Flexible Negotiation in Teamwork: Extended Abstract

Zhun Qiu and Milind Tambe

Information Sciences Institute and Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292, USA
{zqiu, tambe}@isi.edu

Abstract

In a complex, dynamic multi-agent setting, coherent team actions are often jeopardized by agents' conflicting beliefs about different aspects of their environment, about resource availability, and about their own or teammates' capabilities and performance. Team members thus need to communicate and negotiate to restore team coherence. This paper focuses on the problem of negotiations in teamwork to resolve such conflicts. The basis of such negotiations is inter-agent argumentation (based on Toulmin's argumentation structure), where agents assert their beliefs to others, with supporting arguments. One key novelty in our work is that agents' argumentation exploits previous research on general, explicit teamwork models. Based on such teamwork models, it is possible categorize the conflicts that arise into different classes, and more importantly provide a generalized and reusable argumentation facility based on teamwork constraints. Our approach is implemented in a system called CONSA (COllaborative Negotiation System based on Argumentation).

Introduction

The past few years have seen an explosion of interest in multi-agent systems in general, and multi-agent collaboration or teamwork in particular. In multi-agent teamwork, agents must plan or act together in service of their common team goal. Unfortunately, in a complex, dynamic multi-agent setting, such coherent team action is often jeopardized by agents' conflicting beliefs about factors such as their environment, overall resource availability, and their own or their teammates' past and present performance and capabilities. Such conflicts in agents' beliefs may arise due to a variety of reasons. First, typically in a distributed environment, agents have access only to local information (not global information), obtained from local sensors or information access mechanisms. Thus, information locally sensed by one agent is unavailable to the other agents, leading to conflicts. Second, even if the same information is available to all of the agents, their interpretations of this information may differ due to their distinct individual contexts, or their differing sensor capabilities, leading to conflicting beliefs. Third, agents' individual problem solving or planning activities may often need to proceed without all relevant information from all others, and agents may thus produce local plans that conflict with those of its teammates. Finally, unreliable communication among agents may lead the sending and receiving agents to believe in conflicting information.

While recent research on teamwork has made progress in enabling agents to flexibly coordinate and communicate in a team, it has so far not addressed the problem of inter-agent negotiation for conflict resolution. This paper is focused on such collaborative negotiations in the context of teamwork. The topic of inter-agent negotiations has long been a subject of intense investigation in the multi-agent literature. Beyond multi-agents, "negotiations" is well-studied topic, in different forms, in different areas such as economics, and political science[1]. Indeed, investigations inspired from economics, such as game theoretic approaches (e.g., Rosenschein and Zlotkin [Rosenschein and Zlotkin 1994]), dominate much of the existing work on agent-based negotiation. However, much of this literature has focused on negotiations among self-interested agents, that attempt to maximize their individual utilities, rather than collaborative negotiations within a team. Furthermore, this literature often focuses on the outcome of the negotiation, e.g., by providing rules of encounter or conventions that ensure that agents will not have any incentive to deceive others. However, it does not provide guidance on constructing such agents' internal (cognitive) processes , e.g., their representations and reasoning employed, to enable effective and/or efficient negotiation.

Our research strongly contrasts with the above research thread, given its focus on *building* agents that can participate in collaborative negotiations in service of teamwork. In contrast with game-theoretic approaches, our approach is based on the notion of argumentation. Previous research in this area [Chu-Carroll and Carberry 1996, Freeman and Farley 1993, Mouaddib 1997, Parsons and Jennings 1996] has provided several general purpose techniques for argumentation-based negotiations. Building on this previous work, and particularly, Toulmin's pattern of argumentation [Toulmin 1958], we have developed an argumentation-based negotiation system called CONSA (COllaborative Negotiation System based on

[1] Indeed, books on negotiation date at least as far back as 2,500 years ago, e.g. Gautama's five volumes of "Nyaya-Sutra", or the "science of discussion" was written in 550BC.

Argumentation).

