A decision-theoretic approach to resourcing plans in joint missions

Chukwuemeka David Emele, Timothy J. Norman, and Frank Guerin
Computing Science Department
University of Aberdeen, Aberdeen
United Kingdom. AB24 3UE
{*c.emele, t.j.norman, f.guerin*}*@abdn.ac.uk*

Simon Parsons
Department of Computer and Information Science,
Brooklyn College
City University of New York
2900 Bedford Avenue, Brooklyn. 11210 NY
parsons@sci.brooklyn.cuny.edu

***Abstract*—In joint missions, coalition members are often guided by the social characteristics (e.g policies/norms) of the nations or organisations they represent. Developing and resourcing plans in such collaborative context is a complex problem, and requires an understanding of the policy and resource availability constraints under which others operate. We present a novel combination of argumentation, machine learning and decision theory for deciding who to talk to, and what information needs to be revealed to convince others during collaborative activities. In a set of experiments, we demonstrate that coalition members can improve their resourcing strategies and performance while reducing communication overhead by building accurate models of others' policies and resource availabilities, and using heuristics based on decision theory.**

I. Introduction

The development and coordination of coherent and efficient plans for joint action is an important area of research interest in distributed artificial intelligence [1] and multi-agent systems [2], [3]. An important aspect of the problem is how these plans can be resourced. Resourcing plans is a complex task on its own, and becomes even more challenging when policies (or norms) are involved. By policies, we mean the operating procedures that govern the behaviour of coalition members and constrain under what circumstances they may provide some resource to another member, and conversely when they are prohibited from doing so. Coalition members often operate under such constraining policies while carrying out tasks assigned to them. Policy constraints are distinct from what we refer to as resource availability constraints; if a resource is not available it is not physically possible to provide it to any other coalition member, whereas a policy may, for example, forbid the provision of a resource whether or not it is available. Agent support in these situations can help to overcome some of the challenges and lead to a more efficient collaboration. In particular, software agents can enforce the correct protocol for team behaviour [4]; ensure timely delivery of important data [5], [6]; filter messages [7]; monitor progress in the performance of a task [8] and coordinate team actions [9], [10], [11].

Furthermore, agents can give guidance in making policy-compliant decisions [12]. In such scenarios, software agents could act either as *critic* or *censor* agents. A *critic* agent detects policy violations of coalition members in the course of communication between them and inform the sender about the set of policies violated. The sender can then choose whether to adhere to the advice or overrule the agent's advice. On the other hand, a *censor* agent interferes with the communication by removing parts (or the whole) of the exchanged messages that contain policy violations. This prior research focuses on advising coalition members on violations of their own policies. An important and open question, however, is how can agents advise coalition members about the policies of others? If agents are to provide this level of support then agents must be able to develop accurate models of others' policies.

To do this, we propose an argumentation-based framework, that uses machine learning to model the policies and resource availabilities of others. Further, we present a decision-theoretic model to aid in deciding who to talk to and what information needs to be revealed if another coalition member is to provide that resource. We describe an experimental framework and present results of our evaluation in a resource provisioning scenario [13].

In the research presented in this paper, we intend to validate the following hypotheses: (1) Allowing agents to exchange arguments about their policies and resource constraints during dialogue and incorporating appropriate machine learning and decision-theoretic model will lead to a better performance (that is, higher utility) than when there is no such combination of techniques. (2) The use of decision-theoretic heuristics will lead to significant reduction in communication overhead.

The remainder of this paper is organised as follows: In Section II we briefly discuss negotiating with evidence. Section III presents the reasoning framework of the agent and Section IV describes our simulation environment. Experimental results are reported in Section V and Section VI discusses related work and future direction. The paper is concluded in Section VII.

II. Argumentation-based Negotiation with Evidence

Here, we present the argumentation-based negotiation protocol employed in this framework, which will be used in guiding the negotiation process and for obtaining additional evidence from the interaction.

Negotiating for resources

The negotiation for resources, as shown in Figure 1, takes place in a turn-taking fashion. The negotiation dialogue starts with an agent, the *seeker*, sending a request to another agent, the *provider*, for the use of some resources needed to fulfill a plan. If the *provider* agent has the requested resource in its resource pool and it is in a usable state then it checks whether there is any policy constraint that forbids it from providing the resource to the *seeker* or not. If the *provider* agent needs more information from the *seeker* in order to make a decision, the *provider* agent would ask questions regarding the use of that resource. This is the information gathering stage. The information gathering cycle will continue until the *provider* has acquired enough information (necessary to make the decision), or the *seeker* refuses to provide more information and the negotiation ends. There is a cost attached to the revelation of private information to another agent. An agent might refuse to reveal a piece of information if doing so is expensive [14]. The decision-theoretic model provides a metric (based on expected utility) for determining whether to reveal more information or not.

