SUPPORTING COOPERATION THROUGH CLANS

Nathan Griffiths

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK
nathan@dcs.warwick.ac.uk

ABSTRACT

Cooperation is the foundation of multi-agent systems, and allows agents to interact to achieve complex goals. In dynamic environments cooperation gives rise to inherent uncertainty since we cannot predict how agents will respond to environmental change. Consequently, agents must have some mechanism for coping with this uncertainty. In particular, agents must manage the *risk* resulting from interacting with others who have different objectives, or who may fail to fulfil their commitments. In this paper, we introduce the concept of a *clan*: a group of agents who trust each other, have similar objectives, and treat each other favourably with respect to cooperation.

1. INTRODUCTION

In a multi-agent system, differences in capabilities, knowledge and resources mean that agents must typically cooperate to achieve their goals. Agents are not benevolent however, and for autonomous[1] agents to offer assistance to others they must receive some individual benefit. Previous work has utilised the notions of motivation and trust to provide a framework for cooperation that accounts for the value received from assisting others [7].

Existing models of cooperation are typically concerned with attaining cooperation to achieve specific tasks. Typically, unless agents have common goals, or similar motivations, *at the time of forming a group* they will not cooperate. However, if in the long-term, their goals are similar, they may have benefited *overall* from cooperating even though there was no direct *immediate* benefit. Existing approaches are also limited in scalability. In particular, an agent must typically consider all other agents in establishing cooperation. As the number of agents increases, the search space and communication cost also increases. *Clans* provide a mechanism for agents to consider the long-term benefits of cooperation and enable *self-organisation* of the space of agents to increase scalability. In this respect we build upon Brooks and Durfee's model of *congregations* [3, 4], taking the essence of congregations and applying it in the broader context of autonomous agents to address scalability issues and to provide a degree of self-regulation in a society of agents.

In this paper we introduce the notion of clans, and describe how they are formed and how they influence agent behaviour. A clan can be viewed as a loosely coupled composite entity whose abilities are represented by the union of its members' abilities. We introduce a flexible, highly configurable, framework for achieving sophisticated cooperation amongst autonomous agents. Numerous factors are involved in governing how an individual cooperates. In this paper we indicate the most significant of these factors, and use them to provide a simple instantiation of the framework.

2. COOPERATIVE AGENTS

We adopt a BDI-based approach and take an agent to comprise: *beliefs* about itself, others and the environment; a set of *desires*, or goals, representing the states it wants to achieve; and *intentions* corresponding to the plans adopted in pursuit of these desires [2]. Agents have a library of partial plans from which to select (we do not assume the ability to plan from first principles). In addition to the traditional BDI model however, we concur with the views of some that motivation is an extra component required for autonomy [11, 14].

Motivations are high-level desires characterising an agent, guiding behaviour and controlling reasoning; they cause the generation and subsequent adoption of goals, and guide reasoning and action at both individual and cooperative levels. An agent has a fixed set of motivations, each having an intensity that varies according to the situation. We represent a motivation by a tuple (m, i, l, f_i, f_g, f_m), where m is the name of the motivation, i is its current intensity, l is a threshold, f_i is the intensity update function, f_g is the goal generation function, and f_m is the mitigation function. Differences between agents are characterised by their motivations, which can lead not only to differences in goals, but also to differences in social behaviour.

[1] We assume that autonomous agents are also self-interested.

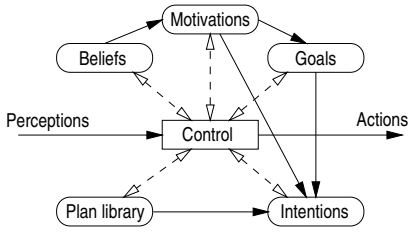

Figure 1: The mBDI agent architecture

The intensities of an agent's motivations change in accordance with its beliefs (as determined by f_i), which in turn are determined by perceptions. When the intensity of a motivation exceeds its threshold, l, a set of goals is generated using the function f_g. Thus, an agent responds to changes in its beliefs, resulting from perception, by generating goals. These goals are evaluated according to their motivational value (i.e. the amount by which their achievement would reduce the motivational intensity, as determined by f_m), and the most important are adopted as intentions by selecting an appropriate plan, and committing to its execution. Finally, an agent selects an intention to pursue and acts toward its achievement, again using motivational value as the guiding measure. The resulting architecture is illustrated in Figure 1, in which solid arrows represent the flow of information, and dotted arrows the control structure. The corresponding reasoning cycle is given in Figure 2.

