Cooperative clans

Nathan Griffiths
Department of Computer Science, University of Warwick, Coventry, UK

Abstract

Purpose – To provide a mechanism for agents to form, maintain, and reason within medium-term coalitions, called clans, based upon the notions of trust and motivation.

Design/methodology/approach – The model is based upon the notions of trust (representing an agent's assessment of another's honesty and reliability) and motivations (which represent an agent's high-level goals). The paper describes the motivational factors that can lead to clan formation, a mechanism for agents to form a clan or join an existing clan, and subsequently how clan membership influences behaviour (in particular though sharing information and acting on behalf of other members). Finally, describes the conditions under which agents leave a clan.

Findings – The proposed mechanism shows how agents can form medium-term clans with trusted agents based on motivations that are essentially self-interested. It is shown how this mechanism can be used to reduce missed opportunities for cooperation, improve scalability, reduce the failure rate and allow sharing of trust information (i.e. establish a notion of reputation).

Originality/value – Proposes a new approach to coalition formation based on the notions of trust and motivation, which allows self-interested agents to form medium-term coalitions (called clans) to increase their own (motivational) returns.

Keywords Modelling, Cybernetics, Control systems, Intelligent agents, Trust, Motivation (psychology)

Paper type Research paper

1. Introduction

Agents in a multi-agent system typically must cooperate to achieve their goals, due to differences in their capabilities, knowledge and resources. In general, agents are not benevolent and to cooperate 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 individual benefit received from cooperating with others (Griffiths, 2000). Motivation and trust are the fundamental components on which cooperation is built. Motivations represent an agent's high-level desires and determine the desirability of cooperating with respect to a particular situation. Trust embodies an agent's assessment of the risk involved in cooperating with another, and enables the uncertainty involved in cooperating to be managed. In this paper we extend the notion of using motivations and trust to achieve cooperation, by providing a mechanism for agents to form medium-term coalitions, called *clans*, to enhance their future interactions. Clan formation was previously described in Griffiths (2003). In this paper, we describe the process of clan formation in more detail, and describe how agents reason in, and maintain, clans.

Previous approaches to cooperation can be broadly divided into two groups: task-based and coalition-based. Task-based approaches, such as Tambe (1997), are concerned with attaining short-term cooperation to achieve specific tasks. Unless agents have common goals, or similar motivations, *at the time of establishing cooperation* they will not cooperate. If agents' goals are similar in the long-term (but out of step in terms of time), they may have benefited *overall* from cooperation even if there

Kybernetes
Vol. 34 No. 9/10, 2005
pp. 1384-1403

0368-492X
DOI 10.1108/03684920510614722

was no *immediate* benefit. In the long-term, task-based approaches can cause opportunities for beneficial cooperation to be missed. Furthermore, task-based approaches tend to require a group of agents to be re-established for subsequent tasks, even if the tasks and group members are similar. A long-term approach, where cooperative groups persist and are re-used where appropriate, can significantly reduce the computation required to achieve cooperation. Thus, we must take a long-term view to avoid missed opportunities and reduce the computation in establishing cooperation.

Coalition-based approaches take a long-term view, although often directed toward a specific goal, where the benefit of joining a coalition tends to be assessed according to the utility gained by the group if a coalition is formed (Breban and Vassileva, 2001; Klusch and Shehory, 1996; Shehory and Kraus, 1995). Calculation of such utility often requires agents to reveal their individual utilities, and does not account for any motivational reasons an agent might have. Motivations represent agents' high-level desires, and motivational value cannot be compared across agents since these desires differ. Consequently, utility-based coalition formation approaches cannot be directly applied to motivated agents, and a motivation-based approach is required.

Existing approaches are also limited in terms of their scalability. In particular, all known agents must typically be considered when establishing cooperation. As the number of agents increases, the search space and communication cost also increases. *Congregations* aim to reduce this cost, such that rather than searching the whole population, agents congregate into interest groups and search within the congregation (Brooks and Durfee, 2002; 2003). Since a single goal connecting members of a congregation is not required, some of the limitations of task-based approaches are avoided. Agents are divided into *labellers* and *congregators* and the former label their congregations so as to attract similar congregator agents. *Clans* take the essence of congregations and allow agents to consider the long-term benefits of cooperation; clans enable *self-organisation* of the space of agents to increase scalability (Griffiths and Luck, 2003).

Clans are loosely coupled composite entities, and are similar to congregations in representing groups of agents. The key distinction is that similarity is defined for clans in *motivational* terms, and the notion of *trust* binds the group together. In forming clans agents are not explicitly divided into labellers and congregators, instead these roles are implicitly incorporated into the cooperative process. In particular, when an agent wishes to form a clan, or to increase the membership of an existing clan, it can act analogously to a labeller by requesting others join the clan. Other agents act similarly to congregators by evaluating the initiator's stated interests to assess whether to join the clan. Numerous factors are involved in governing how an individual cooperates. In this paper, we indicate the most significant of these factors, namely trust, motivation and reputation, and use them to provide a flexible framework for cooperation. The relationship between these factors is summarised in Figure 1. An agent's trust models,

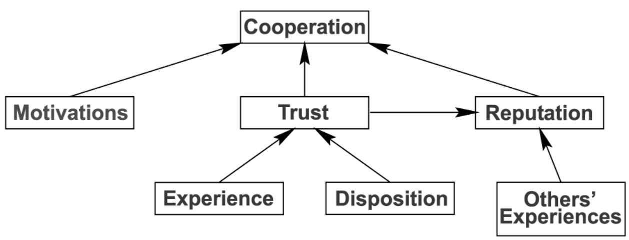

Figure 1.
Overview of the factors in cooperation and clan formation

i.e. the level of trust it places in others, are determined by its own individual experience and disposition. Reputation is determined both by individual trust, and information given by others about their own experiences (and the trustworthiness of those providing information). An agent's decision to cooperate and its subsequent commitment is a function of its motivations, trust of others, and the reputation it perceives potential cooperative partners to have.

