Cooperation and Group Utility

Guido Boella, Rossana Damiano and Leonardo Lesmo

Dipartimento di Informatica and
Centro di Scienza Cognitiva - Università di Torino
C.so Svizzera 185 10149 Torino ITALY
email: {guido,rossana,lesmo}@di.unito.it

Abstract. In this paper, we propose an definition of cooperation to shared plans that takes into account the benefit of the whole group, where the group's benefit is computed by considering also the consequences of an agent's choice in terms of the actions that the other members of the group will do. In addition, the members of a group consider whether to adopt the goals of their partners: an agent should adopt these goals only when the adoption results in an increase of the group's benefit.

1 Introduction

Jennings and Campos [16] claim that socially responsible agents, when they decide what actions to perform, should not consider just the individual benefit they may achieve or the society benefit alone, but a combination of both. This hybrid measure of benefit, called *joint benefit*, allows to improve the performance of the agents and of the overall community of agents as well. [16] evaluates the advantage of performing an action without including explicitly the consequences that the action may have on the choices that the other agents will make afterwards. But in some cases these consequences need to be taken into account. For example, as [11] highlighted, an agent belonging to a group should decide to perform an action for the sake of a partner only if this decision reduces the overall cost. But in order to evaluate that cost, he must consider how his action affects the subsequent actions of his partner.

But the notion of society benefit can be seen from different perspectives. In a general model of agent behavior (as the one described in [8]), the social behavior is reduced to the existence of social norms that are usually obeyed by an agent. The norms interact with the individual benefit according to a flexible schema, where the personality of the agent suggests him to give more or less weight to the norms.

Although this approach seems reasonable, it draws a picture where just individuals and society are relevant. On the contrary, at least three levels seem required: the individual level, the group level, and the society level. In fact, not all interactions among agents are governed by social rules. In some cases, a group can be formed to achieve some specific goal. If this is the case, the internal dynamics of the group are mainly ruled by the common decision to cooperate towards the achievement of the shared (or common, or joint) goal. In this context, social rules do not disappear, but they interact with the need of an agent to take into account how the other agents in the group may behave to achieve the common goal. In other words, the existence of a common goal

N.R. Jennings and Y. Lespérance (Eds.): Intelligent Agents VI, LNAI 1757, pp. 319-333, 2000.
© Springer-Verlag Berlin Heidelberg 2000

can force an agent to execute (and another agent to accept) an action which goes against the social norms.

It is possible that the distinction between the group level and the social level is not so sharp. For instance, [13] observes that when the number of agents involved in a group grows above a certain threshold (the authors refer to groups of hundreds or thousands of agents), computational constraints prevent an agent from taking into account the effects of his actions on all agents in the group. It could be speculated that it is this difficulty that gave rise to societies: since an agent cannot, by himself, determine by means of rational evaluation the effects of his actions on all the people that surround him, then social norms are established, in order to favour the overall benefit (which, unfortunately, is not the case for all social norms).

In this paper, we address the problems that arise in the internal coordination of (small) groups of agents. We claim that in such a context, when an agent chooses a line of action, it does so by trying to figure out how his individual action can affect the advancement of the whole group towards the common goal. But the evaluation of the impact of an action on the group activity is not an easy task. The choice of action A_i by agent G_i can be very profitable in case another agent in the group (G_j) chose the action A_j for carrying out his task, but the same action A_i can have destructive consequences in case G_j chose A'_j. This happens, for instance, when some resources are shared among the group's members. If A_i decides to use the only big truck available to the group to carry around something, just because the big truck is easier to load than a smaller truck, then the goal of the group is not achieved in case another agent needs that truck to accomplish his duty.

So, the approach described in this paper is a three-step approach: first, figure out the state resulting from your action; second, imagine how the partners in the group can start from this state to plan their part; third, figure out the states resulting from the actions executed by the partners and choose the action according to the final resulting state. It is clear that this approach is based on various assumptions, some of which are safe and some of which are just simplifications of problems that require deeper analysis:

1. any state can be evaluated in order to decide how close it is to the achievement of the goal of the group. This is a safe assumption: the evaluation can be more or less accurate (a kind of heuristic function), but some static evaluation function is always required, unless it is possible to build a complete group plan that achieves the goal.
2. an agent can determine which states result from his actions. Again, this seems to be safe, provided that some probabilistic setting describes the possible outcomes. Again, the probability values associated with the outcomes can be approximate, but they appear to be the most reasonable way to reason about action effects (see, for instance, [21]).
3. an agent can foresee the choices of other agents. In principle, the model should include a set of nested beliefs (i.e. beliefs about what the other agents believe they can do, about how they evaluate the cost of their actions, about how they evaluate the states resulting from their actions, etc.). With respect to this problem, we assume identity between all agents, that is an agent believes that the other agents in the group know exactly the action he knows and use the same functions to evaluate cost and benefits. This is clearly a simplifying assumption.

