Structured Negotiation *

Charles L. Ortiz, Jr.
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
ortiz@ai.sri.com

Eric Hsu
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
hsu@ai.sri.com

ABSTRACT

Structured negotiation is proposed as a new method through which collaborating agents can seek consensus on the apportionment of tasks and resources. The approach draws on research in collaborative planning and human dialog understanding: agent interactions are organized in a manner that reflects the structure of a shared plan. Negotiations are incremental and interleaved with the shared planning process while communications supporting negotiations are made efficient by drawing on knowledge of a prevailing context. Agent proposals to team members are annotated with causal information that compactly expresses relationships between new proposals and the current context. Normative guidelines for proposal generation further restrict communications of ancillary information to only those fragments that represent departures from the norm. Finally, a set of interpretation rules allows agents to infer information not explicitly communicated.

Keywords

Coordinating multiple agents and multiple activities; conflict resolution and negotiation; agent communication languages and protocols

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Distributed Artificial Intelligence*

General Terms

Algorithms,experimentation,theory

1. INTRODUCTION

Negotiation is one mechanism through which agents can arrive at a consensus regarding the apportionment of tasks and resources. A number of strategies for negotiation have been put forward in recent years; some draw on work in game theory while others seek a closer correspondence to the process that occurs between groups of human agents. Most approaches have focussed on negotiation among self-interested agents; that is, agents that maximize the expected utility of outcomes to their actions. This paper introduces a new form of negotiation, termed *structured negotiation*, which is concerned with the organization of negotiation among artificial agents in collaborative settings. Negotiation is viewed as a mechanism through which agents can exchange beliefs and intentions relevant to the collaborative planning process.

In this paper, all agents are assumed to work together as part of a team: as such, they are assumed to share the same utility function and are also assumed to be truthful. Communication is assumed to be costly and, hence, should be limited to valuable information. This last assumption directly motivates the need for negotiation: if agents shared *all* information, they could, in principle, individually compute optimal strategies for acting.

Given these assumptions, structured negotiation embodies the following principles: (1) communications that support negotiation should be efficient; (2) negotiation should be interleaved with planning; (3) processing should be incremental; and (4) interactions should be organized around evolving plans. Roughly speaking, one communication is more efficient than another if its message length is shorter and both communications result in equivalent transmission of information. Such information-loading is common in natural language dialogs: when an utterance is interpreted within some context, it will usually carry with it additional information not explicitly transmitted. In bandwidth-restricted environments, efficiency is a desirable property.

The process of collaborative planning is one that takes place over some period of time. It is unrealistic to suppose that agents will suspend negotiations until group deliberations are complete; similarly, it is unrealistic to suppose that agents can suspend deliberations until they have arrived at a consensus regarding the division of tasks and resources. Negotiations must be interleaved with planning; therefore, a communication language for negotiation should be able to refer to elements of a shared plan as well as relations between sub-plans. When negotiation is interleaved with planning, it cannot range over every possible issue or option at once: this would require that agents negotiate over *every* possible plan; an activity that is computationally prohibitive.

One way of realizing incrementality is by organizing negotiations so that agents can systematically elaborate their

*This work was supported by DARPA contract F30602-99-C-0169 and ONR contract N00014-00-C-0304.

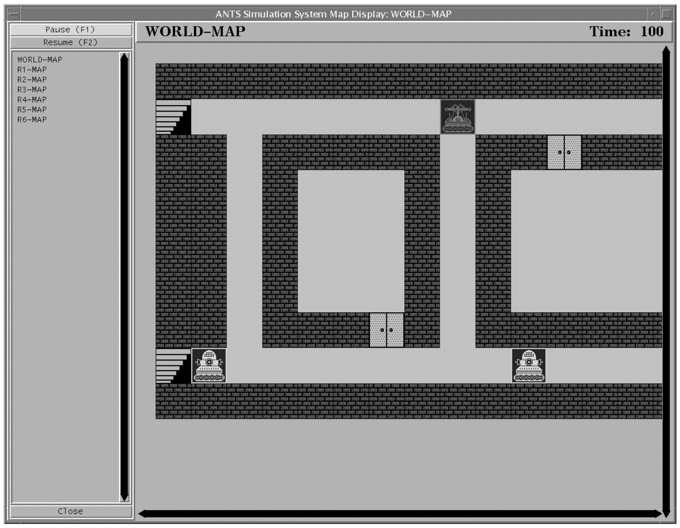

Figure 1: Map for hostage rescue scenario.

plans while at the same time seeking consensus on the division of resources and tasks. By structuring a negotiation, communications can be restricted to well-defined elements of a sub-plan. Structured negotiation organizes interactions in terms of task abstraction hierarchies as well as belief and intention dependencies. The former distinguishes this approach with conventional approaches which restrict task negotiation to range only over primitive tasks.

The next section begins by presenting a running example that will be used throughout the paper. A representation for actions and plans as well as a set of causal relations used to indicate relations between actions in sub-plans is then discussed. This leads to an algorithm for structured negotiation and an examination of communications *in situ*. The paper concludes with a discussion of the savings gained from a structured perspective on negotiation and a comparison with other approaches.