The remainder of this paper is structured as follows. In Section 2, we introduce the nature of the agents that we are concerned with. Section 3 describes the set of criteria that agents use to determine when to form a clan. Sections 4 and 5 describe how clans are formed, and how they influence cooperation, respectively. The conditions under which agents should leave a clan are discussed in Section 6. In Section 7, we describe how agents can join existing clans. Finally, in Section 8 we identify areas of future work, and conclude the paper.

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 plans adopted in pursuit of these desires (Bratman *et al.*, 1988). In addition to the traditional BDI model, however, we concur with the views of some that *motivation* is an extra component required for autonomy (Castelfranchi, 1995; Luck and d'Inverno, 1995; Norman, 1996) and we refer to the resulting architecture as motivated BDI, or *mBDI*. In accordance with the standard BDI model, agents also have a library of partial plans from which to select the most appropriate to achieve their goals. Actions within a partial plan can be individual, joint or concurrent. Individual actions are performed by a single agent alone, joint actions require simultaneous contributions from two or more agents, and concurrent actions comprise a set of individual or joint actions to be performed concurrently with synchronisation at the start and end of a concurrent block. Joint and concurrent actions correspond to the notions of strong and weak parallelism introduced by Kinny *et al.* (1992). Partial plans can also contain subgoals, for which subplans must be selected at execution time.

Motivations facilitate autonomy, and 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. Differences between agents are characterised by their motivations, which can lead to both differences in goals, and in social behaviour. An agent has a fixed set of motivations, each having an intensity that varies according to the current situation. For example, suppose an agent's motivations include "hunger" and "survival". If the agent's energy is low then the intensity of the "hunger" motivation will be high, causing the generation of a goal to eat food. However, while the intensity of a given motivation fluctuates, motivations themselves are not transient and the set of motivations belonging to a particular agent does not change. Thus, although the agent's "survival" motivation may have a low intensity and not be contributing to the current behaviour, the motivation is always present and may become active in certain situations. Motivations provide an agent with autonomy and allow it to generate goals in response to changes in its environment and select appropriate actions to perform. Furthermore, as we discuss in the remainder of this paper, motivations guide decisions with respect to cooperation. As with individual actions, cooperative activity must be *motivated*.

A single motivation is represented 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 the goal generation function, and f_m the mitigation function. As in the standard BDI approach, mBDI agents perceive their environment using their sensors and update their beliefs according to these perceptions. The intensities of an mBDI agent's motivations change in accordance with its beliefs (through the application of f_i). Thus, perceptions determine both the beliefs that agents hold, and the intensities of their motivations.

Motivations provide a way for agents to have goals appropriate to the situation. If the current situation causes the intensity of a motivation to exceed its threshold, l, then a set of goals is generated using the function f_g. 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. The agent then selects an intention to pursue and acts toward its achievement, again using motivational value to guide its choice. This mechanism is embodied by the mBDI reasoning cycle.

(1) Perceive the environment and update beliefs.
(2) For each motivation apply f_i to update its intensity based on the current perceptions.
(3) For each motivation whose intensity i is greater than the threshold l apply f_g to generate a set of new goals.
(4) Select an appropriate plan for the most motivationally valuable of these generated goals, and adopt it as an intention.
(5) Select the most motivationally valuable intention and act toward it by performing the next step in the plan.
(6) On completion of an intention, apply the mitigation function f_m 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.

We represent a complete mBDI agent as a tuple (M, B, D, I, PL) where M signifies the agent's set of motivations, B represents its beliefs, D corresponds to the agent's desires (or goals) as generated from its motivations, I are the intentions that it is committed to, and PL is the plan library. The resulting architecture is illustrated in Figure 2, in which solid arrows represent the flow of information, and dotted arrows the control structure.