Motivations can be thought of as embodying the notion of utility (although they provide much more than mere utility). In the context of this paper, motivations can be thought of as ensuring that entities always act so as to maximise their individual utility.

2.1. Trust

Since cooperation involves a degree of risk, arising from the uncertainties of interactions, agents model the trustworthiness of others. The notion of *trust* is widely recognised as a means of assessing the perceived risk in interactions [5, 12]; it represents an agent's *subjective* estimate of how likely another agent is to fulfill its cooperative commitments. As agents interact they can infer trust values based on their experience and, over time, improve their models of trustworthiness. We base our model of trust upon Marsh's formalism [12] and the work of Gambetta [6], and define the trust in an agent, A, to be a value from the interval between 0 and 1, thus $T_A \in [0, 1]$. The numbers merely represent comparative values internal to an agent's individual representation, and are not meaningful in themselves (or indeed directly comparable across agents since they are subjective). Values approaching 0 represent complete distrust, and those approaching 1 represent blind trust.

1. Perceive the environment and update beliefs.
2. For each motivation apply f_i to update its intensity.
3. For each motivation apply f_g to generate a set of new goals.
4. Select an appropriate plan for the most motivated of these newly generated goals, and adopt it as an intention.
5. Select the most motivationally valuable intention and act towards it by performing the next step in the plan.
6. On completion of an intention the mitigation function f_m is applied to each motivation to reduce its intensity according to the motivational value of achieving the goal.
7. Finally, return to the beginning of the cycle.

Figure 2: The mBDI reasoning cycle.

In our approach, trust values are associated with a measure of *confidence*, and as an agent gains experience this confidence increases. Trust values are inferred according to an agent's *disposition*: optimistic agents infer high values, while pessimists infer low values. This disposition also determines how trust is updated after interactions [13]. After a successful interaction, optimists increase their trust more than pessimists, and conversely, after an unsuccessful interaction pessimists decrease their trust more than optimists. The magnitude of change in trust is a function of several factors depending on the agent concerned, including the current trust and the agent's disposition. An agent's disposition is embodied by the default trust value that is ascribed in the lack of any other information, and functions for updating trust after successful and unsuccessful interactions.

Trust values are based on experience, and it is important to consider the recency of such experiences. In particular, we assume that trust decays over time, and that given a sufficient period of time an agent's trust of another will tend towards the default value. This means that the positive effect of successful interactions on trust will reduce over time, as will the negative effect of unsuccessful interactions. The rate at which trust decays is individual to a particular agent, and is a function of that agent's memory length.

3. COOPERATION FRAMEWORK

Previous work has described a framework for cooperation based upon the notions of trust and motivation [7]. Cooperation is more than simultaneous actions and in-

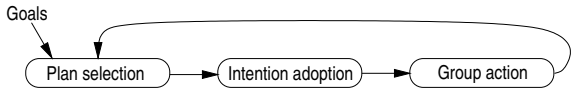

Figure 3: Stages of cooperation

dividual intentions; agents need some form of *commitment* to the activity of cooperation itself [1, 10] along with an appropriate set of *conventions* [15] specifying when and how a commitment can be abandoned. Where a group forms appropriate commitments to cooperate and adopts suitable conventions we say that they have formed a *cooperative intention*. There are several distinct tasks surrounding the formation and execution of a cooperative intention: plan selection, intention adoption, and group action (as illustrated in Figure 3).

3.1. Plan Selection

Motivations give rise to goals that must be adopted as intentions by selecting a plan and committing to its execution. Since plans may require cooperation an agent must consider, when selecting a plan, the nature of those it might cooperate with. Specifically, it should consider the likelihood of finding agents to assist and the likelihood that they will execute the plan successfully, i.e. their trustworthiness. We have described elsewhere a mechanism for assessing the contributions contained in a plan, in terms of the risk associated with the agents who are believed capable of executing them [8]. By combining this risk with plan cost we obtain a measure for selecting between plans that balances an agent's desires to minimise cost and risk.

3.2. Intention Adoption

After selecting a plan, an agent commits to its execution by forming an intention. If no assistance is required the plan can simply be adopted, and action can begin, otherwise the agent must solicit assistance towards its execution. In order to gain assistance, the initiating agent annotates each contribution in the plan with the identifier of the agent considered best able to perform it, based on knowledge of their capabilities, and their believed trustworthiness [9]. The assistance of these agents is then be requested. On receiving a request for assistance, agents inspect their own motivations and intentions to decide whether or not to agree, and send an appropriate response. If sufficient agents agree then a cooperative intention can be established among them. However, if insufficient agents agree then either failure is conceded, or intention adoption is retried.