The key novel aspects of CONSA, particularly in contrast with systems of legal argumentation [Freeman and Farley 1993], are based on its focus on collaborative negotiation. This focus enables CONSA to exploit previous research on explicit, general models of teamwork [6,12], that can provide significant advantages in investigations of collaborative negotiations. In particular, a principled framework of teamwork enables a clearer understanding of collaborative negotiation, and the strengths and limitations of current techniques. For instance, such a principled framework enables an understanding of the different conflict types that can arise in teamwork, and the extent to which such conflict types have been addressed in previous work on collaborative negotiations. Indeed, such categorization of differences in conflict types is absent in previous work on collaborative negotiation. More importantly, teamwork models provide a more detailed, yet domain-independent expertise for agents to engage in argumentation. Such argumentation expertise is reusable across domains. For instance, agents can reuse argumentation knowledge based on role constraints and task relationships, or own and teammates' responsibilities in teamwork.

Domains and Motivations

The motivation for current research on negotiation is based on our previous work in complex, multi-agent domains such as real-world battlefield simulations [Tambe et al. 1995]. We have been building different teams of synthetic pilot agents that successfully participate in combat simulations in these environments. These pilot agent teams include companies of attack helicopter pilots, divisions of transport and escort helicopters pilots, and teams of autonomous pilots for the future generation unmanned air vehicles (UAVs). The second domain is Robocup [Kitano et al. 1997] where we have twice successfully participated in the RoboCup tournaments (twice in the top four teams). We are also investigating the role of agent teams for disaster rescue scenarios.

These above agent teams have been developed based on a teamwork model called STEAM [Tambe 1997]. STEAM is based on the joint intentions [Cohen and Levesque 1991] and SharedPlans [Grosz and Kraus 1996] theories of teamwork, but with practical extensions for monitoring and replanning as well as decision-theoretic communication selectivity. STEAM has provided significant teamwork flexibility in all of these applications. Yet, STEAM does not address the problem of conflicts in agents' beliefs and relevant negotiations to resolve such conflicts, limiting teamwork flexibility in key instances. We describe here just a few key examples that outline some of the basic issues for collaborative negotiations:

- *The proceed case*: A helicopter pilot team in the combat simulation environment must plan firing positions, i.e., positions for individual helicopters to hide and attack the enemy. Typically, the commander pilot agent plans all these positions, one per each team member, and sends each individual planned position to the relevant team member. For instance, a team member with the call sign *"cheetah102"* may obtain a position with coordinates that hide it behind a small hill. Once all of the positions are communicated, the commander asks its team to "proceed". Unfortunately, in one case, one of the positions never reached a team member (*"cheetah102"*), i.e., the message was sent but lost due to radio interference. Thus, the commander thought the message was sent, but *"cheetah102"* never received it, leading to a conflict in beliefs. Unfortunately, given its inability to resolve this conflict, when the team was asked to proceed, *"cheetah102"* just proceeded, without a firing position, and thus got stuck and did not attack the enemy.
- *The firing position case*: In the above firing position planning, the commander pilot agent typically successfully plans firing positions that are at least one or two kilometers apart from each other, as required by doctrine. However, as individual pilots fly to those positions, they may unexpectedly encounter enemy vehicles and react by autonomously changing the allocated positions. This may lead pilots to take up positions that are inappropriately close to each other. Or, two separate commanders may plan positions that may accidentally conflict with other. In such cases, individuals must resolve the conflict in firing positions via negotiations. Without such negotiations, they may take up firing positions that are inappropriate and dangerous.
- *The ball position case*: In RoboCup soccer, players have limited vision and thus inform each other about ball position. Typically, rather than communicating precise ball position (which constantly changes), they exchange high-level information, e.g., the defenders only inform each other if the ball is a threat, or if the ball is cleared (and hence not a threat). However, the players' sensors may deliver imprecise information about ball location, due to inbuilt uncertainties in the simulation, leading to a conflict in players' beliefs. Furthermore, the players own interpretation of what constitutes a threat may vary, leading to a similar conflict in beliefs. Without a negotiation capability, the agents are unable to resolve this conflict. Thus, one defender may insist that the ball is a threat, while another insists it is not. Defenders may thus continually exchange these messages, without any resolution, while the ball is kicked past them towards their goal.
- *The enemy position case*: In the combat simulation environment, a very similar situation to the one above in terms of ball position does arise (except that instead of the ball, agents are concerned about enemy position). The key difference here though is that there can be two or more different enemy units, rather than a single ball as in Soccer.

As seen in these examples, while team members do acquire conflicting beliefs, the lack of negotiation capabilities leads to a significant degradation in their individual performance as well as the overall team performance.