4. an agent can predict the choices of other agents (again). This is the same as above, but here we point out a different underlying difficulty. Agent A_i decides what to do on the basis of what he foresees will do A_j (let's consider just two agents). But the choice of A_j does not depend (in our model) just on the desirability of the state resulting from his actions, but also on the consequences that his choice has on the subsequent behavior of A_i. So, we have a one-level lookahed of A_i that cannot take into account the one-level lookahed that A_j will apply next (otherwise, we would have an infinite lookahead). So the model implies that the foreseeing of A_j moves is inaccurate. A possible solution could be to let the degree of lookahed vary according to the importance of the consequences.

In this paper, we propose an operational definition of cooperation to shared plans that takes into account the benefit of the whole group, where the group's benefit is computed by considering also the consequences of an agent's choice in terms of the actions that the other members of the group will do. In addition, the members of a group consider whether to adopt the goals of their partners: an agent should adopt these goals only when the adoption results in an increase of the group's benefit.

In order to define the notion of benefit, our proposal exploits the decision theoretic planning approach of [12]: the utility of an outcome is computed as a weighted sum of the utility functions associated to the single goals of the agents and to resource consumption. By using these multiattribute utility functions, we can take into account both the individual utility of an agent and the utility of performing a plan for achieving the group's goal, given the related costs for the group - in terms of resource consumption.

2 The Deliberative Agent Architecture

In [1], we describe the agent architecture of the system that underlies this work. In this system, the knowledge about how to act is stored in three libraries of actions. For each action, its preconditions, its constraints and its effects are specified. The libraries are organized as abstraction/decomposition hierarchies. The abstraction hierarchy represents alternative ways of executing an action; the decomposition of actions is given in terms of recipes: the execution of the subactions (steps) in a recipe constitutes the execution of the action.

Here, we assume that an agent has a set of goals and that he does planning in order to find a plan which satisfies one or more of these goals and maximizes the agent's utility. The chosen plan constitutes the current intention of the agent. Then, the (possibly partial) plan is executed in a reactive manner, i.e. monitoring effects and triggering replanning in case of failure.

Once an agent has selected a set of intentions, the process of plan formation can be re-triggered by other external events; in fact, external events can lead to the formation of new goals, which must be balanced against the existing intentions. The agent, exploiting his utility function, considers if it is possible to modify the previous plan to include the new goal - so that it becomes an intention beside the existing ones - or if it is better to continue with his previous plan - the goal does not become an intention; in summary, he has to consider both the utility of adopting the possible goal as a new intention (and doing something to achieve it) and the utility that results from the consequences of not adopting it.

As stated in the introduction, the strategy for action selection described in the paper applies to an agent acting within a group. A first basic problem we do not address in the paper concerns the way the group is formed: we assume that a group is already at work, with different tasks assigned to the various members of the group.

We also assume that the shared plan is partial [11] in the sense that the assigned actions need not be (and usually are not) expanded to a great level of detail. In this stage, all involved agents are committed to execute the actions which they are in charge of. However, the main point of the paper is that the agents are not alone in carrying out their tasks, but that in some cases they may interact either to ask for help or to volunteer some help.

The request for help is rather easy to model, since all plans may fail, and all agents are assumed to have some means of communication available that enable them to interact (at some cost) with other agents in the group (and, possibly, also with agents outside the group). So, if an agent detects some obstacle in his autonomous activity, he can start up a planning (or re-planning) activity, that could result in a new multi-agent plan, in the sense that it is a plan involving the help of another agent. The process is standard, provided that the agent is able to take into account the decreased group utility resulting from the need to distract another agent in the group from his current task.

More complex is the case of volunteering help. In principle, this would require every agent continuously monitoring the other agents in the group, with an indefinite waste of resources devoted to checking if any partner is doing something that could be done more efficiently (greater utility) if he adopted the partner's goal.[1]
A possible approach to this kind of help involves two kinds of mechanisms. First, the model requires that every agent has at his disposal a complete picture of the original goals of the partners (and this, in fact, is what happens when the initial shared plan is formed). So, for instance, if an agent A has been assigned the task of providing some fruit for a dinner, in case another agent B builds a plan involving going to the supermarket to buy some pasta, B could realize that a pending goal of A could be solved more cheaply by adopting it ("If you like, I can buy some fruit at the supermarket"). The model we present in the paper is able to cope both with requested help and with this first kind of volunteered help (a special case of which are the notifications discussed in section 5).

The second kind of mechanism is outside the reach of the current implementation, since it involves a filtered monitoring of sensed events. Some heuristic filter should be available to agents in order to enable them to detect potentially relevant environmental events, which could lead to replanning. In particular, such an event could produce the need to replan the whole shared plan, but more frequently they just require a local modification in the plan to include the agent's intervention in the activity of a single partner. Although we have not worked on this, from a general point of view it can be observed that the reaction to requests could be viewed as a special case of it, where the incoming request goes through the filter and is considered for a possible reaction (which must be planned) by the receiver.