3.3. Group Action

Once a a cooperative intention has been form the *group action* stage is entered and the plan is executed. On successful completion of the cooperative intention, the agents dissolve their commitment and cooperation is finished. Alternatively, if execution fails, the agent that first comes to believe this informs the others in accordance with the specified conventions, and again their commitments are dissolved. In both cases, agents update the information they store about others to aid future decisions about cooperation. In particular, the trust values inferred for these agents are updated.

4. CLAN FORMATION CRITERIA

Clans provide a means to address some of the limitations of existing approaches to cooperation. We view clan formation as a *self-interested* activity — an agent attempts to form a clan for its own benefit alone, and not in response to any external influence. In particular, an agent can form a clan to: minimise missed opportunities for cooperation, address problems of scalability, cope with a lack of information, and increase the robustness of commitments. To determine when to form a clan, an agent must assess the extent to which these issues are affecting performance. Figure 4 gives a skeletal algorithm outlining the decision process. In the remainder of this section we describe the component steps of this algorithm.

4.1. Missed opportunities

Since agents act according to their motivations, any responses to requests for assistance are determined by motivations. Missed opportunities for cooperation can arise when agents' motivations are out of step in time, leading to reduced benefit in the long-term. In dynamic environments, short-term fluctuations in the intensity of motivations can lead to failures to establish cooperation that would actually benefit individuals longer term. Therefore, an agent needs some means of assessing the extent to which it is missing such opportunities. Since, motivations are private an agent cannot inspect others' motivations to determine whether missed opportunities are occurring. Instead, agents must consider the converse, namely those requests from others that were declined. If few requests have been declined then there are, at most, few missed opportunities (from this agent's perspective). However, if many requests have been declined then there *may* be many missed opportunities; each missed opportunity leads to a declined request, although requests may be declined for other reasons. When an agent experiences a high

function ASSESS-WHEN-TO-FORM-CLAN **returns** boolean
  local: *missed-opportunities* ←false
    scalability ←false
    lack-of-information ←false
    high-failure-rate ←false
  if (*plan-annotation-failure-rate* > *annotation-threshold*)
    and (MOTIVATIONAL-VALUE(FILTER(*previous-rejected-requests*)) > *rejection-threshold*)
      then *missed-opportunities* ←true
  if (PROPORTION-COOPERATIVE(*recently-applicable-plans*) > *scalability-threshold*) **then** *scalability* ←true
  if (AVERAGE-TRUST(*agent-models*) < *trusted-threshold*)
    or (AVERAGE-CONFIDENCE(*agent-models*) < *confidence-threshold*)
    and (**exists** *agents* **such-that** (AVERAGE-TRUSTWORTHINESS(*agents*) > *trusted-threshold*)
    and (AVERAGE-CONFIDENCE(*agents*) > *confidence-threshold*))
    then *lack-of-information* ←true
  if (*failure-rate* > *failure-threshold*) **then** *high-failure-rate* ←true
  if (*missed-opportunities* **or** *scalability* **or** *lack-of-information* **or** *high-failure-rate*) **then return** true **else return** false
end

Figure 4: Assessing when to form a clan.

rate of plan annotation failure (above the *annotation-threshold*) it should inspect previous incoming requests (within some memory limit). If the current motivational value of previously rejected requests (that are similar to the current plan) exceeds the *rejection-threshold*, then we take the agent to be at risk of missed opportunities. This heuristic represents a simple approach for assessing missed opportunities and, as with other aspects of this framework, more sophisticated approaches are possible. For example, to determine when a declined request represents a missed opportunity rather than a one-off occurrence an agent might consider the extent of fluctuation in motivations, or the motivational value of the request over previous iterations.

4.2. Scalability

When establishing cooperation, an agent typically considers all other agents' suitability. Identifying and communicating with these agents is costly. The significance of this cost is indicated by the number of agents modelled by an individual. The frequency of cooperation must also be considered — if cooperation is rare, then the impact is much less than if each plan requires cooperation. The proportion of an agent's plans that are cooperative influences the frequency with which it cooperates. However, since agents may not utilise all of their plans, we can filter out those that are less likely to be significant. In particular, we can measure the proportion of plans that are cooperative in the last n reasoning cycles, where n corresponds to the agent's memory length, i.e. we consider all *applicable plans* in last n iterations. An agent can inspect the set of recently applicable plans and, if the proportion of these that are cooperative exceeds the *scalability-threshold* then it should attempt to form a clan.

