Conflict Management for Agent Guidance

Karen L. Myers
Artificial Intelligence Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
myers@ai.sri.com

David N. Morley
Artificial Intelligence Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
morley@ai.sri.com

ABSTRACT

For agent technology to be accepted in real-world applications, humans must be able to customize and control agent operations. One approach for providing such controllability is to enable a human supervisor to define *guidance* for agents in the form of policies that establish boundaries on agent behavior. We consider the problem of conflicting guidance for agents, making contributions in two areas: (a) outlining a space of conflict types, and (b) defining resolution methods that provide robust operation in the face of conflicts. These resolution methods combine a guidance-based preference relation over plans with extensions to the set of options considered by an agent when conflicts arise.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Plan execution, formation, and generation

General Terms

Human Factors, Languages, Algorithms

Keywords

Agents, Advisability, Conflict Resolution

1. INTRODUCTION

Many potential applications for agent technology require that humans and agents work together in order to accomplish tasks effectively. This requirement is especially important for domains where task complexity precludes formalization of agent behaviors for all possible eventualities. In such domains, the availability of mechanisms by which a human supervisor can provide direction will enable agents to be informed by the experience, breadth of knowledge, and superior reasoning capabilities that a human expert can bring to the problem-solving process.

In previous work, we defined a framework for *agent guidance* that supports dynamic directability of agents by a human supervisor [2, 3]. Guidance imposes boundaries on agent activity, thus enabling a human to customize and direct agent operations to suit his or her individual requirements. Our guidance framework focuses on two types of agent directability, namely *adjustable agent autonomy* and *strategy preferences*. Guidance for adjustable autonomy enables a supervisor to vary the degree to which agents can make decisions without human intervention. Guidance for strategy preferences constitutes recommendations on how agents should accomplish assigned tasks. For example, the directive Use helicopters for survey tasks in sectors on the west coast imposes restrictions on how resources can be used to perform a certain class of task.

Guidance is expressed in a formal language that builds on the underlying *agent domain theory* (e.g., individuals, relations, goals, plans) and a corresponding *domain metatheory*. The metatheory defines distinguishing semantic properties for domain theory objects, thus enabling users to express preferences among otherwise equivalent options. For example, *features* within the metatheory are used to designate semantic attributes of a plan that distinguish it from other plans for the same task. Among plans for route determination, there may be one with the features *Optimal* and *Slow* while a second has the features *Heuristic* but *Fast*. Although the two plans accomplish the same goal, their intrinsic characteristics differ significantly; the domain metatheory enables a user to express prefences based on such characteristics.

The semantic model defined in [3] interprets guidance as a *filter* on the plans that an agent can execute. Thus, when selecting from a set of applicable plans for a given goal, an agent will eliminate from consideration those plans that violate any current guidance.

User guidance provides a powerful mechanism for runtime customization of agent behavior. However, it also introduces the potential for problems should guidance recommend inconsistent responses. Such conflicts cannot arise with adjustable autonomy guidance, but are a significant issue for strategy preference guidance. Robustness of operations requires mechanisms for detecting conflicts and responding in a manner that guarantees agent stability.

We address two issues related to conflicting strategy preference guidance for agents. First, we identify different types of conflict that can arise. Second, we define automated techniques for resolving conflicts. Our approach combines the selective dropping of problematic pieces of guidance with a proactive capability to eliminate the source of conflicts by modifying current activities.

We have implemented our conflict resolution techniques within the Taskable Reactive Agent Communities (TRAC) framework [2, 3], which provides a domain-independent guidance capability for PRS agents [1]. The techniques have been evaluated within a simulated disaster relief task force in which a human supervisor manages a team of agents engaged in a variety of information-gathering and emergency response tasks. We call this testbed TIGER (TRAC Intelligence Gathering and Emergency Response).

AAMAS'02, July 15-19, 2002, Bologna, Italy.

2. TYPES OF GUIDANCE CONFLICT

Two types of conflict can arise with strategy preference guidance: *plan selection* and *situated guidance*.

2.1 Plan Selection Conflict

A plan selection conflict occurs when multiple pieces of guidance make incompatible recommendations for responding to a goal within a given cycle of the agent executor. Conflicts of this type can arise in different forms: *direct* and *indirect*.

A direct conflict occurs when guidance yields contradictory plan selection recommendations. Such conflicts can arise both at the level of plans (e.g., guidance reduces to the constraints *Execute plan P* and *Don't execute plan P*), and variable bindings (e.g., *Instantiate variable V to A* and *Instantiate variable V to B*, where $A \neq B$). An indirect conflict among guidance occurs when there is no direct conflict, yet the plans recommended by the guidance cannot complete successfully because of interplan interactions. Such a situation could arise due to future contention for resources, deadlock/livelock, or race conditions (among others).