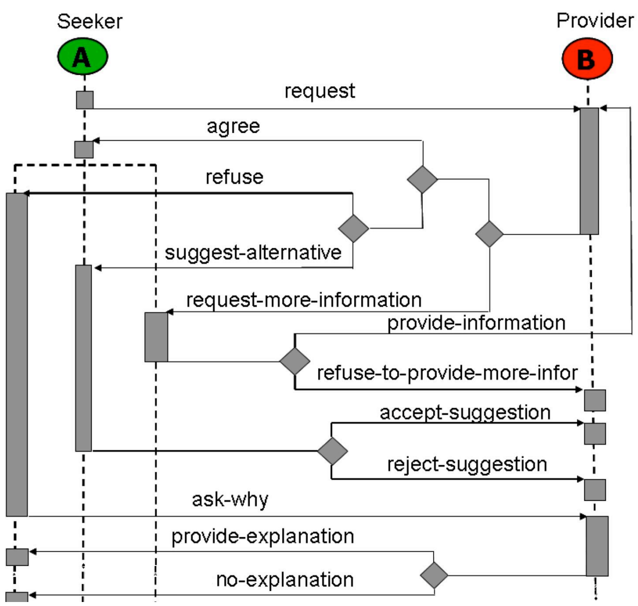

Fig. 1. The negotiation protocol.

The *provider* agent releases the resource to the *seeker* agent if there is no policy that prohibits the *provider* agent from doing so. Otherwise, the *provider* agent offers an alternative resource (if there are no policies that forbid that line of action and the alternative resource is available). When an alternative resource is suggested by the *provider* agent, the *seeker* agent evaluates it. If it is acceptable, the *seeker* agent accepts it and the negotiation ends. Otherwise, the *seeker* agent refuses the alternative. In principle, this cycle may be repeated until an alternative is accepted or all alternatives are exhausted and the negotiation ends. For the sake of simplicity, we allow only one alternative to be suggested per request.

Evidence from dialogue

The suggestion of alternative resources is a positive evidence that the *provider* agent does not have any policy that forbids the provision of the alternative resource to the *seeker*. In addition, it provides an evidence that the alternative resource is also available. If there is a policy constraint that forbids the provision of the resource, or the resource is not available then the *provider* agent will refuse to provide the resource to the *seeker* agent. From the *seeker*'s perspective, the refusal could be as a result of policy constraint or unavailability of resource. In order to disambiguate which of these constraints are responsible for the refusal, the *seeker* agent switches to argumentation-based dialogue. The *seeker* agent asks for explanations for the refusal so as to gather further evidence and thereby identify the underlying constraints. The *provider* could respond with some (or no) explanations and the negotiation ends.

Following the argumentation-based negotiation protocol described earlier, the agents could ask for more information (with respect to a request or the response to a request), which indicates what constraints others may be operating within. For instance, let us assume that a *provider* agent has a policy that forbids it from providing a *helicopter* to any *seeker* agent that intends to fly it in an area covered with *volcanic ash*. Then, whenever a *helicopter* is requested the *provider* agent will ask for more information to ascertain that the *seeker* does not intend to deploy the *helicopter* in an area with *volcanic clouds*. From the *seeker* agent's point of view, this pattern of behaviour by the *provider* agent serves as extra evidence for the kind of policies under which the *provider* operates. This evidence could be exploited by the *seeker* in preempting what extra information the provider might want to ask for. To this end, the seeker may provide the location where the *helicopter* is to be deployed along with the request.

Dialogue examples

To illustrate the sorts of interaction between agents, consider the following examples. Let agents x and y be the *seeker* and *provider* agents respectively. Suppose we have an argumentation framework that allows agents to ask for and receive explanations as in Scenario 1 (Table I, *lines* 9 *and* 10), suggest alternatives as in Scenario 2 (*line* 8) or ask for more information regarding the attributes of a request (*lines* 2 *to* 7) in both scenarios, then agent x can gather additional information regarding the policy rules guiding y concerning provision of the resources involved.

In the foregoing example, if the *helicopter* is intended to be deployed in an area with volcanic clouds then the *provider* is forbidden from providing the resource but might suggest a ground vehicle (e.g. *jeep*) to the *seeker* if there are no policies and/or availability constraints on that resource. Furthermore, whenever a *seeker* agent's request is refused then the *seeker* agent will ask for explanations/justifications for the refusal. This additional evidence is beneficial, and could be used to improve the model and, hence, the quality of the decisions made in future encounters.

TABLE I
DIALOGUE EXAMPLES.

#	Scenario 1	Scenario 2
1	x: Can I have a *helicopter*?	x: Can I have a *helicopter*?
2	y: What do you need it for?	y: What do you need it for?
3	x: To transport relief materials.	x: To transport relief materials.
4	y: To where?	y: To where?
5	x: A refugee camp in Iceland.	x: A refugee camp in Iceland.
6	y: Which date?	y: Which date?
7	x: On Friday 16/4/2010.	x: On Friday 16/4/2010.
8	y: No, I can't provide you with a *helicopter*.	y: I can provide you a *Jeep* to transport the materials
9	x: Why?	x: I accept a *Jeep*.
10	y: There is volcanic eruption, and I am not permitted to release a *helicopter* in such circumstances.	

III. REASONING ABOUT POLICIES

Policies from different coalition members may vary and may conflict. Such conflicts may affect collaboration. In the face of policy differences, collaborative activities become more complicated, especially at the planning stage [15]. For example, a coalition member may be *prohibited to disclose the source of information regarding an imminent attack (because the information was obtained from a highly classified sensor)*; another member may be *obliged to verify the reliability of the source of an intelligence before acting*, in which case such a member will seek to know the source of an intelligence.