2. MOTIVATING EXAMPLE

Consider a futuristic world in which a team of robots is deployed within some area where hostages are being held by terrorists; the robots may be tasked with goals of locating objects such as terrorists, guards, hostages, or explosives. Figure 1 illustrates such a scenario taking place in a small area consisting of a series of corridors and rooms. High-level directives flow from some central point. In this scenario, robots R1 and R4 are tasked with patrolling corridor 1. They must then negotiate on a method for accomplishing that goal: for example by splitting the effort equally, or by having one of the robots perform the entire task on its own if the other is busy. As the example will illustrate, the robots' beliefs can differ: in particular, R1 is not aware of the other activities of R3 or R4.

Given such a task, consider the sample negotiation shown in Figure 2 between two robots, R1 and R4. R1 is located at the extreme left and R4 is at the extreme right. A third robot, R3 is at the top of the figure while, another robot, R2, is not shown. Corridor 1 is the bottom row, corridor 2 the top row and corridor 3 is the middle column intersecting R3 and corridor 2. Corridor 4 is outside the figure. In this

```
(1) R1> I propose we use group recipe r26 for
patrolling corridor 1
(2) R4> Ok.
          (3) R1> I propose that I patrol the
          west half and you patrol the east half.
               (4) R4> I can't patrol the east
               half as I have to patrol corridor
               4 as well. I propose patrolling up
               to the intersection of corridors 1
               and 3.
                       (5) R1> Can R3 help with D?
                       (6) R4> No.  He is busy
                       patrolling corridor 2.
          (7) R1> Ok.  I accept your proposal.
(8) R4> Good.  Let's get going.
```

Figure 2: Sample negotiation between R1 and R4.

exchange, R1 first proposes the "normal" division in which the patrol task is divided evenly. Since, all of the agents are on the same team and have the same utility function, R4 interprets this proposal as an indication that R1 is unaware of R4's other commitments: it therefore shares that information with R1 and makes a counter-proposal. R1 is not sure whether R3 can help (in fact, given the shared preferences, it is unsure whether R4 knows whether it can help); this explains exchange (5). Robot R4 interprets this as a request for information and therefore shares the information in (6) (we can assume that, for example, R3's antenna just went down and R1 cannot communicate with it). Having updated its beliefs in the course of these exchanges, R1 accepts the proposal (7) and R4 confirms this (8). The embeddings shown reflect the context. For example, message (5) is interpreted as "Can R3 help you with patrol of D *so that you can help patrol half of corridor 1 during the times we have discussed, while maintaining your other commitments* ?" in which the italicized fragment is understood as part of the prevailing context.

Notice that not all negotiation involves task selection: some will involve exchanging useful information and establishing beliefs that represent preconditions for actions [10].

3. REPRESENTATION LANGUAGE

This paper makes use of a multi-agent representation language called $\mathcal{HL}$ [11, 12]; details of the syntax and semantics of the language can be found in the cited references. The language is a sorted modal first order language with sorts for events, times, fluents (properties of the world that change with time), and objects. The language contains two predicates: $occurs(e,t)$ reports the occurrence of event type e at time t, and $holds(f,t)$ reports that fluent f is true at time t; time constants range over the integers and the truth of a formula is given relative to some world. An agent i's belief at time t in some ϕ is expressed as $holds(Bel(i,\phi),t)$, where ϕ can be a temporal term written in one of the functional forms $Holds(\psi,t')$ or $Occurs(\alpha,t')$[1]. Complex event types can be constructed through operators normally found in dynamic logic. Among these are: $occurs(\alpha;\beta,t)$ (α is followed by β), $occurs(\alpha^*,t)$ (α occurs zero or more times), $occurs(\alpha\cap\beta,t)$ (both α and β occur at t); $occurs(\alpha\cup\beta,t)$ (ei-

[1]Notice the upper case convention.

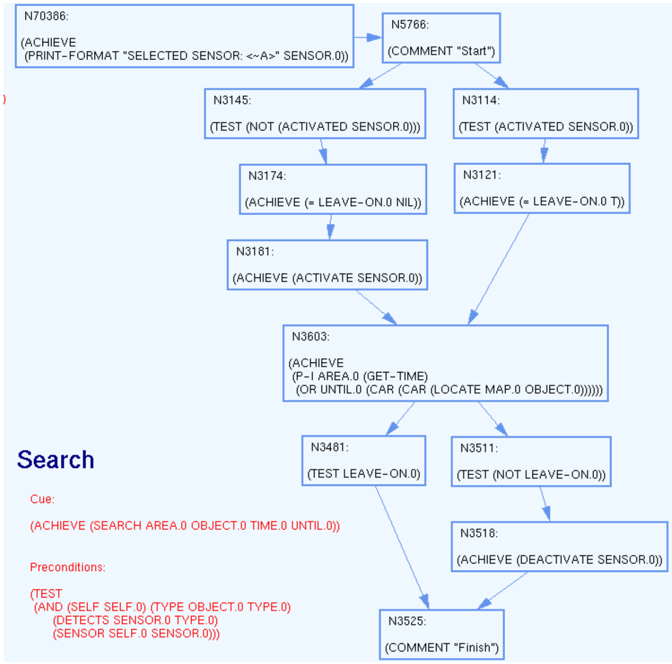

Figure 3: PRS recipe searching for an area.

ther α or β occurs at t); and $occurs(\phi?, t)$ (true if $holds(\phi, t)$ is true). From these operators one can then construct expressions such as: $occurs(\text{IF } \phi \text{ THEN } \alpha \text{ ELSE } \beta, t)$ which reports the occurrence of α if the condition ϕ holds and the occurrence of β otherwise.