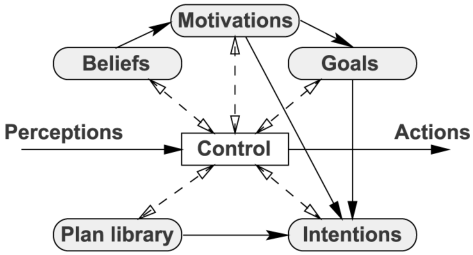

Figure 2.
The mBDI agent architecture

2.1 Trust models

The notion of *trust* is widely recognised as a means of assessing the perceived risk in interactions arising from uncertainty about how others will behave (Castelfranchi and Falcone, 1998; Marsh, 1994a). Trust represents an agent's *subjective* estimate of how likely another agent is to fulfil its cooperative commitments. We base our model of trust upon the formalism proposed by Marsh (1994a) and the work of Gambetta (1988), and define the trust T_α in an agent α to be a value from the interval between 0 and 1. Values approaching 0 represent complete distrust and those approaching 1 represent blind trust. Trust values are determined by an agent's previous experience, and are updated after each interaction. The numbers themselves primarily represent comparative values internal to an agent's individual representation, and are meaningful only in the context of the agent's experience. These values represent the degree of trust *subjectively* ascribed to another based on *individual* experience and disposition. Different agents may ascribe different trust values to a third party based on their own individual interactions. Trust values are not directly numerically comparable across agents because they are subjective and based on individual experiences and dispositions. A specific trust value has a different meaning for different agents. Trust is an individual assessment of another based on experience, and should not be confused with the notion of reputation, which represents an assessment of another based on both individual experience *and information obtained from others*. We discuss the notion of reputation in Section 5.

Trust values are associated with a measure of *confidence*, and as experience is gained confidence increases. However, it is important to consider the recency of experience. 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 toward 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.

An agent has a trust model of each other agent with whom it has previously interacted or has acquired knowledge. If there have been no previous interactions and there is no acquired (or pre-built) knowledge then there is no corresponding agent model and the default trust value T_{initial} is used when assessing its trustworthiness. Otherwise, for each other agent an individual will have a trust model representing the capabilities that it is believed to possess, and the trust ascribed to it. We represent a trust model as a tuple (id, C, t) where id is the identifier of the agent being modelled, C is a set corresponding to its believed capabilities, and t is the ascribed trust.

2.2 Inferring trust

Trust values are inferred according to an agent's *disposition*: optimists infer high values, while pessimists infer low values (Marsh, 1994b). After a successful interaction, optimists increase their trust more than pessimists, and conversely, after unsuccessful interactions 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. The range of this disposition is a continuum between blind optimism and blind pessimism, where a blind optimist only ever increases its trust of others, and a blind pessimist only ever decreases its trust. At the extremes of this continuum trust ceases to be a useful concept, since eventually blind optimists will place complete trust in all agents and blind pessimists will have complete distrust of all others.

The trust disposition of an agent is described by three characteristics:

(1) the initial trust it ascribes given a lack of other information, T_{initial},

(2) the function used to update trust after a successful interaction, $\text{update}_{\text{success}}$, and

(3) the function used to update trust after an unsuccessful interaction, $\text{update}_{\text{fail}}$.

The functions for updating trust are simple heuristics, and there is no standard definition, rather, it is the responsibility of the system designer to choose appropriate heuristics. We take a simple approach by defining the update function for a successful interaction as

$$\text{update}_{\text{success}}(T) = T + ((1 - T) \times d_{\text{success}})$$

and the update function for an unsuccessful interaction as

$$\text{update}_{\text{fail}}(T) = T \times d_{\text{fail}}$$

where d_{success} and d_{fail} represent the agent's disposition for increasing and decreasing trust, respectively, such that $d_{\text{success}}, d_{\text{fail}} \in [0, 1]$.

3. Clans

Motivation and trust are the fundamental components that lead to cooperation: motivations give rise to the wish to cooperate, and trust guides the decision about who to cooperate with. Cooperation is more than simultaneous actions and individual intentions; agents must be *committed* to the activity of cooperation itself (Bratman, 1992; Levesque *et al.*, 1990) and must abide by an appropriate set of *conventions* (Wooldridge and Jennings, 1999) specifying when and how cooperation can be abandoned. Where such commitments and conventions are adopted, we say that agents have formed a *cooperative intention*. Three basic stages (Figure 3) are involved in the formation and execution of a cooperative intention: plan selection, intention adoption, and group action, as follows.

(1) *Plan selection*: Motivations cause the generation of goals, which must be adopted as intentions by selecting plans and committing to their execution. When selecting plans agents consider both the likelihood of finding other agents to assist and their trustworthiness. By combining plan cost with an estimate of the risk associated with the potential cooperative partners, agents can balance their desire to minimise cost and risk (Griffiths and Luck, 1999).

(2) *Intention adoption*: If the selected plan requires cooperation then the agent must solicit assistance. The initiating agent annotates each contribution in the plan

Figure 3. Stages of cooperation

with the identifier of the agent considered best able to perform it, based on their believed capabilities and trustworthiness, and requests their assistance (Griffiths *et al.*, 2003). Requests are evaluated, and responses sent, according to agents' motivations and intentions. A cooperative intention is formed if sufficient agents agree to assist.

(3) *Group action*: Once a cooperative intention has been formed the plan is executed. On successful completion, commitments are dissolved and cooperation is finished. If execution fails, the agent that detects the failure informs the others in accordance with the conventions, and again commitments are dissolved. In both cases, agents update the trust values ascribed to others involved in cooperation according to the trust update functions described above.

Since agents cooperate in pursuit of a specific goal this is a short-term approach to cooperation, and there are four primary problems that arise. First, since agents are autonomous and driven by their individual motivations, missed opportunities for cooperation can occur. Secondly, where the environment contains a large number of agents, the overhead of establishing and maintaining cooperation can lead to scalability problems. Thirdly, the approach to cooperation described above assumes that agents have sufficient information about others' capabilities (and trustworthiness) to establish cooperation. Finally, in a dynamic environment the intensities of motivations can fluctuate, which can give a lack of robustness to cooperation.