---

[1] We must thank an anonymous referee for this observation.

Turning back to the details of our approach, it is clear that, since in many situations a decision must be taken about which plan to choose, we need some techniques to balance the different possibilities. [18] is a first attempt to integrate agent theories and decision theory. A more recent solution is the one in [12], where it is described a way to relate the notions of goals and planning to that of utility. So, in order to develop our approach, we adopted the decision theoretic planner DRIPS described in [12], and we built our architecture on the top of it.

DRIPS is a hierarchical planner which merges some ideas of decision theory with standard planning techniques. A utility function is used for evaluating how promising a given plan is. The utility function does not compute just the payoff of the possible outcomes of the plan: it is computed starting from simpler utility functions associated with the goals of the agent [2], so that utility depends directly on goal satisfaction [6]. The important properties of DRIPS are that it allows to model different degrees of achievement of a goal (satisfying a goal to a greater extent is considered preferable to satisfying it to a lesser extent) and that the success in satisfying one goal component is traded off against the success in satisfying another goal, or against consuming resources.[3]

The choice to adopt an approach based on utility functions has been challenged by Jennings and Wooldridge's claim that the existing models based on the notion of utility "… are not computational models and ignore the practicalities of computing an appropriate action to perform" [14], p.5. Although DRIPS is based on decision theory, it is in fact a computational model based on the notion of utility, which exploits various techniques to account for probability and uncertainty: these are important factors that, as [10] has noticed, tend to be underestimated in logic-based formalizations. It can be also noted that the use of utility for choosing a line of behavior is gaining favour also in applicative settings (see, for instance [20]).

However, DRIPS deals only with single-agents plans and individual utility. So its basic planning mechanism was extended to build the planning architecture which satisfies our definition of cooperation to a shared plan.

3 The Definition of Cooperation

We introduce the notion of a utility function, shared by the group and associated to the goal that the group wants to achieve by means of a shared plan. The group utility function is just one of the components of the global utility functions of each single agent: therefore, if the utility of achieving a personal goal overcomes the utility of performing the shared plan, the agent can give up the cooperation.

The group utility function has a fundamental role in case of helpful behavior: we claim that any member, beside doing his part in the shared plan, will consider whether to adopt the goals of his partners, but will adopt them only if this results in an increase

---

[2] This aggregation of simpler utility functions in a global one is possible only if some independence assumptions hold (see [12]).

[3] Notice that the utility is associated in DRIPS with a resulting state, but in such a state the consumption of resources in executing the plan is accounted for, so that what one actually gets is the utility of that plan and not only the utility of the associated goal.

of the overall utility. The notion of goal adoption has been introduced by [7]: "an agent B performs an action for a given goal, since this action or the goal are included in the plan of another agent A".

A group of agents GR composed of agents $G_1 \ldots G_n$ cooperates to a shared plan for α[4] with an associated recipe R^x composed of steps $\beta_1^{x,i_1} \ldots \beta_m^{x,i_m}$ when:[5]

1. each step $\beta_r^{x,i}$ has been assigned to an agent G_i in GR for its execution
2. each agent G_i of the group GR has the single agent intention to perform his part $\beta_r^{x,i}$ of the shared plan for α formed on the basis of the recipe R^x;
3. the agents of GR have the mutual belief that each one (G_i) has the intention to perform his part $\beta_r^{x,i}$ of the shared plan for α;
4. all agents mutually know that they share a utility function GF based on a weighted sum of the utility of the goal which the shared plan aims at and of the resource consumption of the single agents; each agent, when he plans his own part of the shared plan, has to consider also this global utility as part of the his individual utility function F_i;
5. when an agent G_i becomes aware that a partner G_j has a goal ϕ that stems from his intention to do his part $\beta_p^{x,j}$, G_i will consider whether to adopt it; if G_i believes that the adoption of ϕ produces an increase of the utility GF of the whole group, then he adopts that goal;
6. each agent remains in the group as long as the value of the utility function GF can be increased by executing his part of the shared plan for α or by adopting some of the goals of the partners.

For what concerns point 5, the goals that can be potentially adopted by an agent G_i are all the goals stemming from the intention of a partner G_j to perform an action $\beta_p^{x,j}$: therefore, he considers not only the steps he may execute to assist G_j in performing $\beta_p^{x,j}$, but also the need to make its preconditions true and other goals possibly deriving from the reactive execution of $\beta_p^{x,j}$. In particular, G_i will consider G_j's goals deriving from the single agent intention of G_j to perform $\beta_p^{x,j}$: knowing how to perform $\beta_p^{x,j}$, monitoring the effects of its execution and, possibly, replanning. If G_i knows what G_j is currently doing, he can infer what move G_j will subsequently consider.

So, plan recognition techniques (e.g. [5], [2], [1]) can play an important role in helping agents to infer what their partners are doing and therefore improving the cooperation of the group.