The SharedPlans theory of collaboration is used in this paper to structure negotiations. It is based on a mental-state view of plans [6]. Rather than associating a plan for some action, α, with a group of actions that can achieve α, a plan is instead a rich structure consisting of a set of beliefs and intentions.[2] Intentions come in two varieties: an intention-to perform some action represents an individual commitment on the part of an agent to perform that action, while an intention-that instead represents a commitment to some condition. The table shown in Figure 1 lists some of the other operators used in the theory.

The approach taken in this paper follows that described in [12] in which mental actions, describing updates to beliefs and intentions, are reified. In this paper, these actions include $update(\phi)$ for update one's beliefs with ϕ and two negotiation actions discussed in a later section. The causal relations shown in Figure 2 are part of that theory and will prove useful in the specification of relations between plans.[3]

Agents are assumed to have access to a common library of recipes [6] that describe how tasks are decomposed. In our implementation, recipes are represented using the Procedural Reasoning System (PRS) [4, 5].

Figure 3 is an example of a PRS recipe for searching an area using a particular sensor. Recipes can either describe individual or group actions. Group activities are assumed to

[2] Other collaborative systems include STEAM [18] which is based on the joint intentions model; the latter differs from SharedPlans in that it argues for the utility of a separate mental attitude of a joint intention.

[3] In this paper, we view recipes as methods for action.[12]

Operator	Interpretation
FIP	An agent has a full individual plan
FSP/PSP	A group has a full/partial shared plan
CBA	An agent can bring about an act
BCBA	An agent believes it CBA an act
MB	A group mutually believe a proposition
MBCBAG	A group MB they CBA an act

Table 1: Operators used in SharedPlans

be decomposable along a resource dimension. For example, in patrolling a corridor, the size of the area might represent one natural way of dividing that activity and a "fair" division might allocate one-half to each of two agents. Certain high-level actions in a recipe might also be distinguished as representing *roles*: for example, a lookout and a patrol agent. A recipe, R_α, for some action, α, is represented as a tree of subactions; alternative instantiations of a recipe are identified by associating some set of constraints, Θ.

A full shared plan (FSP) is defined as follows. A group, GR, has a full shared plan, n, at time T_p to perform act α at time T_α using recipe R_α in context C_α.
$holds(FSP(n, GR, \alpha, T_\alpha, R_\alpha, C_\alpha), T_p) \equiv holds($

1. GR has a recipe for α:
   $R_\alpha = \{\beta_i, \rho_j\} \wedge MB(GR, R_\alpha \in Recipes(\alpha))$.

2. For each single-agent β_i in R_α, there is a $G_{\beta_i} \in GR$:
   (a) G_{β_i} intends to β_i: $Int.to(G_{\beta_i}, \beta_i, T_{\beta_i}, C_{\beta_i/\alpha})$.
   There is a recipe, R_{β_i} for β_i s.t.,
      i. G_{β_i} believes that it can β_i : $(\exists R_{\beta_i})$
      $[BCBA(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, constr(C_\alpha) \cup \{\rho_j\})$
      ii. G_{β_i} has a full individual plan for β_i:
      $\wedge FIP(G_{\beta_i}, \beta_i, T_{\beta_I}, R_{\beta_i}, C_{\beta_i/\alpha})]$

   (b) The group mutually believe (2a),[4]

   (c) The group is committed to G_{β_i}'s success:
   $MB(GR, (\forall G_j \in GR, G_k \neq G_{\beta_i})$
   $Int.th(G_j, (\exists R_{\beta_i}) CBA(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i},$
   $constr(C_\alpha) \cup \{\rho_j\}), T_{\beta_i}, C_{cba/\beta_i/\alpha}))$

3. For each multi-agent β_i in R_α, there is a $GR_{\beta_i} \subseteq GR$:

   (a) There is a recipe, R_{β_i} for β_i s.t.,
      i. GR_{β_i} mutually believe they can β_i with R_{β_i}
      $\exists R_{\beta_i}[MBCBAG(GR_{\beta_i}, \beta, R_{\beta_i}, T_{\beta_i},$
      $constr(C_\alpha) \cup \{\rho_j\})$

      ii. GR_{β_i} has a full SharedPlan for β_i using R_{β_i}:
      $\wedge FSP(m, GR_{\beta_i}, \beta_i, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})]$

   (b) The group mutually believe (3a)

   (c) GR is committed to GR_{β_i}'s success:
   $MB(GR(\forall G_j \in GR \setminus GR_{\beta_i})$
   $Int.th(G_j \exists R_{\beta_i} CBAG(GR_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, constr(C_\alpha)$
   $\cup \{\rho_j\}), T_{\beta_i}, C_{cba/\beta_i/\alpha})), T_p)$

If an agent is proposing some β as part of step (3a), then that proposal will be interpreted as contributing — in the way described by an appended causal relation — to FSP for α. In this way, the theory of SharedPlans focuses negotiation on the most important aspects of a current plan.

[4] For brevity, formalizations of (2b) and (3b) are not shown.