4.3. Lack of information

Cooperation is inhibited when an agent has insufficient information about others' trustworthiness or capabilities. If an agent's knowledge of others indicates that there are many untrusted agents, or little is known about others' capabilities, then clan membership may be beneficial.

In modelling the trustworthiness of others an agent maintains a measure of confidence in its trust assessments. Trust models based on limited experience are given low confidence in comparison with those based on extensive experience. In considering clan formation an agent must consider both the extent of its models and the confidence placed in them. There is a lower bound below which clan formation is not practical due to lack of information (or confidence) about its potential members. In particular, it is only sensible to form (or join) a clan with agents who are trusted to a reasonable degree of confidence. Therefore, an agent should inspect its models of others; if there are many untrusted agents or agents whose models have low confidence, it should attempt to form a clan, provided there is a subset of confidently trusted agents with whom to do so.

4.4. High failure rate

In dynamic environments fluctuations in motivation intensity can lead to failure. Clan membership gives an additional degree of commitment to cooperation and may help reduce a high failure rate at execution time. Membership of a clan provides an additional degree

```
function FORM-CLAN
  input: redundancy, timeout
  local: current-plans ←{}
    min-size ←0
    ideal-size ←0
  target-agents ←SELECT-MOST-TRUSTED(agent-models, confidence-threshold)
  current-plans ←SELECT-PLANS(EXTRACT-GOALS(active-motivations), plan-library)
  for agent in target-agents do
    if (BELIEVED-CAPABLE(agent, EXTRACT-ACTIONS(current-plans)) = false)
      or (TRUST(agent, agent-models) < trust-threshold) then target-agents ←target-agents - agent
  end
  min-size ←#(EXTRACT-ACTIONS(current-plans)) / #(current-plans)
  ideal-size ←ideal-size × redundancy
  goals-to-communicate ←EXTRACT-GOALS(active-motivations)
  if (#(target-agents) > ideal-size) then target-agents ←TAIL(target-agents, ideal-size)
  for agent in target-agents do REQUEST-FORM-CLAN(goals-to-communicate) end
  responses ←GET-RESPONSES(timeout)
  if (#(responses) > min-size) then
    for agent in ACCEPT(responses) do CONFIRM(agent) end
    return true
  else for agent in ACCEPT(responses) do DECLINE(agent) end
    return false
end
```

Figure 5: Clan formation.

of commitment and, importantly, a mechanism for an agent to obtain motivational value through acting in what may appear to be a semi-benevolent manner. Thus, if an agent is experiencing high execution failure rates, it may be appropriate to form a clan.

5. CLAN FORMATION

If any of the factors for forming a clan are currently relevant, then the agent should attempt to form a clan. In common with other aspects of cooperation, clan formation is guided by trust and motivation. The influence of motivation is determined by the motivational value that may be gained by clan membership (in terms of a higher success rate and quality of future interactions). This is accounted for indirectly by the decision to form a clan in the first instance. The influence of trust, however, is more direct. Clans should only be formed with trusted agents. At a fundamental level, trust determines whether it is practical to form a clan. If an agent has a low trust in others or has low confidence in its models of those it considers trustworthy, it should not form a clan. However, if it has adequate trust in others, it can attempt to form a clan with the most trusted agents possible. Since an agent forms a clan in response to its current situation, it should consider the nature of potential future interactions and target its requests accordingly. In particular, an agent should attempt to form a clan with those agents whose assistance is likely to be required. Since the environment is dynamic and unpredictable this cannot be assessed directly. The set of active motivations, however, tends to be relatively static in the medium-term, and can be used to identify the goals that might be generated in the future. From these goals the set of possible plans and actions for which assistance may be required, can be identified. Using these actions the agent can select the most trusted agents who are believed to have suitable capabilities to assist it in the future.

The number of agents to send requests to depends on the current situation. In general, the more agents that are in a clan the better (computational cost excepted). In particular, ideally all agents to whom requests for assistance are likely to be sent would be clan members (since their acceptance is more likely). However, the disadvantages of large clans are a computational overhead and that there are more agents to whom assistance is inclined to be offered. We take a simple approach to assessing the ideal clan size, based on the current situation. In particular, we consider the plans that are likely to be adopted in the future (as described above), and determine the average number of actions for which assistance is required. We use this to estimate the minimum ideal size. To determine the actual ideal size we add a degree of redundancy to cope with the likelihood that not all agents to whom requests are sent will agree to join the clan.