Clans provide a means to address these problems, along with some of the limitations of existing approaches to coalition formation. In particular existing approaches to coalition formation tend to focus on specific tasks rather than taking a long-term view, and do not consider the trustworthiness of participants (Shehory and Kraus, 1998; Tambe, 1997). The congregations model avoids taking a short-term view, but again does not consider the trustworthiness of the agents involved (Brooks *et al.*, 2000; Brooks and Durfee, 2003).

We view clan formation as a *self-interested* activity – an agent attempts to form a clan for its own benefit, and not in response to any external influence. Clan formation is driven by an agent's motivations and guided by its trust of others. To determine when to form a clan, an agent must assess the extent to which the issues identified above are affecting its performance. The following gives a skeletal algorithm outlining the decision process:

function ASSESS-WHEN-TO-FORM-CLAN **returns** boolean
  local: *missed-opportunities* ← false
    scalability ← false
    lack-of-information ← false
    high-failure-rate ← false
  if ((*request-failure-rate* > *request-failure-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

In the remainder of this section we describe its component steps.

3.1 Missed opportunities

Motivations guide all aspects of an agent's behaviour, including its response to requests for cooperation. As described in the previous section, the intensities of motivations fluctuate in response to changes in the environment, and it is the current intensities that determine whether an agent *desires* to cooperate. A response to a request is determined by a combination of this desire to cooperate, whether it *can* cooperate (determined by its capabilities and intentions), and whether the *risk* from cooperation is acceptable (determined by trust). Where the intensities of agents' motivations are out of step in time, missed opportunities for cooperation can occur, since agents' desires to cooperate may also be out of step. These short-term fluctuations can lead to failures to establish cooperation that would be beneficial long-term. Therefore, an agent needs some means of assessing the extent to which it is missing such opportunities. Since motivations are private and internal to individuals, an agent cannot inspect others' motivations. Instead it must consider the requests for assistance received from others that it has declined. If few requests have been declined then there are, at most, few missed opportunities (from this agent's perspective). Alternatively, if there are many declined requests then there *may* be many missed opportunities; each missed opportunity leads to a declined request, although clearly requests may be declined for other reasons.

When an agent experiences a high failure rate in establishing cooperation (above the *request-failure-threshold*) due to other agents declining its requests, it should inspect all previous incoming requests within its memory limit. These previously rejected requests are filtered so that only those that are similar to the current plan remain. If the *current* motivational value of previously these filtered requests 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 reduce the likelihood of detecting simple one-off failures, an agent might consider the extent of fluctuations in motivation intensities, or the motivational value of the request over previous iterations. The best heuristic to use is dependent on the characteristics of the domain in which agents are situated, and this can only be verified empirically.

3.2 Scalability

To initiate cooperation, an agent typically must consider the suitability and trustworthiness of all other agents that it knows about. There is a cost to identifying and communicating with these agents, and the process of finding cooperative partners reduces the time spent in pursuit of goals. Furthermore, no direct motivational benefit is gained from identifying and communicating with others, only from the cooperative process itself. Thus, not only is there a computational and time cost in establishing cooperation, but the time the agent can spend in pursuit of its motivations is reduced. The number of agents modelled by an individual and the frequency of cooperation gives an indication of the scale of the problem. If cooperation is rare or there are few other agents modelled, then the impact is much less than if each plan requires cooperation and there are many agents to be considered. 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 filter out those that are unlikely to be relevant. In particular, we can measure the proportion of *applicable* plans that are cooperative in the last n reasoning cycles, where n is the agent's memory length. If the proportion that are cooperative exceeds the *scalability-threshold* then the agent should attempt to form a clan.

3.3 Lack of information

For an agent to successfully establish cooperation, it must know of trusted agents that have suitable capabilities. If there is insufficient knowledge of others' trustworthiness or capabilities then it may not be possible to establish cooperation. Recall from Section 2 that agents maintain a measure of confidence in their trust models, depending on the extent of the experiences that have formed them. If an agent does not have sufficient confidence in its models then clan membership may be beneficial, since clan members share information. Furthermore, if many agents are distrusted, then again clan membership may be beneficial. However, there is a lower bound on confidence and trust below which it is not appropriate form a clan, due to a lack of confidence and trust in the potential members. In particular, it is only practical 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 and if there are many untrusted agents (the average trust is below the *trust-threshold*) or agents whose models have low confidence (the average confidence is below the *confidence-threshold*), it should attempt to form a clan, provided there is a subset of confidently trusted agents with whom to do so.

3.4 High failure rate

With changes in the environment, motivation intensities fluctuate and can lead to failure in cooperation, since agents' commitments may change. Clan membership strengthens commitments to cooperation and may help reduce high failure rates. As we describe in Section 5, membership of a clan provides a mechanism for an agent to obtain motivational value through acting in what may otherwise appear to be a semi-benevolent manner, i.e. clans provide a means for an agent to gain individual benefit from assisting others. Thus, if an agent is experiencing execution failures above the *failure-threshold*, then it should attempt to form a clan.