Causal Relation	Interpretation
α **Enables** β	α makes β possible
α **Prevents** β	α makes β impossible
α **Helps** β	α reduces resources needed for β
α **Method** β	α represents a method for β

Table 2: Summary of several useful causal relations.

4. NEGOTIATION SITUATIONS

The analysis of structured negotiation protocols can be simplified by considering progressively more complex agent types along a continuum that varies according to the degree to which each individual agent's beliefs or intentions conflict with those of another agent. First we note that the notion of *full belief exchange* is an idealization: if robots are equipped with sensor suites, this implies that their perceptual information stores are being updated on a continuous basis. It is unreasonable to suppose, and probably unnecessary to assume, that each individual update will be propagated among all team members on a continuous basis.

The simplest case is one in which, at any time, the agents share perfect beliefs about the world; agents exchange only beliefs about their own intentions during a negotiation and there are no problems related to restricting truthful exchange of information. In this case, a negotiation terminates as soon as each agent has sufficient knowledge of those intentions of another agent which might eliminate a proposal from consideration.

A slightly more complex situation is one in which agents also have perfect beliefs and perfect perceptual capabilities; however, each agent is privy to a spatio-temporally restricted "view" of the world. This sort of situation will typically result in belief incompleteness on the part of one or more agents; in this case, agents will exchange not only beliefs about intentions, but also share beliefs that will serve to update those of its team member.

More complex scenarios involve belief *revision* in which an agent's beliefs can be incorrect; old beliefs might be subject to correction based on inputs received from a team member. An agent's intentions might also change as a consequence of a belief revision. Such situations are much more volatile in the sense that even though agents might arrive at a consensus involving a particular task, the agreement might not be justified if further sharing of beliefs takes place.

Finally, agents can be *heterogeneous* in the sense of having different capabilities; these capabilities are manifest in their recipe libraries. In this case, even if two agents share perfect information about their respective beliefs and intentions, they still must coordinate activities that involve the specialized capabilities of a team member.

Combinations of elements from each of these can lead to further complexities: take for example a group of agents that share recipe libraries but not beliefs and, furthermore, those beliefs can be incorrect. In this paper, we will focus on combinations of the first two cases above: we will assume that an agent's beliefs are correct and the central goal of structured negotiation will be to identify those beliefs and intentions relevant to the evolution of the current shared plan and which a team member might be lacking.

5. NEGOTIATION PROCESSES

In this section an algorithm is presented for structured negotiation. The algorithm is expressed in the language described earlier in which mental actions are reified. Two processes are defined: the first forms a proposal for some action, α, and the second responds to a proposal. It is assumed that a group, consisting of agents i and j, has already been chosen and that the agents share the same preferences (from which a suitable utility function can be constructed). Preferences are expressed using an operator $holds(prefer(Agent, \phi), t)$ [2]: the intuition is that, in the current circumstances (time t) $Agent$ prefers ϕ, where the latter is usually a statement of the form $Occurs(Act, t')$; this represents a sort of *action choice*. For simplicity, negotiations are assumed to take place between only two agents.

In contrast to agent communication languages (ACL) based on speech acts, only two primitive communication actions are made use of here: $send(i, j, Msg)$, referring to $i's$ communication of Msg to j; and $receive(i, j, Msg)$. Also, in contrast to discourse understanding systems based on SharedPlans [10], a stack is not used to record context: their is no reason to restrict artificial agents regarding the order of negotiations. Hence, a context is simply the set of plans and intentions regarding elements of a plan. Those plans which are partial can be referred to in negotiations. Since causal relations need to specify a particular plan, this can be accomplished by appended additional descriptive information to the action description: for example, $occurs(\alpha, t)$ **enables** $occurs(\beta@FSP(232), t')$ which says that α performed at time t will enable the β that is planned as part of FSP number 232 at time t'.

The algorithm for negotiation is described by two processes: *propose* which assumes a group of two agents, $\{i, j\}$, where agent i has been made aware of some new, multi-agent task, α that must be performed at time T_α; it triggers a negotiation with agent j. The *process_proposal* is triggered at the receiving side by the other agent: it takes a proposal for α and determines whether that proposal is possible from that agent's point of view.

The definition shown in Figure 4 can be glossed as follows. If agent i proposes some action which represents a division of some resource or task with agent j, then the proposal should either: (1) confirm the typical distribution of the task or resource, or (2) give reasons for a departure from the typical distribution.[5] For example, if i proposes to divide the task of patrolling a particular area in a less than even way, i should communicate a reason for doing so: for example, because of other commitments. The prevention clause corresponds to the reason for the agent's proposal. This has a strongly counterfactual flavor: if ϕ represents the preventing condition, then the clause states that if the state of affairs described by ϕ had *not* obtained then the proposal would have been acceptable. In general, it can be difficult to choose the correct preventing condition if several conditions and actions would jointly cause some desired state (i.e., the state that would follow if the proposal were accepted). Instead of the approaches based on argumentation [1], structured negotiation focuses on *causal reasons*, giving preference to the following explanations: (1) any existing commitment that conflicts with the proposal (the proposing agent can be assumed to have not been aware of this, otherwise it would not have proposed); (2) there is some fact

[5]To simplify the presentation, the definition only covers one round of negotiation; the definition should be embedded in a loop that backtracks over alternatives. See [13].