After determining the most trusted agents, they are

```
function PROCESS-FORMATION-REQUEST
    returns response
  input: requestee, request-goals
  local: motivational-value ←0
  if TRUST(agent-models, requestee) < trust-threshold
    then return decline
  if (ATTEMPT-TO-FORM-CLAN) then return accept
  for goal in request-goals do
    motivational-value ←motivational-value
        + MOTIVATIONAL-VALUE(goal)
  end
  if motivational-value > threshold return accept
    else return decline
end
```

Figure 6: Responding to a clan formation request.

sent a request to form a clan. Ideally, no further information would be required, since they might also consider it beneficial to join a clan based on their own assessments of the factors described in Section 4. However, due to differences in experience this is generally not the case. In particular, although an agent may benefit from clan membership, its own assessment of whether to form a clan (using the algorithm in Figure 4) may not indicate this. An agent must, therefore, give some incentive for joining the clan. Since we do not assume that agents have negotiation or persuasion capabilities abilities, we take a simple mechanistic approach. Specifically, the request must include the set of most frequently generated goals from the most active motivations, as a means for others to assess the usefulness of clan membership. The disadvantage to this approach lies in revealing what is essentially private information. However, since the agent should (hopefully) gain motivational benefit from forming the clan, we argue that this is justified. Figure 5 outlines the requesting process. The first part of the algorithm is concerned with determining who to invite to join a clan. Once these agents are identified, each is sent a request.

On receiving a request to join a clan an agent must consider the motivational value of joining and its trust of the requestee. Firstly, if the trust of the requestee is below the minimum trust threshold, the request is declined. Secondly, the criteria described in Section 4 are considered to give an indication of how beneficial clan membership would be *in general*. If the assessment of the general value of clan membership is such that the agent desires to form a clan, then the request is accepted. Finally, the goals contained in the request are used to estimate how useful it would be to join the clan *in particular*. The motivational value of each goal is considered in a situation independent manner. If this value exceeds a threshold then the agent agrees to form a clan (assuming the requestee is trusted). This process is outlined in Figure 6 in the function PROCESS-FORMATION-REQUEST.

If sufficient agents respond positively (i.e. more than the minimum clan size) then the initiator sends acknowledgements and a clan is formed (with those who accepted). Alternatively, if insufficient agents accept then those agents that did accept are informed and clan formation abandoned. This process is shown in Figure 5 in the latter part of the function FORM-CLAN.

6. REASONING IN A CLAN

There are three aspects to the influence of clan membership on behaviour: sharing of information, increased commitment to cooperation and improved scalability. In the first case, if a clan member requires assistance but does not know of any trusted agents having the necessary capabilities, then it can request information from other clan members. In the second case, clan members are not only more likely to cooperate, but are also more likely to fulfil their commitments due to the motivational value of cooperation. In order to ascribe motivational value to clan membership, and to ensure that agents remain self-interested, we introduce an additional *kinship motivation* to all agents. This motivation is mitigated by offering assistance to other clan members. Kinship intensity is determined by such factors as the proportion of goals that require cooperation, and the extent and quality of the information an agent has about others. The final benefit of clan membership is that the search cost of finding cooperative partners can be reduced by simply searching through the members of the clan. This allows agents to address the scalability problems described in Section 4.

6.1. Sharing Information

Clan membership causes agents to be inclined to share information about third parties with other clan members. Through this mechanism agents can discover other trusted agents (outside the clan) to assist them. Where an agent is faced with a plan containing actions for which it knows of no trusted and capable agents it asks the trusted[2] members of its clan. The process of requesting information from other clans members is outlined in Figure 7.

Other clan members gain motivational value, via the *kinship* motivation, from offering information about

[2]Although by definition, all members of the clan will have been trusted at the time of formation, some of them may have come to be distrusted over time, but not so much as to justify leaving the clan. Thus, their trustworthiness should be checked when interacting with them.

```
function REQUEST-INFORMATION
  input: plan, timeout
  local: problem-actions ←{}
    trusted-agents ←{}
    responses ←{}
  for action in plan do
    trusted-agents ←{}
    for agent in CAPABLE(agent-models,action)
      if (TRUSTED(agent, trust-threshold)) then trusted-agents ←agent
    if (trusted-agents = {}) then problem-actions ←problem-actions ∪ action
  end
  if (problem-actions ≠ {}) then
    target-agents ←SELECT-TRUSTED(CLAN-MEMBERS( agent-models), confidence-threshold)
    for agent in target-agents do REQUEST-INFORMATION(problem-actions) end
  responses ←GET-RESPONSES(timeout)
  for action in problem-actions do
    agent-models = agent-models ∪ REPUTATION(FILTER-CAPABLE(responses, action))
  end
end
```

Figure 7: Requesting information

which agents have the required capabilities, and the degree to which they are trusted. The extent of the motivational value received from providing information is determined by the intensity of the kinship motivation. If this motivation is above its associated threshold, then the agent should offer information, otherwise insufficient benefit is received to justify sharing information. Additionally, information should only be shared with agents that are trusted, and so before responding to a request an agent should check that the requestee is trusted above the minimum trust threshold.