In our work, we are particularly interested in individual policies, which are private to that individual member or subset of the coalition (See Figure 2). Our approach exploits argumentation-based dialogues [16], [17] in negotiating for resources required to enact a plan. In particular, agents exchange arguments (in the form of explanations and justifications) for the decisions they make. Such arguments often serve as evidence for the underlying policies they operate within.

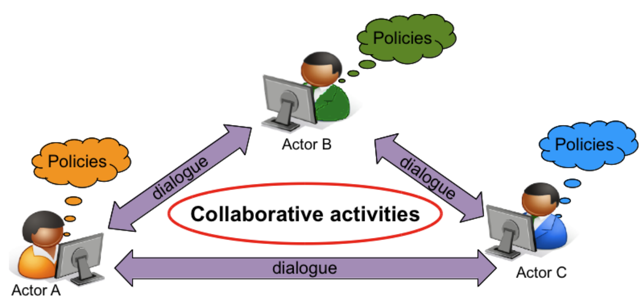

Fig. 2. An overview of the system.

Resourcing plans

Agents in this framework have policies that govern how resources are deployed to others. These policies are based on a number of features which characterise the prevailing circumstances under which an agent operates. The features can take on different values. Our concept of a resourcing policy maps a set of features into an appropriate policy decision.

Let the features that characterise the resourcing policy of agents, denoted as C be a tuple defined as follows:

$$C = (\mathbf{O}, \mathbf{R}, \mathbf{P}, \mathbf{L}, \mathbf{D})$$

Where

- Organisation, denoted by **O** refers to the country/organisation the agent represents.
- Resource, denoted by **R** generally denote physical equipment, capabilities or information that are required to carry out a plan.
- Purpose, denoted by **P** indicates the purpose for which the resource is requested.
- Location, denoted by **L** denotes the particular location or zone where the resource is to be deployed.
- Day, denoted by **D** refers to the day the resource is to be deployed.

Definition 1 ***(Resourcing policy)*** A resourcing policy, denoted as Π_C governs how resources are deployed to others.

$$\Pi_C : \mathbf{O} \times \mathbf{R} \times \mathbf{P} \times \mathbf{L} \times \mathbf{D} \rightarrow \{\text{grant, deny}\}$$

In other words, a resourcing policy maps a tuple consisting of the policy features into a decision ($grant$ or $deny$). $Grant$ means that the policy allows the agent to provide the resource when requested while $deny$ implies that the policy prohibits the agent from providing the resource.

Examples of policies that a *provider* may be operating under may include:

$\mathbb{P}$**1:** *You are* ***permitted*** *to release a* $helicopter$ *to a coalition partner if the* $helicopter$ *is required for the purpose of transporting relief materials.*

$\mathbb{P}$**2:** *You are* ***prohibited*** *from releasing a* $helicopter$ *to a coalition partner if it is to be deployed in an area with volcanic clouds no matter the purpose.*

$\mathbb{P}$**3:** *You are* ***permitted*** *to release a jeep to a coalition partner even if it is to be deployed in an area with volcanic clouds.*

In the next section, we discuss in detail a rule-based approach to learning policies.

Learning policies from evidence

Since policies (or norms) guide the way coalition members act by providing rules for their behaviour it makes sense to use a rule-based learning algorithm. We take inspiration from [18] to define policies as rules based on the actions of an agent (or the attributes of its request).

Following from Pearl [18], a policy can be expressed as a rule, of the form $\Pi_C = \langle C, action\rangle$, which states that if the prevailing circumstance of an agent satisfies the context C, then the agent should perform $action$. Using the features identified above, an instance of an agent's context C_i is an ordered list comprising of the affiliation of the agent $o \in O$, resource required $r \in R$, the purpose $p \in P$, the location $l \in L$ and the day $d \in D$. In other words, an instance of an agent's context C consist of the attributes of the request, and $action$

is the decision as to whether the request should be granted (in which case the resource is provided) or denied. That is,

$$\Pi_C(o, r, p, l, d) = \{grant\}$$

or

$$\Pi_C(o, r, p, l, d) = \{deny\}$$

For instance, policy example $\mathbb{P}1$ cited in the previous section can be written as:

$$\Pi_C(uk, helicopter, \textit{transport-relief, any, any}) = grant$$

The term any is used to denote that the attribute in question is permitted to take on any of the possible values. A policy instance will be activated and the corresponding action taken whenever the set of attributes in the test instance matches a policy instance in the *provider's* policy database. The above representation is the basis for capturing training examples and test instances of the policy rules. These policy rules are then learned systematically using machine learning techniques.

Although in the experimental study discussed here we use sequential covering, other machine learning techniques may be adopted. In earlier experiments, however, sequential covering proved to perform very well in building a good model of others' policies rapidly. We investigated three classes of machine learning algorithms, namely: decision tree learning (using C4.5), instance-based learning (using k-nearest neighbours), and rule-based learning (using sequential covering). We discovered that the sequential covering approach outperformed the other two approaches [19].