$occurs(propose(i, \alpha, \{i,j\}, T_\alpha), t) \equiv occurs($

if there is a recipe

[IF $\exists R_\alpha \exists \Theta . R_\alpha \in Recipes(\alpha)$

$\wedge\ \theta \in R_\alpha \wedge \Theta = constr(R_\alpha)$

that the agent believes is possible

$\wedge\ Bel(i, CBA(\{i,j\}, \alpha, R_\alpha, T_\alpha, \Theta)$

$\wedge\ \exists r \in R_\alpha$

the agent prefers r which can include j-acts

$\wedge\ prefer(i, r))$

create a new FSP id and update mental state

with FSP to do r and then notify j

THEN $newid = x;$

$update(In.th(i, FSP(x, \{i,j\}, do(r))));$

$send(i, j, holds(Int.th(i, FSP(x, \{i,j\}, do(r))))$

explain to j the role r plays

$\wedge\ Bel(i, Occurs(r, T_\alpha)$

Method $Occurs(\alpha, T_\alpha)), t));$

explain if the choice is not the normal one

IF $normal(s) \wedge s \in R_\alpha$

$\wedge \exists p. Holds(p, t)$ **Prevents** $Holds(prefer(i, s), t)$

THEN $send(i, j, Holds(p, t)$

Prevents $Holds(prefer(i, s), t))], t)$

Figure 4: Proposal generation definition.

$occurs(process\_proposal(i, int.th(j, FSP(n, G, \alpha, t, R, \Theta))), t) \equiv$

$occurs($

if there is something physically preventing, inform

IF $\exists \phi. Holds(\phi, t)$ **Prevents** $Holds(CBA(i, \alpha, R, t, \Theta), t)$

THEN $send(i, j, holds(\phi, t)$ **prevents**

$holds(CBA(i, \alpha, R, t, \Theta), t)$

else, if can't help because of a prior commitment

ELSE IF $\exists \beta. Int.to(i, \beta)$ **Prevents** $occurs(\alpha, t)$

THEN $send(i, j, holds(Int.to(i, \beta)$ **Prevents**

$Occurs(\alpha, t), t))$

else, if need to establish belief in ϕ first

ELSE IF $\exists \phi. \neg Bel(i, \phi)$

Prevents $Int.th(i, FSP(x, \{i,j\}, \beta))$

THEN $newid = x; send(i, j, int.th(FSP(x, \{i,j\},$

$helping(j, achieve(\phi))))$

else, if need help establish sub-plan first

ELSE IF $\exists . \neg occurs(helping(j, \alpha), t')$

prevents $Occurs(\alpha, t)$

THEN $newid = y; send(i, j, int.th(FSP(y, \{i,j\},$

$helping(j, \alpha)))), t)$

update context and mental state as in *process*

Figure 5: Proposal processing definition.

about the world, which the proposing agent is unaware of, which prevents the proposed action; and (3) the agent is physically unable to contribute in the way in which the proposer suggests. In an agent receives a message that some condition, ϕ, is preventing a plan, then a set of interpretation rules is invoked which corrects the knowledge base (since ϕ was highlighted because it represented a departure from the norm. See [13].).

Actions that are recommended through the computation of preferences (which include individual *and* group preferences calculated with the *prefer* operator) can be either specific or general. In the former case, an agent might, for example, propose to patrol $area(A, .25)$ which could be taken to mean "1/4 of corridor A." In the latter case the agent might propose to patrol $area(A, .25)$ and perform a $helping(i, \alpha)$ act-type, where $helping(i, \alpha)$ is any act-type performed by agent i that helps in the performance of α.

The *process_proposal* activity is defined for action α using recipe R and under constraints Θ as follows (to simplify, some of the arguments to intentions are not shown). (Due to lack of space the last two steps are not shown. The *helping* term is just an abbreviation for the *help* causal relation, involving instead an agent and an act.) Consider the example discussed earlier. Embeddings are implicit in the expressed dependencies between actions that are part of FSPs. Exchanges (4) and (6) can be understood as a departure from the norm and explained by the clause referring to prior commitment.

6. COMMUNICATION *IN SITU*

When a proposal is processed by an agent, that agent might infer additional information not explicitly transmitted. This additional information corresponds loosely to the notion of perlocutionary force in speech act theory: it represents a side-effect to the communication. Messages between agents are of the form: $int.th(Agent, Formula)$. The formula appearing in the scope of the intention-that message usually refers to a collaborative plan toward some activity. If the agent wishes to transmit additional supporting information, then a conjunction of formulas is transmitted; supporting information takes the form of expressions of causal dependencies between elements of a shared plan (these elements can include beliefs; for example, lack of knowledge as to some ϕ might be preventing completion of a particular plan). In contrast to speech act theory, the set of possible illocutions is not closed under this scheme: the interpretation of a message depends on the prevailing context.

In the processes described earlier, helpful behavior is embedded in the process definitions. In a fully fleshed-out theory, such behaviors would represent outcomes of intermediate inferences. For example, rather than communicating a preventing condition immediately, the system should infer that the other agent probably is not aware of that information and telling it would "help" that agent. Axioms that capture such chains of reasoning are straightforward to define in the representation described.