A significant problem exists in communicating trust information, namely *subjectivity*. In particular, trust values are internal to an agent and depend on its disposition and experience; they are not directly comparable across agents. One approach, is to eliminate small subjective differences between agents by communicating a stratification of trust, where the numerical range divided into subranges [12]. The advantage of stratification is that differences between agents are removed, provided those differences are within the same subrange. However, if a difference in values occurs across subranges, stratification is counter-productive and accentuates the difference. Moreover, the use of stratification leads to a loss of sensitivity and accuracy; it becomes impossible to distinguish between values that are in the same subrange. Stratification also only addresses subjectivity problems if there are small differences in values between agents. In particular, this requires that for the trust ascribed to an agent to be in the "highly-trusted" subrange, the same *meaning* is inferred by different agents. However, since agents' default trust values and dispositions are individual there is no guarantee that two different agents infer the same meaning from a given subrange.

In our view, the loss of sensitivity and accuracy resulting from stratification, coupled with its relatively limited applicability, mean that its use is not appropriate. We therefore take the more simple approach of agents simply communicating numerical values, in the knowledge that these values are not directly comparable across agents. Figure 8 outlines the process of providing information from to other clans members, in the function PROVIDE-INFORMATION.

In sharing trust information we adopt two key constraints, as proposed by Marsh [12]: if agent A_1 obtains information about A_3 from A_2 then (i) A_1 does not trust A_3 more than A_2 trusts A_3, and (ii) A_1 does not trust A_3 more than it trusts A_2. Thus, any trust information is mediated by the trust ascribed to the provider.

Since the information about A_3 is based on another agent's subjective view, the result is an assessment of the *reputation* of A_3. Thus, we use the term *trust* to refer to an individual's assessment of another, and the term *reputation* to refer to the assessment of an agent based on others' trust values; trust is an individual notion while reputation is a social notion representing a collective estimate of trustworthiness[3].

To determine the reputation of an agent, based on information provided by a set of clan members, we take the average mediated value. In practice, we take a simple approach to determining this, such that the reputation from the perspective of agent A_x of agent

[3]It is important to note that since reputation incorporates an assessment of the trust of the information providers, different agents are likely to arrive at different reputation assessments for a given agent.

```
function PROVIDE-INFORMATION
  input: problem-actions, requestee
  local: response ←{}
    agent ←null
  if (INTENSITY(kinship) > THRESHOLD(kinship)) and
    (TRUST(requestee, agent-models) > trust-threshold)
    and (requestee ∈ CLAN-MEMBERS(agent-models))
    for action in problem-actions do
      agent ←SELECT-MOST-TRUSTED(agent-models,
        confidence-threshold, action)
      response ←response ∪
        (agent, TRUST(agent, agent-models), action)
    end
  SEND-RESPONSE(response)
end
```

Figure 8: Providing information

A_y, based on information provided by clan members $A_1, A_2, \ldots, A_n$ is as follows.

$$R_{xy} = \frac{\sum_{i=1}^{n} T_{A_x A_i} \times T_{A_i A_y}}{n}$$

where T_{ij} denotes the trust A_i ascribes to A_j and R_{ij} denotes the reputation A_i has determined for A_j. The latter part of Figure 7 indicates how an agent determines the reputation of another, where the function REPUTATION is assumed to implement the above equation.

6.2. Cooperation through *kinship*

Secondly, clan members are more likely to cooperate and to fulfil their commitments due to the motivational value of cooperation, specifically due to the *kinship* motivation. Kinship functions like any other motivation — its influence is taken into account when deciding whether to cooperate, and in determining when to rescind commitments. Thus, no additions are required to the agent's reasoning cycle to incorporate this inclination to assist clan members. At a philosophical level, introducing a kinship motivation can perhaps be seen to undermine the fundamentally self-interested nature of agents. Recall, however, that agents choose to join a clan for specific reasons that are undeniably self-interested. Furthermore, the kinship motivation is just one of a set of motivations, and does not override the others; if it did then the agent would certainly cease to be self-interested. Given sufficient information and reasoning resources the kinship motivation could be avoided, since an agent would be able to reason explicitly about the benefit it may in the future receive from agreeing to cooperate, or sharing information. However, in practise such information and reasoning resources are unrealistic, and we take this simple approach of introducing an additional motivation.