It must also be observed that at point 5 the term "aware" is used. This is related to the comments made in the Section 2 about requesting help and volunteering help. Currently, we assume that an agent A can become aware of another agent B's need for help either because B communicated to A something which can be interpreted as a direct or an indirect request (see [3]), or when A realizes that there is some action he can do to achieve a sub-goal of B in a cheaper way. Apart from the mutual knowledge

---

[4] Actually, we assume here that the initial shared plan has a particular structure; i.e. it is a one-level plan composed of a top-level action (α) decomposed into a sequence of steps. In a general plan, each step could in turn have been expanded into substeps, and so on recursively.

[5] The notation $\beta_l^{x,i}$ refers to the l-th step of the recipe R^x, a step which has to be executed by agent G_i.

about the top-most goal appearing in the initial shared-plan, the only way an agent A can get knowledge about the plan another agent B is carrying on is B's communication to A (which may activate the plan recognition process).

4 The Planning Algorithm

Since the agent's world is populated by other agents, the consequences of an action may affect the subsequent behavior of other agents. So, in case of interaction, an agent has to consider the consequences of his behavior with respect to what the other agents will do afterwards: in order to evaluate the real expected utility of the plan, he must explore the outcomes that may result from the expected reactions of other agents.

The construction of a plan is carried out by an agent G_i in a stepwise fashion: if G_i is in charge of step $\beta_m^{x,i}$ of the recipe R^x shared with G_j[6] for achieving goal α, then he first has to find the best recipe for $\beta_m^{x,i}$ (let's say R^y, with steps $\gamma_{m,1}^{y,i}, \gamma_{m,2}^{y,i}, \ldots, \gamma_{m,n_m}^{y,i}$), and then he can start refining $\gamma_{m,1}^{y,i}$. The approach of DRIPS to this process is to expand $\beta_m^{x,i}$ in all possible ways (i.e. applying to the current state S all existing recipes); then, it proceeds onward and expands the new partial plans. The search goes on in parallel, but the search tree is pruned using the utility function (applied to the state resulting from the potential execution of the recipe): so, the utility function acts as a heuristic able to exclude some possible ways (recipes) to execute an action.

In order to implement the ideas presented in the previous section, we had to make the evaluation of the heuristics somewhat more complex. The method is as follows:

- Using DRIPS (playing the role of G_i), we expand the current state S according to all alternative recipes for $\beta_m^{x,i}$, thus producing the states $S_1, S_2, \ldots, S_r$ (where r is the number of different recipes for $\beta_m^{x,i}$).
- This set of states is transformed in the set of the same states as viewed by G_j, $S'_1, S'_2, \ldots, S'_r$.[7]
- On each state S'_m $(1 \leq m \leq r)$, we restart the planning process from the perspective of his partner G_j (i.e. trying to solve his current task $\beta_h^{x,j}$).
- This produces a set of sets of states $SS' = \{\{S'_{1,1}, \ldots, S'_{1,n_1}\}, \{S'_{2,1}, \ldots, S'_{2,n_2}\}, \ldots, \{S'_{n_r,1}, \ldots, S'_{r,n_r}\}\}$.
- The group utility function is applied to these states, and the best state of each subset is identified: $SS'_{best}=\{S'_{1,best(1)}, S'_{2,best(2)}, \ldots, S'_{n_r,best(n_r)}\}$. These states are the ones assumed to be reached by G_j's best action, for each of the possible initial moves of G_i.
- The group utility function is applied to the states $S_{k,best(k)}$ $(1 \leq k \leq r)$ from G_i's point of view. This models the perspective of G_i on what could happen next.

---

[6] For simplicity we have assumed a single partner G_j.

[7] The problem of simulating another agent's planning is very difficult. For instance, in some situations, G_j could not be aware of R^x effects. In our implementation, we adopted the simplification that G_j's knowledge of a state is updated by an action of G_i only with the effects which are explicitly mentioned as believed by G_j (e.g. the result of a communicative action having G_j as receiver).

- The best one of these states is selected ($S_{max,best(max)}$). This corresponds to the selection of the best recipe for $\beta_m^{x,i}$ of G_i (i.e. R^{max}).

Note that the algorithm above is just a modification of a two-level min-max algorithm: actually, it is a max-max, since at both levels the best option is selected, although at the second level it is evaluated from G_j's perspective. As in min-max, G_i, when predicting G_j behavior, assumes that his partner is a rational agent, i.e. that he will choose the plan that gets the highest utility for the group.

The mechanism described above for the choice of the best recipe is computationally expensive. However, a first step toward a more efficient solution is achieved by exploiting the DRIPS mechanism of pruning the search tree when a partial plan looks unpromising compared to the other hypotheses: this algorithm is applied both when the agent's plans and his partner's ones are devised.