7. IMPLEMENTATION

The example shown in Figure 1 consists of what we will refer to as "cover" and "point" agents; these appear at the bottom of the map and will negotiate a patrolling pattern

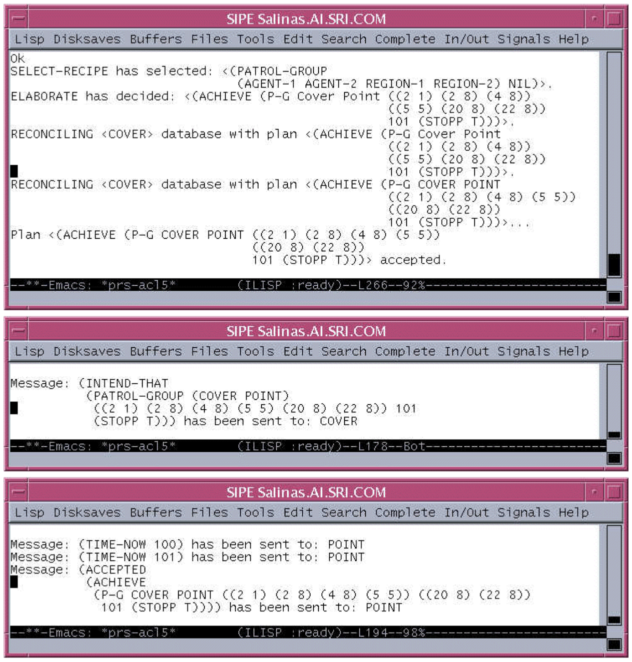

Figure 6: PRS execution trace for the scenario.

within the discrete and uncertain simulation domain. A series of process locks ensures that any simulation action (including deliberation) will take a fixed amount of time, eliminating experimental variables such as processor speed and thread priority. Uncertainty dictates that each agent must maintain its own, possibly incomplete, map; the illustration shows a ground-truth "world view" that is not necessarily accessible to any of the agents.

Figure 6 traces this process as the "cover" agent takes on a larger patrol due to negotiation with the "point" agent. The middle window shows an initiating message from the command center, instructing cover to join point in patrolling an area delimited by six waypoints.

The top window depicts cover's half of the planning process, which begins with the selection of a recipe for joint patrols. The recipe is an abstract plan template which must be elaborated, as shown in the next entry. At first this is done using the default elaboration: an equal division of the patrol area. The next step, reconciling, determines whether the tentative plan is compatible with cover's current intentions and actions. In the example, this step fails for the point agent, and it negotiates a new proposal. In the next line, cover can be seen reconciling this new and unequal proposal with its own constraints. The bottom line shows that the plan is acceptable; point receives the confirmation message shown in the bottom window.

The scenario presented earlier in Figure 1 characterizes an experimental domain in which we use structured negotiation as a high-level controller for autonomous vehicles. Small rovers and helicopters each run their own identical implementations, which listen for high-level mission orders to search particular areas or pursue targets. We have tested these in both simulations of the form shown in 1 and with actual physical robots. we discuss only the former work in this paper. The controllers then make use of the methods spelled out below to arrive at consensus on a team recipe for fulfilling the mission, and on who will be responsible for each part of the strategy. Our implementation interfaces with PRS which maintains the agents beliefs in inference-capable databases that can be consulted against plan constraints, and provides the messaging infrastructure for receiving orders and negotiating. It also handles perception-triggered belief updates, and the dispatch of plans to lower-level actuators. We have added a language for representing hierarchical, potentially partial plans based on the SharedPlans formalism already presented. These form the structures over which the agents negotiate, making use of the following procedures, based on the theory of structured negotiation, implemented to minimize message size and frequency

DEFAULT-SELECTION On receiving a mission or plan, select the pre-arranged default recipe or elaboration, respectively. Reconcile that selection with your knowledge base, identifying any conflicting beliefs. If there is any grounds for deviating, notify collaborators by sending them the justifying beliefs. Signify assent by sending an empty list. On receiving a (potentially empty) list from all agents, compile the union[6] of all justifications and assume the pre-arranged default selection determined by the revised information.

DISCREPANCY-INFERENCE On receiving a proposal that is inconsistent with your knowledge base, signal your dissent by retaining the contradictory belief and sending it to the proposing agent. Make no other effort to synchronize beliefs. On receiving such a message, reconcile the belief with your knowledge base.

CONSTRAINT-IMPLICATION When updated information necessitates a modified plan proposal, send collaborators the new facts before doing any re-planning. On receiving such a message, interpret it as a new proposal over the set of plans enabled by the new information.

ABSTRACT-REFERENCE When soliciting assistance in performing particular actions, refer to them directly by the most specific sub-task they serve. If your most recent correspondence concerns that sub-task, no identifier is necessary at all.

HIERARCHICAL-ORDERING Order above activities by negotiating over increasingly specific sub-plans.

