goal. Looks concerned, scratches head, and continues to look concerned. Surfing is important to Steve and he cannot devise a plan.]*

Jack: [*Perceives Steve's emotional expression and generates an information request.]* Hey Steve, what's wrong?

Steve: *[Locates the appraisal generating the most intense negative emotional excitation. Communicates the associated plan in a distressed tone of voice.]* I want to catch some waves but can't find any good breakers.

Jack: *[Adds Steve's plat into plan memory and locates relevant information. Jack was previously told of a "plan" that establishes Steve's blocked subgoal]* Steve, does it help that someone did say there's some great waves near the pier?

Steve: *[Incorporates the communicated plan fragment. Completes a plan to go surfing and looks hopeful.]*

Jack: *[Perceives Steve's change in expression and seeks to confirm his expectation that the information he provided helped Steve.]* So that information helped?

Steve: *[Handles Jack's request.]* Yes Jack. I plan to drive the car to the beach, then I plan to surf-my-brains-out.

Jack: *[Incorporates Steve's revised plan and finds a conflict with his own plans. Based on personality, Jack attempts to negotiate a fair solution.]* Wait a second. Our plans conflict. I plan to drive the car to the quicky-mart then I plan to buy a-lottery-ticket.

Steve: *[Incorporates Jack's plan and recognizes the same interaction. Based on personality model, Steve responds to interaction differently. He devises a plan that satisfies his own goals without regard to any conflicts it may introduce in Jack's plans. Steve exits stage right.]* Later dude, I'm driving the car to the beach.

Jack: *[Perceives that car has departed without him. Looks angry. Says in angry voice:]* I want to kill-my-roommate.

Cooperative Interaction:

Jack: *[Incorporates Steve's revised plan and finds a conflict with his own plans. Based on personality, Jack attempts to negotiate a fair solution.]* Wait a second. Our plans conflict. I plan to drive the car to the-quicky-mart then I plan to buy a-lottery-ticket.

Steve: *[Incorporates Jack's plan and recognizes the same interaction. Based on Steve having somewhat lower social status, he takes the initiative in repairing the conflict.]* Well, I could change my plans. *[Looks concerned, scratches head, then devises a possible joint plan.]* I have a suggestion. Could you drive the car to the-quicky-mart with-me then I could drive the car to the beach. *[Note that neither agent has been given the goal of returning home.]*

Jack: *[Incorporates Steve's suggested joint plan, determines that it is consistent with his own plans, and agrees to form a joint commitment to the shared plan.]* Sounds good to me.

Social Control Programs

A small change in an agent's static social state can result in a dramatic change in behavior because reasoning at the social level is highly leveraged. Social reasoning is conditioned on dynamic social features that encapsulate a good deal of domain-independent inference and social control primitives allow for considerable differences in how plans are generated and executed at the base level. Social reasoning is represented as a set of condition actions rules that operate at this meta-layer. Social state components serve as the conditions for these social rules whereas control primitives define the space of possible actions.

1.1 Social State

An agent's social state is divided into dynamic and static components. Dynamic components are further divided into communicative state, plan state, and emotional state.

Communicative State: The communicative state tracks what information has been communicated to different agents and maintains any communicative obligations that arise from speech acts. When Steve communicates a plan to Jack, Steve's social layer records that Jack knows this plan, and persists in knowing it until Steve's planning layer modifies it, at which point Steve's social layer records that Jack's knowledge is out of date. If Jack requests Steve's current plans, the social layer creates communicative obligations: the fact that Steve owes Jack a response is recorded in each agent's social layer (though whether Steve satisfies this obligation is up to Steve's social control program).

Plan State: At the base-level planning layer, all activities that an agent is aware of (whether they come from its own planning or are communicated from outside) are stored in a single plan network, allowing the planner to reason about the interrelationship between these activities. The social layer keeps track of the fact that different subsets of this plan network correspond to different plans – some belonging to the agent and some corresponding to (what the agent believes to be) plans of other agents. The social layer also computes a variety of high-level relations between plans. Plans can contain threats and the plans of one agent can introduce threats or be threatened by the plans of another agent (such relations are computed using the basic plan-evaluation routines provided by standard planning systems). Plans of one agent can also be relevant to the plans of other agents (as computed by the plan-relevance criteria proposed by desJardins and Wolverton, 1998). Plans may be interdependent in the sense that one depends on effects produced by another.

Emotional State: The social layer incorporates a model of emotional reasoning, Émile, that derives an emotional state from syntactic properties of an agent's plans in memory (Gratch, 2000). Émile adopts the cognitive view of emotions as a form of plan evaluation, relating events to an agent's current goals (c.f., Ortony et al, 1988; Lazarus, 1991). Émile can compute an agent's overall state, track the emotions arising from a specific plan, and make inferences about the emotional state of other agents (given an understanding of their goals and plans). Emotional state is represented as a real-valued vector representing the intensities of different emotional states (Fear, Joy, etc.) and Émile dynamically modifies this state based on the current world situation and the state of plans in memory.

Static State: Static social state components describe characteristics of an agent that are invariant in the course of a simulation. These components can be rather arbitrary and act simply as conditions to be tested by the social control program. The "personality gui" mentioned above implements five static state components. One can manipulate an agent's top level goals, its social status, its etiquette (its sensitivity to certain social cues), its independence (is it willing to construct plans that depend on the activities of other agents), and characteristics of its relationship with other agents (friendly, adversarial, rude, deferential, etc.).

1.2 Control Primitives

Control primitives define the set of actions one can associate with the social level. These primitives are sub-divided into communicative primitives and plan-control primitives.