6.3. Scalability

Finally, if an agent joined a clan to address scalability problems, i.e. to reduce the search cost of finding cooperative partners, then it can simply search through the members of the clan. This is a special case arising from the reason for forming a clan. Due to space constraints we do not give details here, but in broad terms an agent goes through the standard process of attempting to form a cooperative intention but is restricted to considering agent models corresponding to clan members. If this fails, then the standard cooperative intention formation procedure is undertaken.

7. MAINTENANCE OF CLANS

Over time an agent's active motivations change. Clan membership addresses short-term fluctuations in motivations leading to missed cooperation opportunities. However, in the long-term as the set of active motivations changes, the benefit gained from membership of a specific clan will decrease. Consequently, one or more clan members will eventually no longer receive sufficient benefit to justify continued membership. Clan membership has a cost, both in computational overhead and because an agent may act to assist another clan member, rather than as it would otherwise. Provided the clan is operating effectively there will be sufficient reciprocal action for each member to receive net benefit overall. However, if the set of active motivations changes then it may no longer receive benefit from the clan and should withdraw its membership by notifying the other members. Agents should also withdraw their membership if they come to distrust the other members.

Ideally, an agent would assess the costs and benefits of clan membership, and if the costs outweigh the benefits it should leave the clan. Unfortunately, however, this is not possible to assess from an agents' perspective since there is no way to interrogate what others would do without kinship motivations. If there are many goals achieved through cooperation and/or there is a high cooperation rate then clan membership is likely to be worthwhile. It is not possible to assess whether an agent is getting something in return for kinship, or whether others' clan membership is affecting their behaviour. However, from the agent's viewpoint this does not matter — provided the agent is successful in gaining cooperation we view clan membership as beneficial. (Note that even from an external viewpoint there are many subtle benefits to clan membership that are difficult to assess, such as being more trusted by potential partners due to being "exploited".)

We take a simple approach to assessing whether to leave a clan, by assessing its relevance and the influ-

```
function LEAVE-CLAN returns boolean
  input: recent-applicable-plans
  if (#(COOPERATIVE(recent-applicable-plans))
    / #(recent-applicable-plans) < relevance-threshold)
    then return true
  clan-interactions ←CLAN(recent-interactions)
  none-clan-interactions ←recent-interactions
    - clan-interactions
  if #(SUCCESSFUL(clan-interactions))
    / #(SUCCESSFUL(none-clan-interactions))
      < success-threshold)
    then return true
  if #(UNSUCCESSFUL(clan-interactions))
    / #(UNSUCCESSFUL(none-clan-interactions))
      > failure-threshold)
    then return true
  return false
end
```

Figure 9: Assessing when to leave a clan

ence of clan membership on cooperative success. In particular we consider the proportion of recently adopted plans that required cooperation. If this proportion is below a minimum *relevance-threshold* then the agent should leave the clan. To access the effect of clan membership on cooperative success we consider the proportion of successful and unsuccessful interactions that involved clan members. If the proportion of successful interactions that involved clan members is less than the *success-threshold* then the agent should leave the clan. Conversely, if the proportion of unsuccessful interactions involving clan members is greater than the *failure-threshold* then the agent should leave. This decision process is outlined in Figure 9. Each individual makes its own decision about whether to stay in a clan or leave — there is no formal clan dissolution process. As the number of agents that remain in a clan decreases there will eventually be a single agent remaining, at which point the clan ceases to exist.

8. JOINING EXISTING CLANS

We have given an overview above of how agents form clans and how they act within them; we have described how clan membership is monitored as the environment changes, and the circumstances in which agents should leave a clan. To be flexible, however, and to allow effective partitioning of the space of agents according to the current situation, it is desirable for agents to be able to join existing clans. The key problem in enabling agents to join existing clans is the provision of a mechanism for agents to *discover* which clans exist. A simple solution to this would be to introduce a directory of existing clans. However, since in our model, there is no centralised control or repository, such a centralised approach would be inappropriate. Furthermore, there would be no clear motivational value for agents to provide information about their clan membership to such a repository in order to be interrogated by other, potentially distrusted, agents.