Analysis of Conflicts

As mentioned in the introduction, a teamwork model provides several advantages for collaborative negotiations, including a categorization of the conflicts types, clearer understanding of capabilities of previous collaborative negotiation systems, and more importantly, argumentation knowledge based on such teamwork models. Therefore, this section provides a brief description of a principled teamwork model (which in our case is STEAM [Tambe 1997]).

STEAM

STEAM is an implemented model of teamwork, aimed at enabling development of individual agents that can engage in flexible teamwork [Tambe 1997]. STEAM uses joint intentions as its basic building block and builds up a hierarchy of joint intentions corresponding to a team's goal or plan hierarchy. STEAM facilitates flexible teamwork via two classes of domain-independent actions. The first class of coherence preserving or CP actions is based on the commitments in the joint intentions theory. It includes the execution of the establish-commitment protocol [Tambe 1997] to establish a joint intention, and by requiring agents to jointly begin and terminate joint intentions, STEAM ensure full coherence within a team.

The second class of domain-independent actions in STEAM is maintenance and repair or MR actions, aimed at replanning (particularly via team reorganization). One key aspect of MR actions is an explicit specification of the dependency relationship of the joint intention on individual team members' activities, based on the notion of a *role.* A role constrains a team member vi (or a subteam) to some suboperator(s) op_{vi} of the team operator [OP]. Three primitive role-relationships can currently be specified in STEAM, implying:

AND-combination: $[OP] \Leftrightarrow op_{v1} \wedge ... \wedge op_{vn}$
OR-combination: $[OP] \Leftrightarrow op_{v1} \vee ... \vee op_{vn}$
Role dependency: $op_{vi} \Rightarrow op_{vj}$ (op_{vi} dependent on op_{vj})

These primitive role constraints may be combined, to specify more complex relationships. All these role relationships can create a whole role-constraint hierarchy, corresponding to an agent's plan hierarchy (see below). CONSA, our negotiation algorithm, currently exploits this hierarchy to guide agents to detect conflicts and generate efficient arguments.

STEAM also requires an explicit representation of a team's plan(s) or task. For a concrete example of explicit representation of team plans, consider Figure 1. It depicts an operator hierarchy (similar to a reactive plan hierarchy [Firby 1987]) for our Soar-based [Newell 1990] synthetic helicopter pilots developed using STEAM. One key novelty however is team operators (reactive team plans), which explicitly express a team's joint activities. Thus, operators shown in [] such as [Engage] are team operators. At any point in time, one path through this operator hierarchy is active, which is an agent's currently active joint intentions (corresponding to team operators), and intentions (corresponding to individual operators).

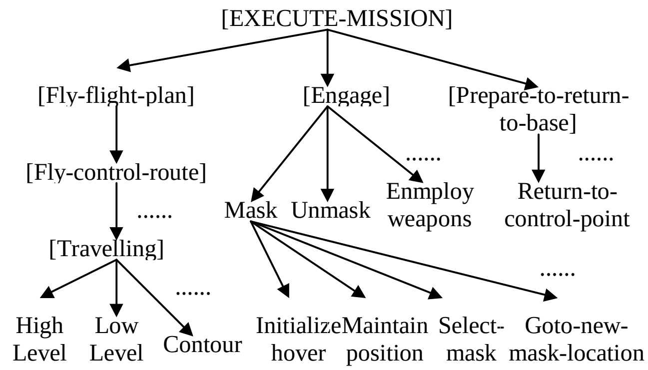

Figure 1: Helicopter pilot domain: Portion of team operator (reactive-plan) hierarchy.

Conflicts in STEAM

Given this background on STEAM, we can now categorize some of the conflicts that can arise in teamwork.

- Conflicts in joint initiation of a team operator: Two team members may have conflicting beliefs about coherently initiating a team operator. Conflict may arise because:
  a) Ability: One team member may be unable to participate in the execution of the team operator (it may not believe that the preconditions of the team operator are satisfied).
  b) Preference: One team member may prefer to execute an alternative team operator (it may have the ability, but it may still prefer the alternative).
  c) Cause: Even if both members are able to and prefer to initiate the joint goal, they may have different reasons to do this, which sometimes lead to conflicts also.
- Conflicts in joint termination of a team operator: Two team members may have conflicting beliefs about coherently terminating a team operator.
  a) Ability: One team member may be unable to terminate the team operator.
  b) Preference: One team member may prefer to continue executing the team operator.
  c) Cause: The termination may occur for different reasons, e.g., one agent may believe the joint goal to be achieved, while another believes it to be unachievable or irrelevant.
- Conflicts in the execution of an individual operator: Two agents, executing two separate individual roles may unintentionally interfere with each other's role

performance.