Communicative Primitives: The social layer defines a set of speech acts that an agent may use to communicate with other agents. As they are defined at the meta-level, they can operate on plans only as an atomic structure and cannot make reference to components of a plan (although one has the option of breaking a plan into explicit subplans). Some speech acts serve to communicate plans (one can INFORM another agent of one plans, REQUEST that they accept some plan of activity, etc.). Other speech acts serve to change the state of some previously communicated plan (one can state that some plan is under revision, that a plan is acceptable, that it should be forgotten, etc.).

Planning Primitives: Planning primitives alter the way the planner treats activities within a base-level plan. Classical planning algorithms can be viewed as a sequential decision process: some critiquing routines identify a set of problems with the current plan network and propose a set of modifications that resolve at least one of these problems (an action should be added, these actions should be reordered, etc.); one modification is applied and the process continues. Planning primitives act by constraining the set of viable modifications.

Recall that from the perspective of the planning algorithm, all activities are represented in a single task network (whether they belong to the agent or represent the activities of other entities). One set of planning primitives allows one to create and manipulate plan objects. Plans can be created and destroyed, and they can be populated with new goals and with activities communicated by other agents.

Another set of planning primitives determines whether the planning algorithm can modify the activities in one of these plan objects. One can make a plan modifiable, allowing the planner to fix any flaws with that plan, or one can freeze its current state (as when adopting a commitment to a certain course of action. One can also modify the execution status of the plan, enabling or disabling the execution of actions within it.

Finally, another set of planning primitives alters the way the planner handles interactions between plans and thereby implements the idea of a social stance. For example, what happens when Steve detects that his plan conflicts with Jack's. He has several options. He could adopt a rude stance towards Jack, running to grab the keys before Jack gets a chance to take the car. This essentially corresponds to a strategy where the planner resolves any threats that Jack introduces into Steve's plans, but ignores any threats that Steve introduces into Jack's. Alternatively, Steve could take a meek stance, finding some other ways to get to the beach or simply staying home. This corresponds to a strategy where the planner treats Jack's plans as immutable, resolves any threats to Jack's plans, and tries to work around any threats that Jack introduces into Steve's plans. Steve could be helpful, adding activities to his plan that ensures that Jack gets to the market. Or he could be authoritative, demanding that Jack drive him to the beach (by inserting activities into Jack's plans). These stances are all implemented as search control, limiting certain of a planner's threat resolution options.

The following are a few paraphrased examples of rules that make up Steve and Jack's social control program. The current implementation has about thirty such rules:

Social-Rule: plan-for-goal
IF I have a top-level goal, ?goal, who's predicate is ?p THEN
Do-Gesture(Thinking)
Say(to-self, "I want to ?predicate")
?plan = create-new-plan()
populate-plan(?plan, ?goal)
enable-modification(?plan)

Social-Rule: help-friend
IF I have a plan, ?plan, that is relevant-to the plan of another agent, ?agent, I am friends with that agent, I am socially adept, and the plan is not already known to that agent THEN
Do-Gesture(Look-at ?agent)
SpeechAct(FACILLITATE, ?plan ?agent)

Social-Rule: you-cause-problems-for-me
IF my plan, ?plan, is threatened by your plan and I don't have an obligation to revise my plan and you don't have an obligation to revise your plan and you don't know my plan THEN
Say(?you, "Wait a second, our plans conflict")
SpeechAct(INFORM_PROB, ?plan, ?you)

Summary

Socially situated planning provides one mechanism for improving the social awareness of agents. Obviously this work is in the preliminary stages and many of the limitation and the relationship to other work could not be addressed in such a short article. The chief limitation, of course, is the strong commitment to defining social reasoning solely at the meta-level, which restricts the subtlety of social behavior. Nonetheless, our experience in some real-world military simulation applications suggest that the approach, even in its preliminary state, is adequate to model some social interactions, and certainly extends the state-of-the art found in traditional training simulation systems.

Acknowledgements

I greatly benefited from discussions with Stacy Marsella and Paul Rosenboom. Thanks to Jeff Rickel for letting me steal his bodies. Clark Elliott, Jeff Rickel, Lewis Johnson, Celia Pearce, Kate LaBore, Wendy Hui Kyong, and Ricardo Dominguez gave greate comments, insights, and suggestions. This research was funded by the Army Research Institute under contract TAPC-ARI-BR

References

desJardins, M. & Wolverton, M. 1998. Coordinating planning activity and information flow in a distributed planning system. AAAI Fall Symposium on Distributed Continual Planning.

Gratch, J. and Hill, R. 1999. "Continous Planning and Collaboration for Command and Control in Joint Synthetic Battlespaces." 8th Conference on Computer Generated Forces and Behavioral Representation.

Grosz, B., and Kraus, S. 1996. "Collaborative Plans for Complex Group Action," Artificial Intelligence, 86(2).

Hill, R., Chen, J., Gratch, G., Rosenbloom, P., and Tambe, M. 1997. "Intelligent Agents for the Synthetic Battlefield," in AAAI-97/IAAI-97, pp. 1006-1012

Lazarus, R.S. 1991. Emotion and Adaptation. Oxford Prs

National Research Council (NRC), 1998. Modeling Human and Organizational Behavior. National Academy Press, Washington D.C.

Ortony A., Clore, G. L., & Collins, A. 1988. The Cognitive Structure of Emotions. Cambridge University Press.

Pell, B. Bernard, D., Chien, S., Gat, E., Muscettola, N., Nayak, P. P., Wagner, M., Williams, B. 1998. An autonomous spacecraft agent prototype *Autonomous Robotics, 5.*

Rickel, J. and Johnson, L. 1998. "Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control." *Applied Artificial Intelligence (13).*

Tambe, M. 1997. "Agent Architectures for Flexible, Practical Teamwork," in AAAI-97, pp. 22-28.