Our alternative is to provide two means for an agent to discover an existing clan: by invitation from an existing member, or by requesting that an existing member joins a new clan. The first case is a straightforward extension of the initial criteria for whether to form a clan. Suppose an agent comes to believe that it should form a clan (using the algorithm given in Figure 5), but on assessment of who to request discovers that a high proportion of the desired members of the new clan are already members of an existing clan to which it already belongs. In this case, the agent can instead send an invitation to join the existing clan to those agents who are not already members. A result of the self-interested nature of agents is that in such a situation, the inviting agent does not ask 'permission' from the existing clan members, rather it should simply inform them about any positive responses from newly invited agents and those agents should update their knowledge of the clan accordingly.

Our second alternative occurs where an agent sends a request to form a clan to agents who are already members of an existing clan. In this case, each member that receives a request either responds in the standard manner described earlier in Section 5 or can invite the proposed members of the new clan to join an existing clan. If the goals communicated by the requestee are similar to the goals that caused the formation of an existing clan, then an invitation to join the existing clan is appropriate, provided all of the agents concerned are suitably trusted. On receiving an invitation to join an existing clan, agents assess the request using the same criteria as a standard clan formation request (as described by the function PROCESS-FORMATION-REQUEST in Figure 5).

9. CONCLUSION

In this paper we have described how clans address some of the limitations of existing approaches to cooperation. In particular they can minimise missed opportunities for cooperation, address problems of scalability, cope with agents' lack of information, and increase commitment robustness. We have briefly described the process through which agents can form, maintain, and reason within clans. Clans provide a mechanism for agents to improve their individual performance through cooperation without compromising their autonomy. A clan

is a loosely coupled entity, and a clans' actions, and indeed its continued existence, depends solely on the self-interested decisions of its members. Any notion of collective intelligence is a transient quality dependent on the current state of the clan's members. A clans' capabilities and knowledge can be viewed as the union of its members' capabilities and knowledge. However, there is no corresponding notion of a clan's motivations, and clan members remain autonomous self-interested entities. The continued robustness and flexibility benefits that result from this individual autonomy are a key advantage of our approach.

There are three key areas of ongoing work. Firstly, we are investigating more sophisticated mechanisms for managing the membership of multiple clans. Currently, agents do not explicitly reason about multiple clans, and they manage multiple clan memberships implicitly by simply acting according to their motivations. Secondly, we are developing an ontology for sharing trust information. This can be seen as an extension of the stratification approach introduced in Section 6 where agents agree on an ascribed *meaning* to the particular trust notions. For example, agents may agree that "highly-trusted" implies a certain degree of previous success given a particular degree of experience. Finally, although we have undertaken limited experimentation of our approach, with favourable initial results, we intend to perform more extensive experimentation.

10. REFERENCES

[1] M. E. Bratman. Shared cooperative activity. *Philosophical Review*, 101(2):327–341, April 1992.

[2] M. E. Bratman, D. Israel, and M. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[3] C. Brooks and E. Durfee. Congregating and market formation. In *Proceedings of the First International Joint Conference on Autonomous Agents in Multi-Agent Systems*, pages 96–103, Bologna, Italy, 2002. ACM Press.

[4] C. Brooks, E. Durfee, and A. Armstrong. An introduction to congregating in multiagent systems. In E. Durfee, editor, *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 79–86, 2000.

[5] C. Castelfranchi and R. Falcone. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, pages 72–79, Paris, France, 1998.

[6] D. Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[7] N. Griffiths. *Motivated Cooperation in Autonomous Agents*. PhD thesis, University of Warwick, 2000.

[8] N. Griffiths and M. Luck. Cooperative plan selection through trust. In F. J. Garijo and M. Boman, editors, *Multi-Agent System Engineering: Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'99)*. Springer-Verlag, 1999.

[9] N. Griffiths, M. Luck, and M. d'Inverno. Annotating cooperative plans with trusted agents. In R. Falcone and L. Korba, editors, *Proceedings of the Fifth International Workshop on Deception, Fraud and Trust in Agent Societies*, 2002.

[10] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 94–99, Boston, MA, 1990.

[11] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 254–260. AAAI Press/The MIT Press, 1995.

[12] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

[13] S. Marsh. Optimism and pessimism in trust. In *Proceedings of the Ibero-American Conference on Artificial Intelligence (IBERAMIA '94)*, 1994.

[14] T. J. Norman. *Motivation-based direction of planning attention in agents with goal autonomy*. PhD thesis, University of London, 1996.

[15] M. Wooldridge and N. R. Jennings. Formalizing the cooperative problem solving process. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence (IWDAI-94)*, pages 403–417, Lake Quinhalt, WA, 1994.