Moreover, since our system admits partial plans, we decided to let DRIPS stop after it has reached a certain level of detail without expanding the plan completely. This is a rather standard method in reactive planning since, as [4] noticed, agents limit the search for solutions to partial ones, because working in a dynamic world makes overdetailed plans often useless.

5 Predictions of the Model

In this section we consider the predictions deriving from our definition of cooperation.
Helpful behavior and communication: in our model, helpful behavior (i.e. adoption) is at the basis of cooperation: the agent considers the goals of other agents and, only if it is useful for the group, he adopts them.

Consider the situation where two agents, G_i and G_j, are preparing a meal together. One of them, say G_i, is following a recipe that requires beaten eggs, but his partner, G_j, is using the food-processor. After G_i communicates to G_j that he needs the food-processor to beat the eggs, for example by means of a communicative act ("Can I use the food-processor to beat the eggs?"), two alternatives are open to G_j: he can give the food-processor to G_i or, since he later needs beaten eggs too, he can volunteer to prepare two portions of beaten eggs. If the latter option results less expensive for the group than G_i doing the action by himself (it does not require that the food processor be unplugged, moved and washed twice), G_j will choose this option, thus adopting G_i's goal. Note that, as stated in section 3, the planning process which considers the possibility of adoption is triggered by an explicit request of G_i, which enables G_j to infer a (sub)goal of G_i, i.e. having some beaten eggs.

Alternatively, suppose that G_j cannot give G_i the food-processor and cannot use it to beat eggs for G_i without interference with the preparation of his own recipe. If this is the case, G_j does not adopt G_i's goal, since this would result in a lower group utility (so that G_i must resort to beating the eggs by hands or to changing recipe).

As a special case of helpful behavior, the model predicts that an agent, whenever he comes to know that the shared goal has been achieved, is impossible to achieve or has become irrelevant, will notify this fact to his partners (see [9]). In fact, as stated above, if an agent G_i knows that a partner G_j has a goal H_j, G_i can infer that G_j will also have the subsidiary goal of having information about the status of H_j. Assume that, suddenly, G_i comes to know (without any further cost) that H_j holds. In his next planning phase he has to reconsider whether to adopt G_j's goal of knowing if H_j holds; if G_i adopts this goal, he has just to communicate to G_j that H_j holds, a low-cost action that does not add much overhead to G_i and therefore to the group's utility. On the contrary, the alternative of going on with his activity involves a greater waste of group resources (G_j will continue his - now useless - work).[8]

Of course, G_i has to consider the cost of communicating with G_j. If the cost of the communication outweights the saving of resources (due to the reduced cost of the work of A_j), then no communication is started up. The same can happen if the communication is not effective (the message has high probability of getting lost or of arriving too late).

In these cases, even if an agent decides that it is better (for the group) not to communicate, his choice does not disrupt the group: in fact, communication is not explicitly prescribed in our definition of cooperation. For the same reasons, adopting another agent's intention doesn't necessarily involve communicating the adoption to the beneficiary; again, communication is subject to utility evaluation, that includes, in turn, evaluating the risk of two agents doing the same action twice.

Hierarchical groups: the consumption of resources need not be weighted in a uniform way for all members of the group; we can induce a sort of hierarchy in the group by allowing actions that are weighted differently depending on who executes them. For example, if G_i's communication is more costly than G_j's one and G_i first succeeds in knowing that the goal holds, he may not communicate this fact to G_j, while G_j would notify G_i when he becomes aware of that.

Conflict avoidance: since agents share a group utility function, we can predict that they will (try to) avoid conflicts with other agents' intentions: performing an action that interferes with the plans of other team members decreases the utility of the whole team.

Considering the differential in utility among the various alternatives of an agent G_i doesn't not rule out every possible conflict; when G_i considers the possible developments of his partial plan, he examines what effects his action will have on the partners' plans. So also the possible interferences are weighted as any other cost that decreases the group utility.

For example, if two partners who are preparing a meal together have only one pan and one of them needs it urgently, he can decide to use it - even if he knows that his partner will need it later - because this is more convenient, for the group, since the pan can be easily washed later. On the contrary, if the shared resource is a not reusable one, like for example eggs, then G_i will use it only if he cannot change his plan without a significant decrease of utility, while the same does not hold for the partner (for example, he is in the middle of the preparation of his recipe, while G_j has not begun yet).

Contracting out: an agent G_k can delegate a step $\beta_l^{x,k}$ of his own part to another agent A that does not belong to the group GR; G_k and A will form a new group sharing the goal to perform $\beta_l^{x,k}$ and a corresponding utility function. In this situation, A does not

---

[8] The same happens for goals become impossible or irrelevant.

become a member of the group: in fact, he will not necessarily know the group's goal and utility function. So, it is easier that he interferes with GR while performing $\beta_l^{x,k}$.
Ending cooperation: when all members know that the top-level goal of the group has been achieved, has become impossible or irrelevant, no more utility can be obtained by any other actions than terminating: in fact, termination gets higher utility by saving resources. Therefore, the shared plan is naturally ruled out, without the need of stipulating conditions for its termination.