7.1 Evaluation

The efficiencies provided by pre-deploying common structures among collaborating agents are intuitive, but difficult to compare with the set of all unstructured approaches. Because most could be considered a form of encoding, any conclusions should be general enough to preclude the particulars of our message compression functions. Hence, this section will first describe the savings provided by each algorithm in abstract terms, before presenting an example of their occurrence in our vehicle coordination system. Then, we will formalize the bandwidth savings in terms of number of messages transmitted as well as number of facts or referents transmitted in those messages. The latter is intended as a general measure of message length; a fact is a single proposition from an agents knowledge base, and a referent is either a plan, sub-plan, or action that an agent refers to in its message. Instead of basing such savings on compar-

[6]The method for resolving potentially conflicting reports should be determined by the domain requirements. Our method is to believe a statement over its negation if we hear it from more agents than its negation. We do not use inference to check for implied inconsistencies.

	Msgs Saved	Decreased Length
Default Selection	6	6
Discrepancy Inference	7	13
Constraint Implication	0	6
Abstract Reference	0	42
Total	13	67

Table 3: Conservation of Message Bandwidth

ison with some arbitrarily selected or mock non-structured approaches, our evaluation will identify a single key feature such that the savings described hold over any method that lacks that feature. Only in this context will we finally list empirical results from running structured negotiation in our unmanned vehicle domain, summarized in Table 3. Such results were derived by running three vehicle agents, each using a structured negotiation controller, over ten different scenarios wherein they received an order to jointly patrol a specific area. By varying the agents capabilities, their knowledge of each others capabilities, and unexpected events in the domain, we were able to induce all of the above algorithms to come into play. Over all our system used a total of 28 messages per agent during the ten runs, containing an average of 1.18 facts or referents per message. Without structured negotiation, the runs would have required 41 messages, each containing 2.4 facts or referents, if they were to use the alternative methods described below.

Default Selection. Because the agents are cooperative, they can predetermine default responses to various proposals, and further predetermine their responses given a specific set of relevant beliefs in their knowledge base. Thus agents do not need to communicate when there are no extenuating circumstances, and when there are, they need only compile the circumstances and know that everyone will arrive at the same conclusion based on that information. For instance, in one of the runs all agents were equally capable of traversing all regions of the patrol area, and believed this to be true of each other. Hence, they each took a pre-determined equal division and did not need to send a single message. In general, the use of defaults saves a single message each time an agent avoids proposing the allocation that has been chosen as the default. When there are extenuating circumstances, the agent spends a message to communicate them, but needs no more communication after that and hence is no worse than a method that does not use defaults. In the experiments six messages were saved in this way (and hence six referents, or units of message length.)

Discrepancy Inference. Depending on the domain, it is at best inefficient and at worst infeasible to synchronize agents beliefs at all times. However, any discrepancies relevant to the success of a particular proposal must be brought forward during negotiation. Using this method, when an agent knows something its partner doesn't it waits to receive an unworkable proposal, and infers that the proposing agent must be missing this knowledge. If such a circumstance does not arise, there is no need to resolve a given discrepancy. For instance, in the scenario played out in Figure 1, R4 infers from R1's initial proposal that R1 is unaware of its commitment to patrol another corridor. This confirms the discrepancy's relevance, and R4 encodes the conflict by communicating the fact that it is committed. In general, the savings over perpetual database reconciliation is boundless if every possible fact is to be synchronized. In interests of fairness, in our experiments we considered a system where recipes are tagged with relevant factors or preconditions, and only beliefs referring to such considerations were updated whenever they were discussed. Usually this meant an update of two or so facts concerning fuel level, the presence of obstacles, and vehicle capabilities. In comparison, structured negotiation was able to save a total of 7 messages through discrepancy inference, and 13 referents.

Constraint Implication. Should an agent acquire new information necessitating re-negotiation, it need not counter-propose a series of possible arrangements satisfying the new constraint. Rather, it refers to the entire set of such new proposals by simply communicating the new information. Combined with default selection, this provides a well-defined and completely determined set of alternatives, ensuring that such referential economy will still be uniformly interpreted. In certain experimental scenarios, a vehicle would become immobilized by an unexpected calamity, and in announcing as much it would be simultaneously proposing the set of task allocations where it would not need to move. In general, the savings are bounded only be the number of alternative plans supported by the domain. In our domain this number was 4 in all relevant runs, of which there were two. Hence, constraint implication was able to save 6 referents over the course of the experiments. It is more difficult to quantify a second benefit of this method. Specifically, the agent can interleave planning with action by transmitting the new information before it has computed the set of new proposals. The greater the complexity of such computation in a particular domain, the greater the savings in speed afforded by this advantage.

Abstract Reference. Because the agents are deployed with or otherwise develop common recipes, they can refer directly to specific portions of the plan template without listing a chain of hierarchical tasks. This keeps messages concise, for instance when an agent asks for help with a particular sub-task. In some of the scenarios, a ground vehicle was able to ask a helicopter to perform a particular task in support of a series of higher-level tasks, without having to explicitly name each higher-level task in the chain. In general, this saves in message length over negotiation methods that either do not assume common plan templates, or do not structure their negotiations at all, instead sequencing sets of primitive actions. The latter case is not so unfair a comparison, give the insulation of many negotiation methods from multi-agent planning. For the sake of experiment, though, we compared structured negotiation with the former approach, where the savings would not be so extreme. Using this model, the savings in message length depends on the average depth of plan decomposition where an agent might refer to its activities. In our experiments this was usually seven levels, so the agents saved six referents whenever they asked for help. This happened seven times, for a total savings of 42 referents.