Since an individual operator is not initiated or terminated jointly, the conflict here may arise not in its initiation and termination, but in its execution. In contrast, in team operators conflicts may arise in initiation and termination. There can be conflicts in the execution of team operators as well. However, this is not mentioned as a separate category, since execution of a team operator involves execution of operators in its subgoal. Thus, execution conflicts for team operators are essentially conflicts in the initiation and termination of team operators that are executed in its subgoal, or conflicts in the execution of the individual operators executed in its subgoal.

We can now re-examine the cases from Section 2. Here, in "the proceed case", a team member's "proceed" message is intended to attain coherence in the termination of their current team operator to "plan-firing-positions". Thus, this is a type 2(a) conflict, due to ability. The "ball position" case is also a type 2(a) conflict, as players have conflicting beliefs about terminating their team operator to detect ball threat. The "firing position case" is a type 3 conflict, in agent's individual activities. Finally, the "enemy position" case belongs to type 2(c), since the helicopters agree to terminate the team operator, but have conflicting beliefs regarding enemy positions.

Approach Outline and Preliminary Implementation

As mentioned earlier, we have designed and implemented an approach called CONSA for collaborative negotiations. The following sections discuss the representations and reasoning in CONSA.

Representations of Beliefs

In devising an approach to address the problem of collaborative negotiation, the following key issue must be addressed. STEAM, particularly its CP actions, are aimed at maintaining coherence in the team, and they succeed in the absence of a "true" conflict, e.g., where an agent sending a message has new information that unavailable to other agents. Any approach to collaborative negotiation must accommodate such CP actions, and not force unnecessary negotiations in the absence of a "true" conflict. Of course, at present, the limitation in STEAM is exactly the opposite - it fails to recognize any conflict situations , and fails to negotiate. The new approach must thus reach a compromise where agents may consider negotiations iff there is a "true" conflict.

Agents' belief representation must thus enable efficient recognition of such true conflict situations. To this end, CONSA relies on equipping agents' beliefs with *strengths*. The strengths of a belief is derived based on the evidence that supports it. For instance, in the proceed case, the agent not receiving its firing position ("cheetah102") can strongly believe that it has not received this position, since there is no evidence that the firing position has been received. In contrast, when executing the team operator [fly-flight-plan] individual team members only weakly believe that there is no enemy on their flight path, because their local sensors only cover a portion of the flight path and cannot guarantee that there is no enemy on the flight path. Thus, for a given belief Bi, an agent "cheetah102", may (i) strongly believe Bi; (ii) weakly believe Bi; (iii) weakly believe not(Bi); (iv) strongly believe not(Bi).

One key novel heuristic used in CONSA, in weighing presented evidence is that *an agent is expert in its own role*. Thus, if "cheetah102" states to the team that it has not received its firing position, the above heuristic ranks this as strong evidence, since "cheetah102" is considered an expert in its own role of receiving the firing position.

The strength based beliefs enables STEAM to efficiently detect true conflict cases. For instance, while agent "cheetah103" may weakly believe that there is no enemy on the team's flight path, it may be informed by an agent "cheetah102" that it ("cheetah102") has spotted enemy on the flight path. "cheetah103" does not treat this as a conflict, given its earlier weak belief, and strong evidence in favor of the presence of enemy. In contrast, if "cheetah102" had strong belief that it has not received a firing position, it will treat any new information stating that it has received a firing position as a "true" conflict, and consider engaging in negotiations.