Increasing the utility of the group sometimes produces an unacceptable decrease of the private utility. But the basis on which an agent decides his behavior is his own utility function, of which the global one is just a component. In this case he will opt out from the shared activity: as [11] notice, this is not an harmless choice since the other agents can retaliate for being abandoned (for example by not helping him in future situations). When the cost of remaining in the group is greater than the consequences of leaving it, the agent will choose to pursue his private goals.

6 A Simple Example

The set of actions reported at the end of this section represents the situation where agents A and B are looking for an object: therefore, the shared plan is composed of the single-agent actions of searching in separate places and when an agent finds the object, the other will not be aware of it. The shared utility function is U(S) = k_1UG(S) + k_2UR(S), where UG(S) is 1 if, in outcome S, the object is found before a given deadline and 0 otherwise; UR(S) is the utility of saving resources and is equal to (fuelA - fuelB)/maxfuel. Until the object has been found, the expected utility of the action of searching is different from zero, while after the discovery, the only utility is provided by saving resources.
Consider the situation where A finds the object (BelAfound = true): the planning process is retriggered by this event. A now has three alternatives: a) going on looking for the object (that amounts to wasting resource without further utility); b) communicating to B that he has found the object (BelBfound = true), as a consequence of the adoption of B's goal of knowing whether the group succeeded in its goal; c) doing nothing more for the group, since the goal has been obtained (or, in general, doing an action for achieving some other goal).

If A chooses alternative (a) or (c), he knows that B will go on searching the object, since BelBfound = false. Only if A communicates with B, the utility of the group results in an increase: in fact, if the action succeeds, BelBfound = true and, therefore, B will stop searching. In this situation, the group will gather the highest utility, since the action of communicating is less costly than B's action of continuing to search.
However, after the deadline has expired, B will stop searching, whatever A does: in this case, A will choose alternative (c) since communication is useless.

```
/* In this example, we have slightly simplified DRIPS' syntax in order to enhance readability.
   E.g. the actual appearance of 'donothingA' below would be:
   (add-action donothingA (cond (t (1 ())))).
   In our syntax, the lists following the action name specify the effects of the action.
   The numbers following the lists are the probabilities that the action achieves the effect;
   for instance, the 'communicateA' action succeeds (BelBfound := true) only with a probability
```

```
   of 0.9 (see Appendix for more details about probabilities in DRIPS).
   The effects can be conditioned to contextual situations (see the 'cond' in 'searchA').*/

/* action of doing nothing more for the shared plan: no resources are consumed by A (fuelA) */
(add-action donothingA ())

/* action of communicating to B that the object they are looking for has been found
   (BelBfound := true means that B knows that). */
(add-action communicateA (time := time + 5) 1
                         (fuelA := fuelA - 2) 1
                         (BelBfound := true) 0.9)

/* action of looking for the object; possibly (0.4) A will eventually find it:
   even if this happens, B will not be aware of this fact */
(add-action searchA (cond ((BelAfound = false)
                                        (time := time + 30) 1
                                        (fuelA := fuelA - 3) 1
                                        (BelAfound := true) 0.4)
                         ((BelAfound = true)
                                        (time := time + 30) 1
                                        (fuelA := fuelA - 3) 1)))

/* the action that is given in input by A to the planning procedure: its main function
   is to specify the more specific actions that can be executed to accomplish the plan,
   i.e. searching, communicating or leaving the group.
   Its effects subsume those of the more specific actions. */
(add-action planA (time := time + (0 5)) 1
                         (fuelA := fuelA - (0 2)) 1
                         (BelBfound := (false true)) 1
                         (more-specific /* list of more specific actions */
                            communicateA searchA donothingA))
```

7 Comparison with Related Work

The problem of modeling group cooperation for achieving a common goal has been widely discussed in the last decade. In [9] the authors propose a formalization of joint intention which has been the basis for much research. The main contribution of that work is the association of the notion of persistent goal with the need for the agents to coordinate their activity during the execution of a joint plan. In particular, their formalization highlights the need of making mutually believed among the members of the group the status of the conditions for the termination (or continuation) of the execution (i.e. that a goal has been achieved, or that it is unachievable, or that it has become irrelevant). Apart from this simplification, we need not associate to the "communicate" action any special status: it is just one of the actions an agent has at disposal. And, as any other action, it undergoes an evaluation of the advantage of choosing it as the next action. In particular, its execution is subordinated to the utility of achieving a given mutual belief (in our simplified version), a utility evaluated in terms of its gain and its cost. As it has been shown, when the termination conditions are met, the evaluation of the utility can produce, depending on the context, different behaviors: usually, but not necessarily, it produces the communication of this information to the partner(s). [19] uses in a similar way utility theory for deciding whether to communicate to the partners that the group has to be disbanded.