Sequential covering [20] is a rule-based learning technique, which constructs rules by sequentially covering the examples. The sequential covering algorithm, SC for short, is a method that induces one rule at a time (by selecting attribute-value pairs that satisfy the rule), removes the data covered by the rule and then iterates the process. An example of an attribute-value pair is $\langle O, uk\rangle$, and this means the value of attribute O is uk. SC generates rules for each class (e.g., *grant* or *deny*) by looking at the training data and adding rules that completely describe all tuples in that class. For each class value, rule antecedents are initially empty sets, augmented gradually for covering as many examples as possible. Figure 3 outlines the sequential covering algorithm in pseudo-code.

Algorithm: Sequential Covering Algorithm

Input the training data (D) and the classes (C)
For each class $c \in C$; where $C = \{grant, deny\}$
Initialise E to the instance set
Repeat
  Create a rule R with an empty left-hand side (LHS) that predicts class c:
  Repeat
    For each $(Attribute, Value)$ pair found in E
      Consider adding the condition $Attribute = Value$ to the LHS of R
      Find $(Attribute, Value)$ that maximises ϕ (ϕ is the probability of occurrence of c for each $(Attribute, Value)$ pair. Break ties by choosing the condition with the largest occurrence of c)
      Add $Attribute = Value$ to R
  Until R is perfect (or no more attributes to use)
  Remove the instances covered by R from E
Until E contains no more instances that belong to c

Fig. 3. Sequential Covering Algorithm.

Decision-theoretic approach

Based on the model of others' policies that has been learned over time, the $seeker$ agent employs decision theory in choosing who among the potential providers to talk to with a view to resourcing a plan. Here, we formalise the decision-theoretic model that is used in this work. Let X be the set of $seeker$ agents such that $x \in X$, and Y be the set of $provider$ agents such that $y \in Y$. Let R be the set of resources such that $r \in R$, and $Tasks$ be the set of tasks such that $t \in Tasks$. The set of information is denoted as I such that $i \in I$.

Definition 2 ***(Benefit)*** The benefit gained in resourcing a task $t \in Tasks$ with resource $r \in R$, denoted as $benefit(r, t)$ is given as:

$$benefit : R \times Tasks \rightarrow \mathbb{R}$$

In other words, benefit is the satisfaction derived or value added for obtaining r to be used in resourcing t. This satisfaction is the *seeker* agent's valuation of r for the task instance considered and this value varies from task to task.

Generally, *seeker* agents receive benefits from obtaining a resource and incur costs in providing information to *provider* agents. In some domains, there may be benefit to the provider in terms of some kind of monetary transfer between them. In this case, the benefit to the seeker is simply the inherent benefit of acquiring the resource minus the cost of revealing information to the *provider* agent.

Definition 3 ***(Unit Cost)*** The cost of revealing a specific piece of information, $i \in I$, to a specific agent, $y \in Y$, denoted as $cost(i, y)$ is given as:

$$cost : I \times Y \rightarrow \mathbb{R}$$

In our framework, cost is an abstraction that captures some notion of, for example, risk in revealing information about private plans.

Definition 4 ***(Information Cost)*** The cost of revealing pieces of information in the set $\mathcal{I}$ to agent y, denoted as $Cost(\mathcal{I}, y)$ is the summation of the cost of revealing each $i \in \mathcal{I}$ to agent y. That is,

$$Cost(\mathcal{I}, y) = \sum_{i \in I} cost(i, y) \quad (1)$$

This cost depends on y, but not on the task/resource. In other words, the cost of revealing a piece of information to agent y is constant irrespective of the task or the resource. We assume that the cost of revealing the resource being requested is zero.

Definition 5 ***(Information Required)*** The set of information required for y to make available resource r according to our model of their policy is denoted as λ.

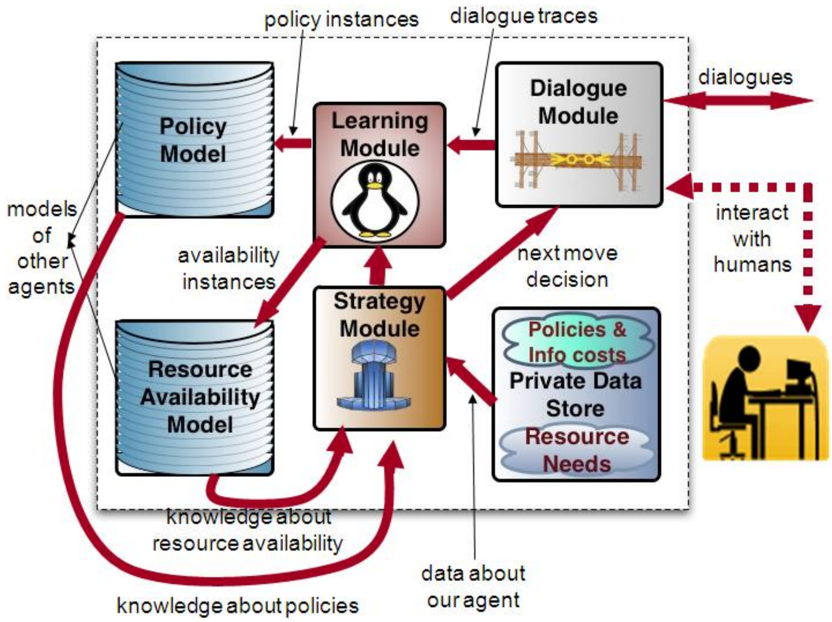

Fig. 4. Architecture of the framework to support resourcing a plan.