2.2 Situated Guidance Conflicts

The filtering semantics for guidance satisfaction from [3] can yield unintuitive results in certain cases by eliminating from consideration plans that a user would like an agent to execute. To illustrate, consider a situation in which all TIGER vehicles are in use for various tasks and the human supervisor has asserted the guidance Adopt medical emergency tasks that involve more than 5 people. Emergency response plans within TIGER require the availability of a vehicle. Now suppose an emergency event arises. Because all vehicles are in use, no response plan could be adopted for the event. Had there been a vehicle available, however, an emergency response of some form would have been initiated.

In this case, there is a clear expectation on the part of the human supervisor for the system to adopt a task in response to the emergency. Supporting this reaction requires a generalization of the filtering semantic model described above.

More generally, this type of conflict arises in situations where a plan p is relevant for a current goal g but some precondition C of p does not hold, making p inapplicable. The unsatisfied condition may be blocked by a contradictory belief of the agent, or some already executing activity. This type of situation can arise independent of guidance. Our interest in such situations relates to cases where guidance would recommend the execution of p but the violation of C eliminates its consideration. In other words, the prior activity or state condition conflicts with the intent of applying the guidance. For this reason, we call this phenomenon a *situated guidance conflict*, as it depends on the consideration of guidance within a particular execution state of an agent.

In certain situations, there may be no recourse to address the violated conditions (e.g., consider a requirement for favorable weather). However, others may be *resolved* by undertaking appropriate actions in the domain. Proactive response of this type lies at the heart of our techniques for resolving for this class of guidance conflict.

3. CONFLICT RESOLUTION

Resolution of guidance conflicts requires a richer semantics for guidance satisfaction than the filtering model of [3]. Our approach extends [3] to incorporate a preference-based model that maximizes guidance satisfaction relative to stated priorities. Furthermore, instead of reducing the set of plans that an agent considers (by filtering plans that violate guidance), we use guidance to expand the set to include options that would otherwise be discarded as inapplicable in the current execution state.

3.1 Preference Semantics for Plan Selection

To resolve plan selection conflicts, we define a preference relation over plans that is grounded in the *priority* of an individual guidance rule q (denoted by $GPriority(q)$). The preference relation both rewards guidance satisfaction and punishes guidance violation. For a set of guidance Q, define Q_p^+ to be the subset satisfied by a plan p and Q_p^- the subset violated by p. The preference relation is determined by the following *guidance ranking* function.

$$GRanking(p) = \sum_{q \in Q_p^+} GPriority(q) - \sum_{q \in Q_p^-} GPriority(q)$$

3.2 Candidate Plan Expansion

Our approach to resolving situated guidance conflicts involves expanding the set of plans considered for a given goal. The expanded set builds on an agent's predefined plans, extending plans that guidance might recommend to compensate for violated applicability conditions. The expansion process requires satisfaction of prerequisites related to *resolvability* and *cost-benefit analysis*.

Resolvability The unsatisfied applicability conditions of the plan recommended by guidance must be *resolvable*. In particular, there must be identified methods (called *resolution plans*) that can be invoked to achieve the unsatisfied conditions. For example, consider conditions related to *serially sharable resources* (i.e., resources that can be used sequentially but not simultaneously). In this context, a resolution plan should free the resource employed by a current activity to enable its use by the guidance-recommended plan. Resolution plans could be defined that *cancel* the prior activity, *modify* the prior activity to eliminate the dependence on the conflicted resource, and *delay* the prior activity.

Cost-Benefit Analysis The benefits in following a piece of guidance must outweigh the costs associated with executing the resolution plans. Within TIGER, users can select from predefined policies grounded in (a) the cost of executing any required resolution plans, and (b) the priorities of the recommended activity and any conflicting prior activities.

To resolve situated guidance conflicts, we dynamically create a set of *proactive plans* to expand the set of candidate plans for a goal g. Each proactive plan is a variant of a *potentially applicable plan* – a plan for g whose applicability is blocked by one or more unsatisfied but resolvable preconditions. The body of the proactive plan incorporates activities from resolution plans to achieve the blocked preconditions, as well as the actions from the potentially applicable plan. In this way, proactive plans can extend the set of actions that can be taken by an agent for a given goal. For a given conflict, only a subset of the space of possible proactive plans will be created, in accord with the constraints imposed by the cost-benefit analysis.

4. ACKNOWLEDGMENTS

This work was supported by DARPA under the supervision of Air Force Research Laboratory contract F30602-98-C-0160.

5. REFERENCES

[1] M. P. Georgeff and F. F. Ingrand. Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.

[2] K. L. Myers and D. N. Morley. Human directability of agents. In *Proceedings of the First International Conference on Knowledge Capture*, 2001.

[3] K. L. Myers and D. N. Morley. Policy-based agent directability. In H. Hexmoor, R. Falcone, and C. Castelfranchi, editors, *Agent Autonomy*. Kluwer Academic Publishers, 2002.