Jennings [15] distinguishes between commitments and conventions in order to keep apart the notion of commitment from the means for monitoring them. In this way, Jennings is able to tune the coordination of the team by means of communication depending on the circumstances (available communication bandwidth, hierarchical relations in the group). In our model, coordination is not explicitly prescribed by the definition of cooperation but it is derived by the fact that agents consider what their partner will do as a consequence of their behavior. Different coordination strategies will arise, since the decision to communicate is sensitive to the context: in fact, it takes into account the characteristics of the communication means (cost, times, reliability) as well as the group's utility produced by communication.

Grosz and Kraus [11] propose a formal specification of the notion of sharing a plan. They introduce the operator Intend-that in order to account for the commitment of each group member to the shared plan; from the intention-that the plan be performed, the agents derive that they have to avoid conflicts and to coordinate the group's behavior through communication. In addition, the definition of shared plans prescribes that agents intend that their partners are able to do their part in the plan. In our model, we have tried to obtain a similar effect by means of the interaction of the shared utility function with the mechanism of goal adoption. In particular, conflicts are avoided since a plan that interferes with the partners' actions normally makes the utility of the group decrease. The goal adoption mechanism makes an agent consider whether, by adopting the partners' goals, a gain for the group is achieved.

Jennings and Campos [16] have proposed a method for modeling socially responsible agents on the basis of a combination of member benefit/loss and social benefit/loss. They have shown that taking into account social benefit/loss makes the group work better towards the achievement of the common goal(s). We believe that their proposal is compatible with the work described herein: we have focused on the way different possible courses of actions can be identified and examined and on how the choice of the action is affected by this search process; on the contrary, the way utilities are evaluated locally has received less attention. We believe that Jennings and Campos' contribution is fundamental in this respect.

In [14] Jennings and Wooldridge introduce in their definition of cooperation the characterization of the preliminary phases of forming a group. On the contrary, in this work we start from the situation where the group has already formed and a recipe for achieving the shared goal has been selected. However, as [14] notice, an agent chooses a cooperative solution when he believes that this allows to achieve the goal more accurately or more quickly than acting in isolation.

DAI has devoted much attention to distributed planning. For example Lesser [17] has introduced a graph formalism to model the interdependences among the actions of agents performing a distributed goal search. Our work has a narrower scope: we assume that a recipe for executing a shared plan has already been given and the planning mechanism is exploited by an agent to find the way for executing his part in the plan which maximizes the utility of the group. Moreover, the goal adoption mechanism accounts for the new goals that arise during the execution of the plan.

Finally, with respect to the probabilistic schema, [21] proposes a framework for dealing with uncertainty in action outcomes. Although our work is not focused on uncertainty (we have used almost entirely the method embodied in DRIPS, see the appendix), we can note that the methods for choosing the the initial plan are rather different. In [21], there are explicit *enablement* links between the action chosen by the agent and the one chosen by his partner, while in our approach the reasoning is based on the states resulting from the actions. It seems that the method in [21] is more knowledge-intensive than ours, but it enables the authors to face more efficiently the problems associated with failures in action execution.

8 Conclusions

[10] has highlighted the role that the economical theories of rationality, as decision theory is, can have in AI, notwithstanding the many restrictive assumptions that it is necessary to make in order to exploit this kind of theories.

In this paper, we show how a decision theoretic approach to planning can be exploited in understanding coordination in a team of agents. In doing this, we tried to overcome the limitation of rational decision noticed by Jennings and Campos [16]: the methodological solipsism that leads agents to maximize their individual utility without considering the society in which they act.

In particular, taking into consideration benefits and costs for both an agent and his partners, as suggested in [16], and exploiting the expectations about what he is going to do, it is possible to predict conflict avoidance among agents and to tailor coordination to the context.

Acknowledgements

This work has been partially supported by the project “Conoscenza, intenzioni e comunicazione” (Knowledge, intentions, and communication) of the National Research Council and by the project “Approcci modulari all’analisi linguistica” (Modular approaches to linguistic analysis) of the Ministry for University and Scientific Research.

References

1. L. Ardissono and G. Boella. An agent model for NL dialog interfaces. In *Lecture Notes in Artificial Intelligence n. 1480: Artificial Intelligence: Methodology, Systems and Applications*, pages 14–27. Springer Verlag, Berlin, 1998.
2. L. Ardissono, G. Boella, and L. Lesmo. An agent architecture for NL dialog modeling. In *Proc. Second Workshop on Human-Computer Conversation*, Bellagio, Italy, 1998.
3. L. Ardissono, G. Boella, and L. Lesmo. Plan based agent architecture for interpreting natural language dial ogue. *to appear in International Journal of Human-Computer Studies*, 2000.