4. Forming a clan

Based on its assessment of missed opportunities, scalability, lack of information, and failure rate, an agent determines whether it should form a clan. If clan formation is considered necessary then it should try to form a clan with an appropriate set of agents, namely, those who are trusted and likely to be relevant to any future cooperative activity. Trust determines whether it is practical to form a clan, since if an agent has a low trust of others or low confidence in its trust models, then it should not form a clan. If, however, it has adequate trust in others (above the *trust-threshold* with confidence greater than the *confidence-threshold*), then it can attempt to form a clan. In order to target its requests toward appropriate agents it should estimate its future goals and attempt to form a clan with those agents whose assistance is likely to be required. In a dynamic environment 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 future goals that might be generated. The set of actions for which assistance may be required, is obtained from these goals by considering the possible plans for them. Based on these actions the most trusted agents who are believed to have suitable capabilities are selected.

Ideally, all agents whose assistance is requested would be clan members, since this increases the likelihood of them agreeing to cooperate and keeping their commitments. However, large clan sizes have a disadvantage in terms of computational overhead and because there are more agents to whom assistance is inclined to be offered (at a potential cost of restricting an agent's other activities). Consequently, there is a *preferred* clan size which balances the conflicting desires for all future requested agents to be clan members, against the computational cost and the restrictiveness of acting on behalf of other clans members. We take a simple approach to assessing this preferred 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 extract the average number of actions for which assistance is required, using this to estimate the preferred size. As not all agents to whom requests are sent will join the clan, we add a degree of redundancy to this preferred size.

Based on this preferred clan size, the set of most trusted agents who are believed to have relevant capabilities are sent a request to form a clan. These agents must then determine whether to accept the request. Typically, although clan membership may be beneficial, an agent's assessment of whether to form a clan (based on the factors described in Section 3) may not indicate this, i.e. although the clan formation factors may be relevant, they may not be sufficient for the agent to accept the request. The requesting agent must, therefore, give some additional 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 to join a clan should include an indication of the expected future activities of the clan. This is determined by considering the most active motivations, and extracting the most frequently generated goals. This set of goals represents the "general purpose" of the clan and corresponds to the activities clan members are likely to be asked to perform by the initiating agent.

Although this involves revealing essentially private information, we argue that the motivational benefit that the agent will (hopefully) gain from forming the clan, justifies giving this information.

If sufficient agents agree to form a clan (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 that did accept are informed of the failure to obtain sufficient positive responses and clan formation abandoned. The following outlines the clan formation process:

```
function FORM-CLAN returns boolean
  input: redundancy, timeout
  local: current-plans ← { }
    min-size ← 0
    preferred-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)
  preferred-size ← min-size × redundancy
  goals-to-communicate ← EXTRACT-GOALS(active-motivations)
  if (#(target-agents) > preferred-size) then target-agents ← HEAD
(target-agents, preferred-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
```

The first part of the algorithm is concerned with determining who to invite to join a clan, and requesting that they join. The latter part of the algorithm shows how the responses are processed.

To determine whether to agree to join a clan those agents that receive requests must consider both the trustworthiness the requesting agent and the motivational value of joining. If the trust of the requesting agent is below the minimum trust threshold, or is not *confidently* trusted, then the request is simply declined. If the requesting agent is sufficiently trusted then the criteria described in Section 3 are considered to give an indication of how beneficial clan membership would be *in general*. If this general assessment indicates that the agent desires to form a clan, then the request is accepted.

Otherwise, 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, i.e. the general motivational value is considered without reference to the current motivation intensities. If this value exceeds a threshold then the agent agrees to form a clan. The outline of the process of assessing requests for clan formation is:

```
function PROCESS-FORMATION-REQUEST returns response
  input: requester, request-goals
  local: motivational-value ← 0
  if (TRUST(agent-models, requester) < 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) then return accept else return decline
end
```

5. Reasoning in a clan

Clan membership influences three main aspects of behaviour: information sharing, commitment to cooperation, and scalability. In the first case, a clan member that requires information on others capabilities or trustworthiness, can request information from other clan members. In the second case, clan members are more likely both to cooperate and to fulfil their commitments, due to increased motivational value of cooperation. In order to ascribe motivational value to clan membership, and to ensure that agents remain self-interested, we introduce a *kinship motivation* to all agents. Kinship is mitigated by offering assistance to other clan members, and its intensity is determined by the proportion of goals that require cooperation, and the extent and quality of an agent's trust models. In the final case, scalability is increased by reducing the search cost of finding cooperative partners by simply searching through the members of the clan.

5.1 Sharing information: reputation

When an agent requires information about the capabilities and trustworthiness of others, it can ask other clan members. In particular, when faced with a plan containing actions for which no confidently trusted and capable agents are known, it can ask trusted members of its clan. Note that although clan members will have been trusted at the time of clan formation, some may have come to be distrusted over time, but not so much as to justify leaving the clan. Therefore, the trustworthiness of clan members must still be checked when interacting with them. Clan members gain motivational value, via the *kinship* motivation, from sharing information about other agents' capabilities and trustworthiness. The motivational value received from giving such information is determined by the intensity of the kinship motivation. If the intensity is above its associated threshold, then an agent should share information, otherwise insufficient benefit is received to justify offering information. Additionally, information should only be shared with trusted agents, and before giving information an agent should check that the requester is trusted. This mechanism allows agents to discover other trusted agents outside the clan to assist them. The process of requesting information from other clans members is outlined thus:

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

Subjectivity is the primary problem in sharing trust information, since trust values are internal to agents and depend on disposition and experience; they are not directly comparable numerically across agents. Some researchers take the approach of eliminating small subjective differences between agents by using a stratification of trust, dividing the numerical range into subranges (Marsh, 1994a; Abdul-Rahman and Hailes, 2000). Stratification removes subjective differences between agents provided those differences are within the same subrange. However, if trust values differ across subranges then stratification is counter-productive and accentuates differences. Furthermore, stratification of the numerical range leads to a loss of sensitivity and accuracy; it becomes impossible to distinguish between values that are in the same subrange. Stratification only addresses subjectivity if the differences in trust values between agents are small. Agents' dispositions and experiences must be such that if two agents ascribe a trust value in the "highly-trusted" subrange, they infer the same *meaning* for this value. However, as discussed in Section 2, a consequence of agents having individual dispositions is that, in general, two different agents will not infer the same meaning from a given numerical value. 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 take a more straightforward approach in which agents simply communicate numerical values, knowing that these values are not directly comparable across agents. The following outlines the process of sharing information with another clan member:

```
function PROVIDE-INFORMATION
  input: problem-actions, requester
  local: response ← { }
    agent ← null
  if ((INTENSITY(kinship) > THRESHOLD(kinship)) and
    (TRUST(requester, agent-models) > trust-threshold)
    and (requester ∈ CLAN-MEMBERS(agent-models))) then
    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
```

When sharing trust information, we adopt two key constraints, as proposed by Marsh (1994a): if agent α_1 obtains information about α_3 from α_2 then

(1) α_1 does not trust α_3 more than α_2 trusts α_3, and

(2) α_1 does not trust α_3 more than it trusts α_2.

Thus, any trust information obtained is moderated by the trust ascribed to the provider. Since the resulting information about α_3 incorporates another agent's subjective view, the result is an assessment of α_3's *reputation*. Recall that *trust* refers to an individual's assessment of another, while *reputation* refers to an assessment based on others' trust values, i.e. trust is an individual notion and reputation is a social notion. Since reputation also includes subjective elements in terms of the trust of the information providers, different agents within a clan are likely to arrive at different reputation assessments for a given agent, although in practice these differences are typically small.

To determine the reputation of an agent, based on information provided by a set of clan members, we take the average value where each component is moderated by the trust ascribed to the provider. Thus, the reputation from the perspective of agent α_x of agent α_y, based on information provided by clan members $\alpha_1, \alpha_2, \ldots, \alpha_n$ is determined as follows

$$R_{xy} = \frac{\sum_{i=1}^{n} T_{\alpha_x \alpha_i} \times T_{\alpha_i \alpha_y}}{n}$$

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

It should be noted that other notions of reputation have been proposed elsewhere. However, our model differs from other approaches in that reputation is determined directly from individual trust and agents' dispositions. The REGRET system, for example, considers reputation in an online marketplace scenario, where agents record "impressions" of others after interactions (Sabater and Sierra, 2001). Reputation is determined by combining impressions and individual experience. This approach is similar to ours in that reputation is a combination of an agent's own experience, and that of others. However, there is no explicit representation of individual trust and, although related to trust, the impressions used in REGRET do not represent an individual's assessment of risk. Rather, they are a subjective evaluation made by an agent on the outcome of an interaction. Mui *et al.* (2002, 2003) propose a mechanism for assessing trust and reputation by statistical estimates of cooperation in a Prisoner's Dilemma interaction, where agents either *cooperate* or *defect*. Reputation measures the likelihood that an agent reciprocates another's actions, and will cooperate in the Prisoner's Dilemma game. Information is propagated via embedded social networks in which agents are assumed to reveal the trust and reputation information they ascribe to others. Our approach differs since reputation and trust are subjective estimates based on experience, not on probabilities. Furthermore, we do not assume that the propagation of information is automatic, since we require there to be motivational justification for information sharing.

5.2 Cooperation through kinship

The kinship motivation serves to increase the likelihood of clan members cooperating, and of fulfilling cooperative commitments, by providing motivational value from cooperation. Kinship functions like any other motivation in guiding behaviour; 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 reasoning cycle (as described in the mBDI reasoning cycle above) to incorporate this inclination to assist clan members. At a philosophical level, the kinship motivation can perhaps be seen to undermine the self-interested nature of agents. However, recall that agents choose to join a clan for specific reasons that are undeniably self-interested. Furthermore, kinship 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. Decisions about cooperation continue to be driven by *all* of an agent's motivations, and kinship is just one factor that contributes to a decision.

5.3 Improving scalability

In general, when determining agents to ask for assistance all other known agents are considered. However, as described in Section 3 this can lead to scalability problems in systems that contain large numbers of agents. Where an agent joins 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 rather than considering all known agents. However, the cost of this is that the agent will overlook the most appropriate agents if they are not in the clan, even if they are trusted and are known to have appropriate capabilities. Given this disadvantage, agents should only restrict their search to clan members when it is necessary to do so. In particular, if the scalability criterion for clan formation, described in Section 3, is applicable then the agent should initially only consider other clan members. When faced with a plan that requires cooperation the agent should use the standard process of attempting to form a cooperative intention but be restricted to considering agent models corresponding to clan members, meaning that only clan members are asked for assistance. If this fails, then the standard cooperative intention formation procedure is undertaken, where all known agents are considered.