CONSA's Approach to Negotiation

CONSA can now be outlined as involving the following steps:

1) *Step 1: Conflict detection*: Agents do not directly accept other agents' beliefs communicated through the CP or other communicative actions (as currently done in STEAM). Instead, they check for conflicts, based on the above belief representation technique. In particular, the communicating agent's belief and any supporting evidence provided is weighed against the receiving agent's own beliefs with possibly contrary supporting evidence. If the communicating agent sends no evidence (if it is considered already available to everyone), the receiving agent uses simple plan-recognition to re-construct this evidence, and check for any conflicts.
2) *Step 2: Decision on negotiation*: Even if conflicts are detected, agents should not automatically engage in protracted negotiations. As the "ball position case" in Section 2 clearly illustrates, the cost of negotiations may outweigh the benefits (e.g., the ball may be shot into the goal by the time the defenders complete their negotiations). CONSA relies on a decision theoretic approach to address this issue. CONSA weighs three alternatives once it detects a conflict. First, an agent can negotiate in detail with its teammate. Second, it can avoid the negotiation, and just accept the teammates' belief. Third, it can refuse to negotiate, but not accept the teammates' belief either (working with its own beliefs instead). Under different

circumstances, the cost and utility of each of these choices will vary. In the "ball position case", the agent will reject the first option as its expected utility is lower than the remaining two options, given its significant cost. Thus, CONSA will terminate, and no negotiations will occur.

3) *Step 3: Begin negotiations*: If an agent does decide to continue with negotiation, it will send arguments to its teammate with whom the conflict is discovered, including the supporting evidence for its own beliefs (that conflict with the sender's beliefs).
   i. 3a: Construct the argument in support of the agent's own belief: build a proof chain to attack the sender's conflicting beliefs.
   ii. 3b: Prune the argument: it's not always necessary to communicate the whole argument to the other agent. Instead, those mutually believed or easily inferred parts can be ruled out for sake of efficiency;
   iii. 3c: Communicate the argument.
4) *Step 4: Continue negotiations*: if the conflict is resolved, stop; else, if conflict detected go back to step 2, but with the original message receiver now the sender, and the sender now the message receiver.

In Step 3, agents construct their arguments in support of their beliefs. In our approach, argumentation is based on Toulmin's argumentation structure [Toulmin 1958]. According to Toulmin, *arguments* consist of *data* and *claims* and *warrants*[1]:

- *Claim*: some conclusion whose merit we are seeking to establish;
- *Data*: the facts we appeal to as a foundation for the claim;
- *Warrants*: the authority for taking the step from the data to the claim.

In our context, claims are the facts the agents individually or mutually believe in. For example in a role relationship situation, the facts that "the role relationship for a team operator T is an AND combination", or "one of the role is fulfilled", or "the top goal has been achieved", etc., are all claims. Warrants are the rules of actions an agent should take based on role constraints, such as :

- Warrant W*: "AND-combination(T) + All-roles-fulfilled(T) $\rightarrow$ achieved(T)"
  That is, for a team operator T, if it is an AND-combination, and all of its roles are achieved , then the team operator is achieved.
- Warrant W**: "OR-combination(T) + Any-role-fulfilled(T) $\rightarrow$ achieved(T)"
  where T is a team operator

In CONSA, data also have supporting structure consisting of warrants and evidence data, recursively, so we can view them as claims also. Thus, henceforth only "leaf" claims (which have no further supporting argument structure) are referred to as *data*.

Consider the case of joint goal termination (case 2, from section 3.2). Here we utilize STEAM's role-constraints to create warrants and guide agents to evaluate propositions and construct counter-arguments.

In the *detection* step (step 1), when an agent receives a joint-goal-completed message, it will first compare the message with the agent's private beliefs. If no conflicts are found, the agent will then backchain the message based on the role relationship constraints (using applicable warrants of role constraints) to infer the sender's beliefs in fulfillment of all relevant roles. For instance, if the message is "achieved(T)", then from the applicable warrant "AND-combination(T) + All-roles-fulfilled(T) $\rightarrow$ achieved(T)" and the claim "AND-combination (T)", the agent can infer back that the sender believes that "All-roles-fulfilled(T)" is true.

Next, the receiving agent can again compare all these Sender's inferred claims to its own existing beliefs. If still no conflicts are found, it will then backchain through such claims (just inferred from last round of backchaining). In the example above, the receiving agent will backchain through the inferred claim "All-roles-fulfilled(T)", to generate other inferred claims for comparison with its own beliefs. In this case, the inferred claims refer to the role fulfillment of individuals or subteams in the operator hierarchy. This backchaining process continues until the agent finds a conflict or reaches the bottom of the tree where the roles are individually executed[2].