Note that this depends upon r because this represents the minimum information needed to convince an agent to provide r according to x's model of y. If there is no way to convince y (that is, if we predict that there is no way to convince the provider to provide the resource) then λ is the empty set.

From our earlier example, if we predict that agent y needs to know the purpose and day the $helicopter$ will be used then: $\lambda(helicopter, y) = \{$**purpose, day**$\}$.

Let $Pr(YesP|y, r, \mathcal{I})$ be the probability of a Yes response given we ask agent y for resource r and the information required is $\mathcal{I}$ (that is, $\lambda(r, y) = \mathcal{I}$). Let $Pr(Avail|y, r)$ be the probability of the resource being available given we ask agent y for resource r. We note that $\mathcal{I} \in 2^I$. These probabilities are computed from data gathered from previous encounters with providers.

Definition 6 ***(Joint Probability)*** The probability that the policy is Yes and the resource is available given we ask agent y for r and $\lambda(r, y) = \mathcal{I}$, denoted as $Pr(Yes|y, r, \mathcal{I})$ is given as:

$$Pr(Yes|y, r, \mathcal{I}) = Pr(YesP|y, r, \mathcal{I}) \times Pr(Avail|y, r) \quad (2)$$

Definition 7 ***(Utility Function)*** Every seeker agent x has a utility function $u_y^{r,t,\mathcal{I}} : Y \times R \times Tasks \times 2^I \to \mathbb{R}$.

The expected utility, $E(u_y^{r,t,\mathcal{I}})$ that agent x derives from acquiring a resource r for a task t from agent y requiring the revelation of $\mathcal{I}$ to y is computed as follows:

$$E(u_y^{r,t,\mathcal{I}}) \;=\; benefit(r,t) \times Pr(Yes|y, r, \mathcal{I}) - Cost(\mathcal{I}, y) \quad (3)$$

In order to select the agent that has the right balance between cost due to revealing information and likely provision of the resource, we need a selection strategy.

Definition 8 ***(Selection Strategy)*** The selection strategy an agent uses to select which *provider* to ask for the provision of a resource for a given task is a function y_{opt}^t, computed as:

$$y_{opt}^t = \arg \max_{y \in Y, \mathcal{I} = \lambda(r,y)} E\left(u_y^{r,t,\mathcal{I}}\right) \quad (4)$$

Using this decision-theoretic heuristic, we can make an informed judgement about whether or not to even bother asking anyone at all. For instance, if E$(u_{y_{opt}}^{r,t,\mathcal{I}}) \leq 0$ then one could argue that a rational *seeker* agent should not attempt to acquire the resource as it could yield a negative or zero utility. In this evaluation, the selection strategy only chooses the *provider* with the highest utility. Here, we assume that the *seeker* makes a single decision about which provider to choose, irrespective of whether it fails or succeeds.

IV. Simulation Environment

We implemented a simulation environment for agent support in team-based problem solving, and integrated our argumentation, machine learning and decision theoretic model into the framework. The architecture is sketched in Figure 4. Each agent has three main modules: the dialogue module, the learning module and the strategy module. The dialogue module embodies the dialogue controller, which handles all communication with other agents (and possibly humans) in the domain. The dialogue module sends/receives messages to/from other agents. If an agent is playing the role of a *seeker* then the dialogue module sends out the request for resources. On the other hand, if the agent is a *provider* then the dialogue module receives a request and passes it on to the learning module.

The learning module reasons over the dialogue and attempts to build models of other agents' policies and resource availabilities based on arguments exchanged during encounters. The arguments includes the features that an agent requires in order to make a decision about providing a resource or not. For example, following from [21], a *provider* agent B may need to know what the purpose for requesting a *helicopter* is before deciding whether to release the *helicopter* or not. Also, the decision of B after the purpose has been revealed will also be learned for future interactions. After this, the policy and resource availability models are updated accordingly.

The strategy module looks up policy and resource availability models, and selects which potential provider yields the highest utility (See Section III). The private data-store acts as a repository where an agent stores its private information (e.g. cost of revealing information to other agents) and constraints.

The simulation environment allows us to generate multiple *providers* with randomised policies, *seeker* agents with randomised initial models of the policies of *providers* in the simulation and randomised problems for the *seeker* to solve (that is, random resource requirements). The *seeker* predicts (based on the model of the *provider*) whether the *provider* has a policy that forbids/permits the provision of such resource in that context. The *seeker* requests the required resource from the *provider* agent, and based on the meta information learned so far the *seeker* could preempt the decision attributes of the *provider* and provide them ahead of time.