6. Leaving a clan

Clan membership has a cost, in terms of computational overhead and because kinship may lead an agent to assist another clan member, rather than act as it would otherwise. It is not possible to directly assess the costs and benefits of clan membership, 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. Since others' motivations cannot be inspected it is not possible for an agent to assess whether it is getting something in return for its clan membership, i.e. the extent to which others' kinship motivations are affecting their behaviour. From the agent's viewpoint, however, this does not matter – provided that it is successful in gaining cooperation then clan membership is considered beneficial. (Note that even from an external viewpoint there are many subtle benefits to clan membership that are difficult to assess, such as becoming more trusted by potential partners due to being seemingly "exploited".)

Provided that the clan is operating effectively there will be sufficient reciprocal action for agents to receive net motivational benefit overall. Indeed, this is one of the reasons for forming a clan: to address short-term fluctuations in motivations leading to missed cooperation opportunities. However, over time agents' active motivations may change and the motivational benefit gained from membership of a specific clan will decrease, and eventually agents may receive insufficient benefit to justify continued membership. If an agent's active motivations change such that it no longer receives sufficient benefit from the clan, then it should withdraw its membership by notifying the other members. Agents should also withdraw their membership if they come to distrust the other members.

Since it is not possible to accurately consider whether the benefits of clan membership outweigh the costs, we take a simple approach to assessing whether to leave a clan by assessing its relevance and its influence on cooperative success. In particular, we consider the proportion of recently adopted plans for which cooperation was required, and if this proportion is below a minimum *relevance-threshold* then the clan is considered no longer relevant and agent should leave. 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 thus:

```
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
```

Each individual makes its own decision about whether to stay in a clan or leave and there is no formal clan dissolution process. As the number of agents that remain in a clan decreases, the benefits obtained from clan membership to the members is also likely to decrease. Eventually, a clan will contain a single agent at which point the clan ceases to exist.

7. Joining existing clans

Although we have described how agents can create clans, to be flexible agents must be able to join existing clans as well as creating new ones. The primary problem in enabling agents to join existing clans is providing a mechanism for agents to *discover* a suitable clan to join. In our scenario there is no centralised control or repository, and so a directory of existing clans is inappropriate. Indeed, if such a directory existed there would be no clear motivational value for agents to provide information about their clan membership to be interrogated by other, potentially distrusted, agents.

Our approach is to provide two means for an agent to discover existing clans: by invitation from a clan member, or in response to a request for a member of an existing clan to join a new clan. The first case is a straightforward extension of the criteria for determining when to form a clan. Suppose an agent believes that it should form a clan (using the skeletal algorithm outlining the decision process), but on assessment of who to request discovers that many of the desired members of the new clan are already members of an existing clan. In this case, rather than forming a new clan, the agent can instead invite those agents who are not already members to join the existing clan. Since agents are self-interested, the inviting agent does not ask "permission" from the existing clan members, rather it simply informs them about any positive responses from newly invited agents and those agents update their knowledge of the clan accordingly (and continue to monitor the relevance and effectiveness of the new clan).

Our second alternative occurs when 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 can either respond in the standard manner, or can invite the proposed members of the new clan to join the existing clan. If the goals communicated by the requester are similar to the goals that caused the formation of the existing clan, then an invitation to join the existing clan is appropriate, provided that all of the agents concerned are suitably trusted. Such invitations to join an existing clan, are processed in the same manner as a standard clan formation request.

8. Conclusion

In this paper, we have described how clans can be used to address some of the limitations of existing approaches to cooperation. In particular, the problems of missed opportunities for cooperation, scalability, a lack of information, and high failure rates. We have described how agents can assess when to form a clan, how they should act within a clan, and the conditions under which they should leave. Clans are a mechanism for agents to improve their individual performance through cooperation without compromising their autonomy. A clan can be thought of as 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 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.

Our model of clans has been validated by an initial simulation of a distributed computing scenario comprising a set of agents, each with individually defined capabilities and motivations, situated in a dynamic and unpredictable environment. The capabilities define what an agent can achieve alone, and the motivations give rise to agents' goals according to the current state of the environment. We undertook several simulations, varying the significance agents placed in their kinship motivations by changing the intensity and mitigation functions. In comparison with a control configuration where agents did not form clans, the introduction of clans significantly reduced the number of failed interactions (where agents rescinded commitments to cooperation due to changes in motivation intensities). As more importance was placed in the kinship motivation, less failures were experienced. In general, the number of successful interactions was increased with the introduction of clans. Owing to the computational overheads associated with clans, the benefits obtained with low kinship importance were negligible. An increase in kinship importance brought a corresponding increase in successful interactions. However, as kinship became more important, other motivations were overridden and, although there were increased successful interactions, agents tended to focus on assisting in achieving others goals rather than achieving their own goals.

There are three key areas of ongoing work. First, 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 alternative to the stratification approach that we rejected in Section 5, by allowing agents to 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. This would allow us to have the benefits of stratification in terms of simplicity, while avoiding the associated problems. Finally, although we have undertaken limited experimentation of our approach, with favourable initial results, ongoing work involves performing more extensive evaluation. In particular, we intend to investigate the cost of clan membership on autonomy in terms of agents assisting others rather than achieving their own goals.