When the agent finally finds a conflict between its own beliefs and the backchained claims of the message-sending agent (and it has decided through step 2 to start a negotiation process), it can thus build a complete proof chain to attack the original claim from the whole backchaining path. In most cases, this chain will be too much overhead to communicate completely, so instead of communicating the whole chain to the other agent as a counter-argument, the agent will go through a *pruning* process to rule out the mutually believed or easily inferred components of it and only communicate the remaining claims (step 3).

The following is now a detailed illustration of the above approach in the "proceed" case.

- Mutually known facts in the "proceed" case by all the helicopters beforehand:
  - Warrant w1: AND-combination(T) ^ All roles fulfilled (T) $\rightarrow$ achieved(T)
  - Warrant w2: AND-combination(T) ^ ~ All roles fulfilled(T) $\rightarrow$ ~ achieved(T)
  - Warrant w3: ~ my role fulfilled(T) $\rightarrow$ ~ All

[1] Toulmin actually also has "qualifications" and "rebuttals" as part of his basic argument structure, but we are not using these notions in our approach.

[2] There is an intriguing possibility that if the agent has more trust in the other agent, it may just backchain for a certain number of rounds and accept the claim when no conflicts come out, instead of expanding down each time to the bottom of the role relationship tree, which may be an unnecessary overhead.

roles fulfilled(T)
- Claim c1: AND combination(T)

- Before sending the "proceed" message, the commander takes the following steps:
  - Claim c2: All roles fulfilled(T) (the commander supposes all helicopters received their firing positions and this is mutually believed);
  - Reasoning step: c2 + c1 + w1 → c3 ("achieved(T)");
  - Pruning: w1, c2, c1 (which are all supposed to be mutually believed);
  - Communicate: c3 (to all helicopters).
- "Cheetah102" (helicopter who missed its firing position message) reasons when it gets c3 (the steps shown below are correspondent with the steps in CONSA algorithm):
  - Claim ~c4: ~ my role fulfilled(T);
  - Step1. Detection step (see Figure 2 below):
    1) Check the message (c3) itself and no conflicts found;
    2) Backchaining: c3 + c1 + w2 → c2 (no conflicts found);
    3) Backchaining one more level: c2 + w3 → c4 (conflict found);
  - Step 2. Decided to engage in negotiation;
  - Step 3. Begin negotiation:
    3a. Arguments building step: the attacking chain: ~c4 + w3 → ~c2, ~c2 + c1 + w2 → ~c3;
    3b. Pruning: w2, w3 (which are supposed to be mutually believed);
    3c. Communicate: c4 → ~c2 → ~c3 (back to the commander)
- Reasoning process of the commander afterwards:

  First, evaluate supporting and weakening arguments of "top goal achieved"(c3). Next, decide to accept the negative branch (~c3) since c4 has more weight in that its the area of expertise of "cheetah102" to determine if its own role is fulfilled.

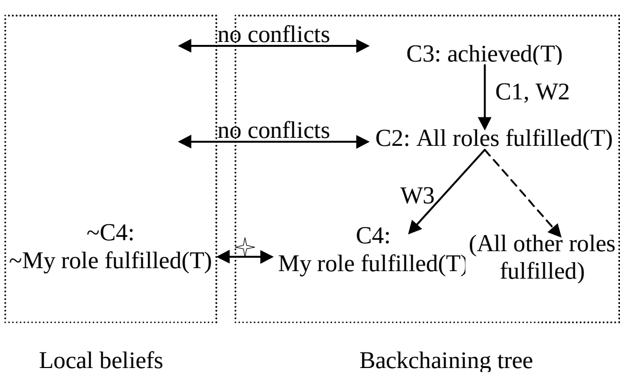

Figure 2: Reasoning process of "cheetah102" when it received the "proceed" message

The above chart shows how "cheetah102" detects the conflicts by backchaining down the role hierarchy. Here the left box represent the local belief base of "cheetah102" and the right box is its backchaining trace, where arrows illustrate backchaining steps, which can be taken as "supported by".

Description of Preliminary Implementation

We have at present partly implemented the above CONSA approach in the Soar problem-solving architecture [Newell 1990]. This implementation currently exploits the role relationship structures associated to every team operator in STEAM. It fully implements step 1 and 3 of the CONSA approach and at present only address the conflicts occurring in the termination phase of a joint intention, by having the agents examine the role fulfillment status through the expanded role relationship tree. It however cannot resolve joint-goal-initiation and individual-goal conflicts as yet.