4. M.E. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.
5. S. Carberry. *Plan Recognition in Natural Language Dialogue.* MIT Press, 1990.
6. C. Castelfranchi and R. Conte. Limits of strategic rationality for agents and m-a systems. In *Proc. of 4th ModelAge workshop on formal models of agents*, Pontignano, 1997.
7. C. Castelfranchi and R. Falcone. From task delegation to role delegation. In M. Lenzerini, editor, *LNAI 1321. AI*IA 97: Advances in Artificial Intelligence*, pages 278–289. Springer Verlag, Berlin, 1997.
8. Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur. Deliberate normative agents: Principles and architecture. In N.R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI — Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000. In this volume.
9. P.R. Cohen and H.J. Levesque. Confirmation and joint action. In *Proc. 12th IJCAI*, pages 951–957, Sydney, 1991.
10. J. Doyle. Rationality and its roles in reasoning. *Computational Intelligence*, 8:376–409, 1992.
11. B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.
12. P. Haddawy and S. Hanks. Utility models for goal-directed, decision-theoretic planners. *Computational Intelligence*, 14:392–429, 1998.
13. Lisa Hogg and Nick Jennings. Variable sociability in agent-based decision making. In N.R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI — Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000. In this volume.
14. N. R. Jennings and M. J. Wooldridge. Cooperative problem solving. *Journal of Logic and Computation*, 1999.
15. N.R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8 (3):223–250, 1993.
16. N.R. Jennings and J.R. Campos. Towards a social level characterisation of socially responsible agents. In *Proc. IEE Proceedings on Software Engineering*, volume 144 (1), pages 11–25, 1997.
17. V.R. Lesser. A retrospective view of FA/C distributed problem solving. *IEEE Trans. on Systems Man and Cybernetics*, 21:1347–1363, 1991.
18. A. Rao and M. Georgeff. Deliberation and intentions. In *Proc. of 7th Conference on Uncertainity in Artificial Intelligence*, Los Angeles, 1991.
19. M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.
20. Thomas Wagner and Victor Lesser. Relating quantified motivations for organizationally situated agents. In N.R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI — Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000. In this volume.
21. Ping Xuan and Victor R. Lesser. Incorporating uncertainty in agent commitments. In N.R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI — Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000. In this volume.

9 Appendix

Some more words must be devoted to the probability that an effect holds after the execution of a recipe R^x. Note that if a recipe R^x of G_i makes a proposition *Prop* true only with probability *p(Prop)* the simulation of G_j's planning phase must be carried on starting from both "possible" worlds resulting from the execution of R^x (i.e. one where *Prop* is true and one where *Prop* is false).[9]
Therefore, we simulate separately what G_j would plan if *Prop* were true and if *Prop* were false; since also G_j's recipes may involve uncertain effects, we adopted a simple scheme of multiplying the probability of the different outcomes of G_j's actions with the probability of G_j's initial states in order obtain the set of worlds representing the possible outcomes of G_j's reactions to the plan R^x.

In the following, we report the procedure that, given a plan, produces a set of refined plans together with their expected utility.

```
/* in input the one-level plan of agent $G_i$ (gi) for $\beta_k^{x,i}$ (plan-x-i-k), the identifier of agent $G_j$ (gj),
   the step in charge of $G_j$, i.e. $\beta_m^j$ (action-j-m) and an initial world*/
plan-shared-actions(gi, Plan-x-i-k, gj, action-j-m, initial-world)
   begin
       /* refinement of plan-x-i-k by selecting an alternative or adding the decomposition
           of an action belonging to the plan */
           refined-plans := refine-plan (plan-x-i-k, gi, initial-world);
           final-worlds := nil;
       /* for each possible outcome of each possible alternative */
       for-each plan in refined-plans
       begin /* outcomes of a plan of $G_i$ from the initial worlds (their probability sums to one) */
           for-each world in resulting-worlds(plan, initial-world)
           begin /* save the probability of the outcome of plan */
               prob := world.prob;
               /* simulate $G_j$ planning from an outcome as it were the only possible one */
               world.prob := 1;
               primitive-plans-j := plan(action-j-m, gj, world);
               /* select best plan from $G_j$'s point of view: $G_i$ considers only $G_j$'s
                   best alternative */
               chosen-plan-j := best-plan-EU(primitive-plans-j, gj, world);
               final-worlds := resulting-worlds(chosen-plan-j, gj, world);
               /* restore the probability of the outcomes w that come after world */
               for-each w in final-worlds begin w.prob := w.prob * prob; end
               /* the probability of worlds in final worlds sums to one */
               all-final-worlds := all-final-worlds + final-worlds;
           end
           /* assign to each $G_i$'s alternative the expected utility from $G_i$'s perspective */
           plan.EU := compute-EU(plan, all-final-worlds, gi);
       end/* eliminate plans that are not promising */
       return(filter-plans(refined-plans, gi));
   end
```

---

[9] Using as G_j's initial world one where *Prop* has *p(Prop)* probability to be true, would correspond to the situation in which G_j is planning with uncertainty about *Prop*.