Hierarchical Ordering. The final feature of structured negotiation does not save in message number or size, but forms the basis of our system's anytime properties. Because the agents are negotiating over action sequences overlaid by a hierarchical structure, they can first focus on the most general tasks at hand. Hence, in case they have run out of time for deliberation or communication fails, they are still able to further elaborate their assigned high-level tasks and attempt them on their own without consensus. We were not able to simulate such conditions in our experiments, but were able to observe the top-down ordering of negotiations.

8. SUMMARY AND RELATED WORK

This paper has introduced a new form of negotiation targeted towards collaborative teams. Structured negotiation has the desirable property of efficiency of communication and incrementality. The latter is made possible through a structuring of negotiation and an interleaving with shared planning. We anticipate that this will also have useful applications for systems which must explain their actions to users; however, our focus so far has been strictly on automated negotiation among artificial agents.

The body of work in automated negotiation among self-interested agents has become quite large [14, 15, 8]. Research in negotiation in collaborative setting has been more limited. There are several areas of research that were very influential in the development of structured negotiation: discourse models [10], studies of negotiation in natural language discourse [16] and formal languages for argumentation in negotiation [7]. Work on natural language negotiation differs from structured negotiation in its focus on negotiations that are prompted by questions of resource-boundedness. For example, one agent might propose, "Let's do A because A enables B and because we want to achieve B," in a setting in which the hearer had not expended sufficient computational resources to be aware of the enablement condition. In structured negotiation, agents are assumed to share the same preferences and are assumed to be able to derive such inferences; the focus is instead on identifying incorrect beliefs that might be in the way of allowing one agent to collaborate with another.

The work on formal argumentation is similar in its use of a representation that refers to an agent's mental state; to, for example, express threats or communicate consequences of proposed actions. Many of the examples focus on self-interested agents; however, such an approach could be adapted to support the sort of negotiations described in this paper.[7] More recent work explores the use of argumentation in the context of a teamwork model [19]. The major contribution of structured negotiation, as compared to these alternatives, is its organizational and inferential elements, where the latter involves the use of causal annotations and the former exploits the structure of a shared plan.

One most commonly finds agent communication languages [17] that are based on speech act theory [9]. Typically, some closed set of speech acts is defined which corresponds to communication act types such as *inform*, *request*, or *warning* actions; the definitions of speech acts are usually expressed in terms of the changes in mental state that they bring about. Some in the discourse community have argued against such an approach on two counts: (1) the same speech act can have different interpretations (bring about different effects) in different contexts and (2) a sort of master-slave relationship is implicitly introduced by virtue of the assumption that a speech act necessarily brings about a change in the intentions of the hearer [10]. The latter can be an unwelcome introduction to collaborative interactions. The approach taken to communication in structured negotiation is parsimonious and avoids the potential problem of later having to define new types of speech acts.

Very little work has been done in interleaving planning and negotiation. Notable in this respect is the work of Ephrati and Rosenschein [3] which examines subplan aggregation through the use of consensus mechanisms. In contrast, the work described in this paper takes at its starting point a richer notion of plans [6].

9. REFERENCES

[1] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth ICMAS*, pages 31–38, 2000.

[2] J. Bell and Z. Huang. Dynamic goal hierarchies. In L. C. A. Rao and W. Wobcke, editors, *Intelligent Agent Systems: Theoretical and Practical Issues*, pages 88–103, 1997.

[3] E. Ephrati and J. Rosenchein. Multi-agent planning as a dynamic search for social consensus. In *IJCAI*, pages 423–429, 1993.

[4] M. P. Georegeff and A. L. Lansky. Procedural knowledge. Technical Report 411, SRI International, 1987. Artificial Intelligence Center.

[5] M. Georgeff. Communication and interaction in multi-agent planning. Technical Report 313, SRI International, 1984. Artificial Intelligence Center.

[6] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(1):269–357, 1996.

[7] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation. *Artificial Intelligence*, 104(1-2):1–69, 1998.

[8] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, 1995.

[9] A. Kronfeld. *Reference and Computation: An Essay in Applied Philosophy of Language*. Cambridge University Press, 1990.

[10] K. E. Lochbaum. A collaborative planning model of intentional structure. *Computational Linguistics*, 34(4):525–572, 1998.

[11] C. L. Ortiz. A commonsense language for reasoning about causation and rational action. *Artificial Intelligence*, 111:73–130, 1999.

[12] C. L. Ortiz. Introspective and elaborative processes in rational agents. *Annals of Mathematics and Artificial Intelligence*, 25(1–2):1–34, 1999.

[13] C. L. Ortiz and E. Hsu. Structured negotiation among collaborative agents. Long version in preparation.

[14] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[15] T. W. Sandholm. Distributed rational decision making. In *Multiagent Systems*, pages 201–258. MIT Press, 1999.

[16] C. L. Sidner. An artificial discourse language for collaborative negotiation. In *Proceedings of AAAI*, pages 814–819, 1994.

[17] M. P. Singh, A. S. Rao, and M. P. Georgeff. Formal methods in DAI: Logic-based representation and reasoning. In *Multiagent systems*, pages 331–376, 1999.

[18] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[19] M. Tambe and H. Jung. The benefits of arguing in a team. *AI Magazine*, pages 85–92, Winter 1999.