Our implementation has enabled agents to begin negotiations in the running "proceed case". Currently we have written nearly thirty Soar rules and below is part of the Soar outputs from CONSA's execution trace in the "proceed" case. The trace shows that the helicopter agent who misses the message is now taking steps to detect the conflict and start up a negotiation process. Here cheetah101 (the commander) and cheetah102 are call signs of two helicopters, and numbers like "130" etc. are decision cycle numbers(steps) in Soar. "O58", "O59" etc. are all executed Soar operators. For example, in step 130, cheetah102 executes the operator O58 to begin an inconsistency control, i.e. to engage in the process of detecting and resolving a conflict in the team.

```
......
Begin inconsistency checking...
130: O: O58 inconsist-control
131: ==S: S23 (operator no-change)
132: O: O59 make-decision
Decide to negotiate
133: O: O60 compare-message
No conflicts in saved message itself.
Doing backchaining...
134: O: O61 backchain
135: O: O62 compare-modeling Conflicts with my belief
role-fulfilled: *no* detected after backchaining.
136: O: O66 communicate-disagreement
Communicate: cheetah102 role-fulfilled *no* to
cheetah101
137: ==S: S24 (operator no-change)
138: O: O67 wait
139: O: O68 wait
......
```

We can see from the above execution trace that after helicopter "cheetah102" decided to start a conflict resolving process (step 2), it first check the received message itself. After finding no conflict there, the agent began backchaining and comparison (step 1). When it found the clash with its beliefs, the agent then communicated its disagreement back to the commander and waited for its response (step 3).

Related Work

We begin discussion of related work by focusing on the closely related research on argumentation-based negotiation. We will also later examine game-theoretic approaches to negotiation.

Chu-Carroll and Carberry [Chu-carroll and Carberry 1996] describe a computational model that captures the collaborative planning process in a recursive *Propose-Evaluate-Modify* cycle of actions. Their model is able to detect potential conflicts regarding both proposed actions and proposed beliefs, and to initiate collaborative negotiation subdialogues to resolve the detected conflicts. They also identify the focus of modification, and select appropriate evidence to justify an intended mutual belief. Parsons and Jennings [Parsons and Jennings 1996] draw upon a logic of argumentation to devise a system of argumentation and use it to implement a form of dialectic negotiation. In their context, an argument is a sequence of logical steps indicating support or doubt of a proposition. They have a function of *flattening*, which can measure the set of arguments into some metric of how favored the proposition is, by determining which *class of acceptability* the arguments belong to. **F**reeman and Farley [Freeman and Farley 1993] present elements of a theory of argumentation as a method for providing decision support and justification for plausible reasoning in weak theory domains. Their work is based on Toulmin schema. In this context they describe a theory of argument as dialectical process, where the format of a two-sided argument is used to intertwine the strengths and weakness of support for competing claims.

One key novel aspect of our work, which differentiates it from the above approaches is that we are focusing on collaborative negotiation and are thus able to use rules of collaboration (teamwork model) as warrants. This enables CONSA to begin to build a reusable system of argumentation.

A second difference in CONSA from the above approaches (except for Freeman and Farley's work) is its basis in Toulmin's argument structure, which facilitate the guidance for the agents to detect conflicts and generate arguments. With reference to Freeman and Farley's work, they have a more varied set of warrants with degrees of certainty than we have, which may not be always sound. CONSA does not allow anything other than deductive reasoning, although the data (e.g. of whether roles are fulfilled) is often uncertain in our case.

Third, instead of focusing just on argumentation (as in a legal argumentation domain in Freeman and Farley's work), we embed negotiation in functioning systems where agents must engage in teamwork. Thus, our agents are behaving in a team context and are designed to execute team tasks. They must detect those conflicts that are critical to team performance, and negotiate only about those issues. In particular, agents must not argue all the time.

Additionally, as mentioned, we take into account the cost of negotiation itself and utilize a decision-theoretic approach to make selective negotiation. Furthermore, we have begun to categorize conflict types in the context of the principled teamwork model of STEAM.