V. Experiments and Results

In a series of experiments, we show how decision theoretic, machine learning techniques and argumentation can support agents engaging in collaborative activities, increase their predictive accuracy, reduce communication overhead, and increase utility; hence improve their performance. The experiments show that agents can effectively and rapidly increase their predictive accuracy of the learned model through the use of dialogue. Also, by effectively preempting the questions and information requirements of the provider, the seeker can reduce the number of messages required to complete a task. Combining the decision theoretic with a more accurate and stable model of the learned model will mean that the utility gained from completing tasks will increase.

The scenario adopted in this research involves a team of five software agents (one *seeker* and four *provider* agents) collaborating to complete a joint activity in a region over a period of three simulated days. The region is divided into five locations. There are five resource types, and five purposes that a resource could be used to fulfill. This results in 375 possible configurations. A task involves the *seeker* agent identifying resource needs for a plan and collaborating with the *provider* agents to see how that plan can be resourced. In the control condition, a simple lookup table was used to record the outcomes from past episodes so that in future episodes the *seeker* can lookup past outcomes.

Results

Experiments were conducted with *seeker* agents initialised with random models of the policies of *provider* agents. 100 runs were conducted for each case, and tasks were randomly created during each run from 375 possible configurations.

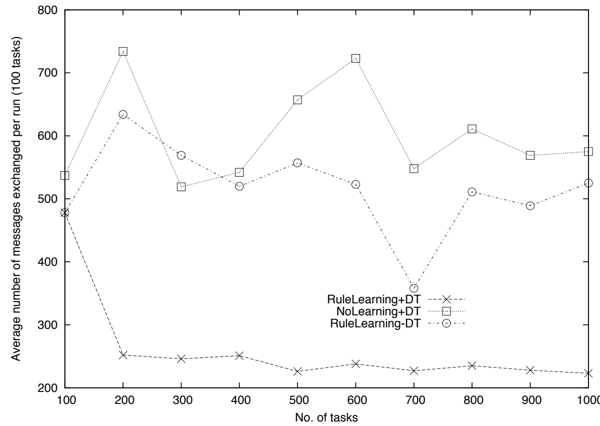

Fig. 5. Graph showing the number of messages exchanged.

Table II illustrates the effectiveness of combining argumentation, machine learning and decision theoretic model in resourcing plans. It shows the average number of messages exchanged and the standard deviations for each of the approaches, namely: Decision theoretic model combined with rule learning with the aid of argumentation-derived evidence (*RL+DT*), Decision theoretic model combined with lookup table with the aid of argumentation-derived evidence (*NL+DT*), and Rule learning with the aid of argumentation-derived evidence alone (*RL-DT*). In each case, the model of others' policies is recomputed after each set of 100 tasks. The number of messages exchanged in *RL+DT* was consistently and significantly lower than those in the other two cases. For instance, just after 200 tasks, the communication overhead has reduced to almost 2 messages per task. The reason for this, after detailed analysis of the data, is because the $seeker$ is (1) able to make an informed decision concerning which provider to approach for a given resource, and (2) able to preempt the information requirements of the provider and thereby present it without having to be asked. Figure 5 gives a graphical illustration of these results.

In the *NL+DT* case, the seeker has the opportunity to use the decision theoretic model to select which provider is more likely to provide the resource. However, since there is no learning component in this configuration, the *seeker* checks the lookup table to see if there is an exact match recorded in past encounters. If an exact match is found, it looks up the outcomes recorded in that encounter and uses them to compute the costs and probabilities. If no exact match is found then the *seeker* assumes a probability of 0.5, and that all the information will be revealed. The results, thus, show that the number of messages per task varies between around 7 and 5. The reason for this poor performance is because the $seeker$ in this setup is unable to build a reliable model of the *provider* agent's policies and resource availabilities. Similarly, in the *RL-DT* approach, the number of messages exchanged per task rose to as high as 6, which confirms that adopting machine learning and argumentation alone might achieve greater utility

TABLE II
AVERAGE NUMBER OF MESSAGES PASSED PER RUN (100 TASKS) AND STANDARD DEVIATION

Case \ Task	100	200	300	400	500	600	700	800	900	1000
NL+DT	537±8.4	734±5.4	519±7.7	542±5.2	657±8.1	723±4.6	548±6.3	611±7.4	569±7.6	575±7.1
RL-DT	478±8.1	634±6.8	569±6.4	520±5.5	557±5.8	523±5.1	358±4.7	511±4.8	489±4.6	525±4.5
RL+DT	478±8.1	252±4.7	246±4.3	251±3.7	226±3.4	238±3.5	227±3.3	235±2.7	228±2.4	223±2.3

(that is, agents may reach their goals) but may not reduce the number of arguments exchanged. Tests of statistical significance were applied to these results. The regression analysis for the average number of messages exchanged shows that as the number of tasks increases, the number of messages exchanged in the *RL+DT* scenario consistently converges with a 95% confidence interval. On the other hand, with significance $p > 0.05$, there is no statistical significance as to whether *NL+DT* and *RL-DT* converge or not respectively. These results show that combining argumentation, machine learning and decision theoretic model can help reduce the communication overhead of agents in resourcing plans.