References

Abdul-Rahman, A. and Hailes, S. (2000), "Supporting trust in virtual communities", *Proceedings of the Hawaii International Conference on System Sciences*, p. 33.

Bratman, M.E. (1992), "Shared cooperative activity", *Philosophical Review*, Vol. 101 No. 2, pp. 327-41.

Bratman, M.E., Israel, D. and Pollack, M. (1988), "Plans and resource-bounded practical reasoning", *Computational Intelligence*, Vol. 4, pp. 349-55.

Breban, S. and Vassileva, J. (2001), "Long-term coalitions for the electronic marketplace", in Spencer, B. (Ed.), *Proceedings of the E-Commerce Applications Workshop, Canadian AI Conference*.

Brooks, C. and Durfee, E. (2002), "Congregating and market formation", *Proceedings of the First International Joint Conference on Autonomous Agents in Multi-Agent Systems (AAMAS-02)*, ACM Press, Bologna, pp. 96-103.

Brooks, C. and Durfee, E. (2003), "Congregation formation in multiagent systems", *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 7 Nos 1/2, pp. 145-70.

Brooks, C., Durfee, E. and Armstrong, A. (2000), "An introduction to congregating in multiagent systems", in Durfee, E. (Ed.), *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, pp. 79-86.

Castelfranchi, C. (1995), "Guarantees for autonomy in cognitive agent architecture", in Wooldridge, M.J. and Jennings, N.R. (Eds), *Intelligent Agents: Proceedings of the First International Workshop on Agent Theories, Architectures and Languages (ATAL-94)*, Springer, Berlin, pp. 56-70.

Castelfranchi, C. and Falcone, R. (1998), "Principles of trust for MAS: cognitive anatomy, social importance, and quantification", *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, Paris, pp. 72-9.

Gambetta, D. (1988), "Can we trust trust?", in Gambetta, D. (Ed.), *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, Oxford, pp. 213-37.

Griffiths, N. (2000), "Motivated cooperation in autonomous agents", PhD thesis, University of Warwick.

Griffiths, N. (2003), "Supporting cooperation through clans", *Cybernetic Intelligence – Challenges and Advances, Proceedings IEEE Systems, Man and Cybernetics, 2nd UK&RI Chapter Conference*.

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

Griffiths, N. and Luck, M. (2003), "Coalition formation through motivation and trust", *Proceedings of the Second International Conference on Autonomous Agents and Multi-agent Systems (AAMAS-03)*, pp. 17-24.

Griffiths, N., Luck, M. and d'Inverno, M. (2003), "Annotating cooperative plans with trusted agents", in Falcone, R., Barber, S., Korba, L. and Singh, M. (Eds), *Trust, Reputation, and Security: Theory and Practise*, Springer, Berlin, pp. 87-107.

Kinny, D., Ljungberg, M., Rao, A., Sonenberg, E., Tidhar, G. and Werner, E. (1992), "Planned team activity", *Proceedings of the Forth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-92)*, pp. 227-56.

Klusch, M. and Shehory, O. (1996), "Coalition formation among rational information agents", in Van de Velde, W. and Perram, J.W. (Eds), *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96)*, pp. 204-17.

Levesque, H.J., Cohen, P.R. and Nunes, J.H.T. (1990), "On acting together", *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, Boston, MA, pp. 94-9.

Luck, M. and d'Inverno, M. (1995), "A formal framework for agency and autonomy", *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, AAAI Press/The MIT Press, Menlo Park, CA/Cambridge, MA, pp. 254-60.

Marsh, S. (1994a), "Formalising trust as a computational concept", PhD thesis, University of Stirling.

Marsh, S. (1994b), "Optimism and pessimism in trust", *Proceedings of the Ibero-American Conference on Artificial Intelligence (IBERAMIA-94)*.

Mui, L., Mohtashemi, M. and Halberstadt, A. (2002), "A computational model of trust and reputation", *Proceedings of the 35th Hawaii International Conference on System Science*.

Mui, L., Mohtashemi, M. and Halberstadt, A. (2003), "Evaluating reputation in multi-agents systems", in Falcone, R., Barber, S., Korba, L. and Singh, M. (Eds), *Trust, Reputation, and Security: Theory and Practise*, Springer, Berlin, pp. 87-107.

Norman, T.J. (1996), "Motivation-based direction of planning attention in agents with goal autonomy", PhD thesis, University of London.

Sabater, J. and Sierra, C. (2001), "REGRET: A reputation model for gregarious societies", paper presented at Fourth Workshop on Deception Fraud and Trust in Agent Societies, pp. 61-70.

Shehory, O. and Kraus, S. (1995), "Task allocation via coalition formation among autonomous agents", *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montréal, Québec, pp. 655-61.

Shehory, O. and Kraus, S. (1998), "Methods for task allocation vai agent coalition formation", *Artificial Intelligence*, Vol. 101, pp. 165-200.

Tambe, M. (1997), "Towards flexible teamwork", *Journal of Artificial Intelligence Research*, Vol. 7, pp. 83-124.

Wooldridge, M. and Jennings, N.R. (1999), "Cooperative problem-solving", *Journal of Logic and Computation*, Vol. 9 No. 4, pp. 563-92.