Rosenschein and Zlotkin's research [Rosenschein and Zlotkin 1994] is representative of a growing work on this topic based on game theory. As mentioned before, they focus on self-interested utility maximizing agents in contrast with our collaborative agents in a team setting. They design conventions to ensure agents to act in certain ways, e.g. their Vickrey's auction mechanism ensure that agents will bid without deception. However, they do not focus on the representation and reasoning processes in negotiation. Also, they assume that agents have complete knowledge of the utility matrix and the precise utility function, which is hard to obtain in a dynamic context where agents sometimes have discrepancies in utility assessment. In addition although they focus on techniques to choose a best action from multiple of available choices based on the utilities and so forth, how agents can plan and generate such actions in the first place is still a big question.

Summary and Future Work

We have developed techniques to enable agents to engage in flexible teamwork, where the teamwork may be in service of applications such as distributed planning, as well as multi-agent plan execution, or multi-agent design, or other such team activities. One key problem in teamwork is collaborative negotiations to resolve conflicts in agents' beliefs. To address the problem, we have designed and partly implemented a system CONSA for collaborative negotiation. The novelty of our approach is its basis in model-based teamwork, which enables us to distinguish between different types of conflicts in teamwork, and guide agents in taking appropriate steps to resolve such conflicts.

Our preliminary framework is based on Toulmin's argumentation structure and is exploiting the role relationships and task structures in STEAM teamwork model, which enables a generalized and reusable argumentation facility. One other novel aspect of our approach is that it considers the appropriateness and cost of negotiation/argumentation, which is non-trivial in complex, dynamic domains.

Although we have currently assumed that the entire role relationship structure is available to all of the agents thus the agents can only argue about the role performance data, we could however assume that not everything about other roles and how their combinations work are known to all agents. This could be an interesting source of argumentation as well in the future.

Acknowledgment

This research is supported in part by DARPA Award No. F30602-98-2-0108, and in part by AFOSR Contract No. F49620-97-1-0501.

References

[1] Chu-Carroll, J., and Carberry, S. 1996. *Conflict detection and resolution in collaborative planning*. In M. Wooldridge, J. Muller, and M. Tambe, editors, Intelligent Agents, Volume II: Lecture Notes in Artificial Intelligence 1037. Springer-Verlag, Heidelberg, Germany.

[2] Cohen, P. R., and Levesque, H. J. 1991. *Teamwork*. Nous, 35.

[3] Firby, J. 1987. *An investigation into reactive planning in complex domains*. In Proceedings of the National Conference on Artificial Intelligence (AAAI).

[4] Freeman, K., and Farley, A. 1993. *Toward Formalizing Dialectical Argumentation.* In Proceeding of the Fifteenth Annual Conference of the Cognitive Science Society.

[5] Grosz, B., and Kraus, S. 1996. *Collaborative plans for complex group actions*. Artificial Intelligence, 86:269-358.

[6] Jennings, N. 1995. *Controlling cooperative problem solving in industrial multi-agent systems using joint intentions*. Artificial Intelligence, 75.

[7] Kitano, H., Tambe, M., Stone, P., Coradesci, S., Matsubara, H., Veloso, M., Noda, I., Osawa, E., and Asada, M. 1997. *The robocup synthetic agents' challenge*. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).

[8] Mouaddib, A. 1997. *Progressive negotiation for time-constrained autonomous agents*. In Proceedings of the International Conference on Autonomous Agents (Agents'97).

[9] Newell, A. 1990. *Unified Theories of Cognition*. Harvard Univ. Press.

[10] Parsons, S., and Jennings, N. R. 1996. *Negotiations through argumentation--a preliminary report.* In Proceedings of the International Conference on Multi-agent Systems, 267-274.

[11] Rosenschein, J. S. and Zlotkin, G. 1994. *Designing conventions for automated negotiation*. AI Magazine, 15.

[12] Tambe, M. 1997. *Towards flexible teamwork*. Journal of Artificial Intelligence Research (JAIR), 7:83-124.

[13] Tambe, M., Johnson, W. L., Jones, R., Koss, F., Laird, J. E., Rosenbloom, P. S., andSchwamb, K. 1995. *Intelligent agents for interactive simulation environments*. AI Magazine, 16(1).

[14] Toulmin, S. 1958.*The uses of argument*. Cambridge: Cambridge University Press.