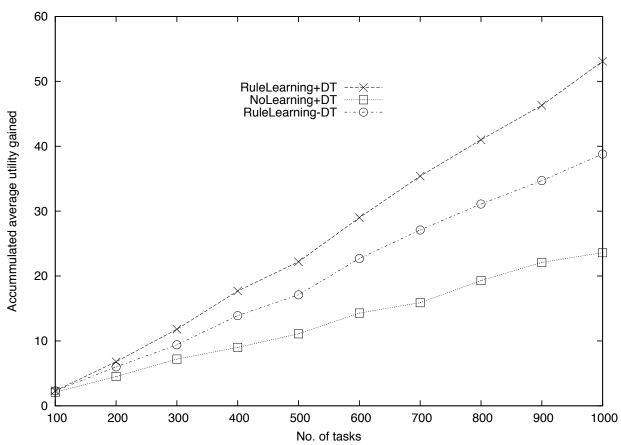

Fig. 6. Graph showing the cumulative average utility of the $seeker$.

Figure 6 compares the cumulative average utility of the seeker agent in the three configurations, namely: *RL+DT*, *NL+DT* and *RL-DT*. The results show that the three configurations recorded increase in utility. However, *RL+DT* and *RL-DT* significantly and consistently outperforms *NL+DT*. This, we believe, is because the rule learning algorithm with the aid of argumentation-derived evidence built more accurate models of others' policies and resource availabilities than the control condition in *NL+DT*. Similarly, *RL+DT* constantly and consistently outperforms *RL-DT*. The reason for this is the fact that *seeker* agents in the *RL-DT* are not equipped with the decision theoretic heuristics which helps to maintain the right balance between cost due to revealing information and likely provision of the resource. After 1000 tasks the cumulative utility of the $seeker$ in *NL+DT*, *RL-DT* and *RL+DT* reached 23, 38 and 53 units respectively. Furthermore, the regression analysis for the cumulative average utility shows that as the number of tasks increases, the cumulative average utilities in both *RL+DT* and *RL-DT* consistently converge with a 95% confidence interval. On the contrary, with significance $p = 0.093$, there is no statistical significance as to whether *NL+DT* converge or not. These results confirm our hypotheses.

VI. DISCUSSION AND RELATED WORK

The research presented in this paper represents the first model for combining argumentation, machine learning and decision theory to learn underlying social characteristics (e.g. policies/norms) of others and exploit the models learned to reduce communication overhead and improve strategic outcomes. There is, however, some prior research in combining machine learning and argumentation, and in using argument structures for machine learning. In that research, Možina et al. [22] propose an induction-based machine learning mechanism using argumentation. The work implemented an argument-based extension of CN2 rule learning (ABCN2) and showed that ABCN2 out-performed CN2 in most tasks. However, the framework developed in that research will struggle to learn and build an accurate model of policies from argumentation-derived evidence, which is the main issue we are addressing in our work. Also, the authors assume that the agent knows and has access to the arguments required to improve the prediction accuracy, but we argue that it is not always the case. As a result, we employ information-seeking dialogue to tease out evidence that could be used to improve performance by increasing utility and reducing communication overhead.

In related research, Rovatsos et al. [23] use hierarchical reinforcement learning in modifying symbolic constructs (*interaction frames*) that regulate agent *conversation patterns*, and argue that their approach could improve an agent's conversation strategy. In our work, we use dialogical structures such as requests for further information, requests for explanations, and so on to obtain evidence from the interaction and learn the entire sequence as against a segment (frame) of the interaction [23]. We have demonstrated the effectiveness of using argumentation-derived evidence to learn underlying social characteristics (e.g. policies) and combined it with decision theory to make effective decisions. Our results show that agents can improve their conversation strategy and make more informed decisions in resourcing their plans.

In recent research, Sycara et al. [12] investigate agent support for human teams in which software agents aid the decision making of team members during collaborative planning. One area of support that was identified as important in this context is guidance in making policy-compliant decisions. This prior research focuses on giving guidance to humans regarding their own policies. An important and open question, however, is how can agents support humans in developing models of others' policies and using these in decision making? Our work seeks to bridge this gap. We employ a novel combination of techniques in learning and building accurate models of others' policies, and exploit these in supporting decision making.

In our future work, we plan to develop further strategies for advising human decision makers on how a plan may be resourced and who to talk to on the basis of policy and resource availability constraints learned in critical and high-stakes situations. Furthermore, we plan to incorporate sunk cost and the cost of failing to resource a task into the framework. We are hoping that some of these ideas will provide helpful feedback with respect to future research on developing strategies for collaborative activities.

VII. Conclusions

In this paper, we have presented a technique that combines argumentation, machine learning and decision theory for supporting agents engaging in collaborative activities. We believe that this is the first study into combining the strengths of these three techniques into learning models of other agents and making more informed choices. The results of our empirical investigations show that combining argumentation, machine learning and decision theory can have a statistically significant positive impact on resourcing plans in particular, and decision making in general. The results also demonstrate that accurate policy models can help in developing more robust and adaptive strategies for advising human decision makers on how a plan may be resourced and who to talk to [12], and may aid in the development of more effective strategies for agents [14]. Our results demonstrate that significant improvements in utility and communication overhead can be achieved by combining argumentation, machine learning and decision theory. Having shown that accurate models of others' policies could be learned through argumentation-derived evidence and utilised in deciding who to talk to, we conjecture that one could, in principle, learn accurate models of other agents' properties (e.g. priorities) and make more informed decisions, in general.

Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] W. Shen, L. Wang, and Q. Hao, "Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans. on*, vol. 36, no. 4, pp. 563 –577, July 2006.

[2] E. Ephrati and J. S. Rosenschein, "Multi-agent planning as a dynamic search for social consensus," in *Proc. of the 13th Int. Joint Conf. on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 423–429.

[3] B. Grosz and S. Kraus, "Collaborative plans for group activities," in *Proc. of the 13th International Joint Conference on Artifical Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 367–373.

[4] M. Esteva, D. de la Cruz, and C. Sierra, "Islander: an electronic institutions editor," in *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2002, pp. 1045–1052.

[5] S. D. Ramchurn, B. Deitch, M. K. Thompson, D. C. de Roure, N. R. Jennings, and M. Luck, "Minimising intrusiveness in pervasive computing environments using multi-agent negotiation," in *Proc. of the 1st International Conference on Mobile and Ubiquitous Systems*, Boston, MA, 2004, pp. 364–372.

[6] B. Roberts, N. J. Pioch, and W. Ferguson, "Verbal coaching during a real-time task," in *Proc. of the 4th International Conference on Intelligent Tutoring Systems*. Springer, 1998, pp. 344–353.

[7] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, no. 7, pp. 30–40, 1994.

[8] T. R. Payne, T. L. Lenox, S. Hahn, K. Sycara, and M. Lewis, "Agent-based support for human/agent teams," in *International Conference on Human Computer Interaction (CHI) Demonstration*. ACM Press, 2000.

[9] J. Allen, A. Mowshowitz, T. Norman, S. Parsons, and A. Preece, "Managing collaboration in hybrid-agent teams," in *Proc. of the 1st Annual Conference of the International Technology Alliance (ACITA 2007)*, University of Maryland, 2007.

[10] H. Chalupsky, Y. Gil, C. A. Knoblock, K. Lerman, J. Oh, D. V. Pynadath, T. A. Russ, and M. Tambe, "Electric elves: Agent technology for supporting human organizations," *Artificial Intelligence*, vol. 23, no. 2, pp. 11–24, 2002.

[11] W. Chen, V. Manikonda, and E. Durfee, "A flexible human agent collaboration (hac) framework for human-human activity coordination (h2ac)," in *Proc. of the AAAI Fall Symposium on Regarding the "Intelligence" in Distributed Intelligent Systems*, Virginia, USA, 2007, pp. 38–41.

[12] K. Sycara, T. J. Norman, J. A. Giampapa, M. J. Kollingbaum, C. Burnett, D. Masato, M. McCallum, and M. H. Strub, "Agent support for policy-driven collaborative mission planning," *The Computer Journal*, p. bxp061, 2009. [Online]. Available: http://comjnl.oxfordjournals.org/cgi/content/abstract/bxp061v1

[13] C. D. Emele, T. J. Norman, F. Guerin, and S. Parsons, "Learning policy constraints through dialogue," in *Proc. of the AAAI Fall Symposium on The Uses of Computational Argumentation*, USA, 2009, pp. 20–26.

[14] N. Oren, T. J. Norman, and A. Preece, "Loose lips sink ships: A heuristic for argumentation," in *Proc. of the 3rd Int'l Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, 2006, pp. 121–134.

[15] A. Bahrami, J. Yuan, C. D. Emele, D. Masato, T. J. Norman, and D. Mott, "Collaborative and context-aware planning," in *Proc. of the Military Communications Conference (MILCOM 2008)*, San Diego, CA, Nov. 2008, pp. 1–7.

[16] S. Parsons, M. Wooldridge, and L. Amgoud, "An analysis of formal inter-agent dialogues," in *Proc. of the 1st International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 2002.

[17] D. N. Walton and E. C. W. Krabbe, *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. USA: SUNY Press, 1995.

[18] J. Pearl, *Causality: Modeling, Reasoning, and Inference*. Cambridge: Cambridge University Press, 2000.

[19] C. D. Emele, T. J. Norman, F. Guerin, and S. Parsons, "On the benefit of argumentation-derived evidence in learning policies," in *Proc. of the 3rd Intl. Workshop on Argumentation in Multi-Agent Systems*, Toronto, Canada, 2010.

[20] T. M. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[21] S. Parsons and N. R. Jennings, "Negotiation through argumentation-A preliminary report," in *Proc. of the 2nd Int'l Conference Multi-Agent Systems (ICMAS'96)*, Kyoto, Japan, 1996, pp. 267–274. [Online]. Available: citeseer.ist.psu.edu/parsons96negotiation.html

[22] M. Možina, J. Žabkar, and I. Bratko, "Argument based machine learning," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 922–937, 2007.

[23] M. Rovatsos, I. Rahwan, F. Fischer, and G. Weiss, "Practical strategic reasoning and adaptation in rational argument-based negotiation," in *Argumentation in Multi-Agent Systems*, ser. LNCS, vol. 4049. Springer-Berlin, 2005, pp. 122–137.