Task Delegation using Experience-Based Multi-Dimensional Trust

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

ABSTRACT

Cooperation among autonomous agents involves an inherent degree of uncertainty. Agents determine for themselves when to initiate cooperation or to assist others, when to rescind commitments, and how to conduct cooperative tasks. For example, an agent may delay the execution of a cooperative task, execute it to a reduced quality, or simply fail to complete it. In this paper, we describe how experience-based trust can be used to minimise the risk associated with cooperation. In particular we propose a mechanism, called multi-dimensional trust, which allows agents to model the trustworthiness of others according to various criteria. This trust information is combined with other factors to enable the selection of cooperative partners. Agents' preferences are represented by a set of factor weightings, which allow trust information to be tailored to the current cooperative priorities. We also describe the experimental validation of our proposed approach.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms, Experimentation, Reliability

Keywords

Trust, Cooperation, Task Delegation

1. INTRODUCTION

Cooperation and delegation are the defining characteristics of multi-agent systems. It is through cooperation and delegation that the agents in such systems are able to function effectively, since they typically lack the knowledge, capabilities or resources to achieve their objectives alone. To achieve flexibility and robustness in response to environmental change agents are typically given the autonomy to control their individual goals and behaviour. By enabling the individuals within a system to respond appropriately to change, we allow the system as a whole to exhibit similar flexibility and robustness. By definition, however, giving agents the autonomy to control their own behaviour implies that they control how they cooperate. In particular, agents can determine for themselves when to initiate cooperation or assist others, when to rescind cooperative commitments, and how to conduct cooperative tasks. Consequently, where a group of agents cooperate any one of them may change the nature of its cooperation, or even cease to cooperate, at any time. For example, an agent may choose to delay the execution of a task, execute it to a reduced quality, or simply fail to complete it. Such failures in cooperation are costly to the remaining cooperating agents since their goals may not be achieved, or not achieved as effectively (e.g. to a lower quality or after a deadline).

When entering into cooperation an agent is entering into an uncertain interaction in which there is a risk of failure (or reduced performance) due to the decisions and actions of another. To function effectively, agents need some mechanism for managing this risk; the notion of trust can provide this. In this paper we describe an approach, called *multi-dimensional trust* (MDT), in which agents model the trustworthiness of others along several dimensions. These trust dimensions are combined with other factors when delegating a task, to enable agents to select appropriate cooperative partners based on their current preferences. Our aim is to provide a trust mechanism that is rich enough to be useful, and yet have low enough overheads to be easily incorporated into a practical system. Several more complex approaches are possible for parts of our proposed framework. However, in this paper we focus on presenting the MDT approach, rather than on more sophisticated analysis of task execution or on the incorporation of the cognitive and belief-based aspects of trust.

Our proposed MDT mechanism is widely applicable, however in this paper we focus upon a particular domain. Specifically, we are concerned with a system comprising a set of autonomous self-interested agents, each having certain capabilities that are made available to others for a cost. Agents have a set of goals that are decomposed (through some unspecified planning process) into sequences of tasks. Agents' individual capabilities are such that the execution of these tasks typically requires cooperation with other agents. We use the term *delegation* to refer to the process of one agent performing a task on behalf of another. In this paper, we are not concerned with agent motivations, i.e. *why* agents cooperate, rather we simply assume that agents *are* motivated to perform tasks on behalf of others in return for imposing a charge. We also assume that autonomy and self-interest governs agents' cooperation, in terms of the success, cost, quality, and timeliness of the execution of delegated tasks. Thus, when delegating a task the choice of cooperative partner determines, at least in part, whether the task is successful and its associated cost, quality, and timeliness. We describe how trust dimensions are combined with other factors, such as advertised cost and quality, to enable agents to delegate tasks appropriately.

2. TRUST

The notion of trust is well recognised as a means of assessing the risk of cooperating with others [6, 9, 12]. Trust represents an agent's estimate of how likely another is to fulfil its commitments. When entering into cooperation an agent can use its trust of potential partners to evaluate the risk of failure. There are two main categories of trust: *experience-based* and *recommendation-based*. In the former, agents assess trust based solely on their own experience; in the latter, trust is based on information provided by others (typically in addition to individual experience). Experience-based trust is the simplest approach, where agents delegate tasks to others and update their trust models according to task outcomes. Recommendation-based trust requires agents to share information (based on their experiences) about how trustworthy another is perceived to be. Although this is a potentially powerful mechanism, there are a number of obstacles to its use. In particular, there is a need for an agreed trust semantics to enable information sharing, and for such information sharing to be motivated. Although in this paper we are not concerned with agent motivations *per se*, we do assume that agents are self-interested and so there must be self-interested justification for sharing trust information. Without considering motivations in detail, it is difficult to provide this justification, given that sharing positive information about a third party may jeopardise any ability for future cooperation (since others are more likely to delegate tasks to it, thereby reducing its availability). Several researchers are, however, investigating solutions to these problems to enable the use of recommendation-based trust [11, 15, 16, 18]. Our work is orthogonal to this, and we envisage experience-based and recommendation-based trust being combined in the future to provide a single trust mechanism. For the purposes of this paper, however, we are solely concerned with experience-based trust.

2.1 Multi-Dimensional Trust

Castelfranchi and Falcone view trust as encompassing beliefs about competence, disposition, dependence, and fulfilment [5, 6, 8]. For an agent α to be said to trust another β with respect to a particular goal g, then α must have some specific beliefs, namely that:

- β is useful for the achieving g and is able to provide the expected result (competence belief),
- β is not only capable, but will actually perform the required task (disposition belief),
- the involvement of β is essential, or least preferable, for the achievement of g (dependence belief), and
- due to β's involvement g will be achieved (fulfilment belief) [6].

We adopt this view of trust, however we take a multi-dimensional approach by decomposing trust to represent these beliefs according to the different dimensions of an interaction, such as the quality of a task or the cost imposed for executing it. Cooperative interactions are typically more than simple succeed or fail tasks. Agents delegate tasks with an expectation of successful performance to a given quality for some anticipated cost. In addition to possible task failure, tasks may succeed but be of lower than expected quality or at a higher than expected cost. Agents can model such characteristics as *dimensions of trust*. These dimensions can be considered as features of task execution, namely those characteristics that are important to the agent modelling trust. Each trust dimension encompasses the corresponding competence, disposition, dependence, and fulfilment beliefs. For example, if an agent is trusted to perform a task to a high quality, then it is believed to be capable of performing the task to a high quality (competence), actually doing so (disposition), being the preferred agent to do it (dependence), and being the means for task achievement (fulfilment). These beliefs are potentially (but not necessarily) independent from beliefs about other trust dimensions. Thus, beliefs about quality, cost, and the likelihood of task success are unrelated, and there is likely to be no correlation between these dimensions of trust. However, if cost is associated with the time to execute a task, then beliefs about cost and timeliness will be related, and there will of course be a correlation between these dimensions. The trust for each interaction is considered along each of these dimensions. This is similar to considering multiple goal elements in Castelfranchi and Falcone's approach, e.g. the goal of performing the task to a high quality, the goal of doing so on time, and so forth.

Our proposed multi-dimensional approach is related to, but distinct from, Marsh's approach of general and situational trust [12]. General trust gives an overall view based on previous interactions, while situational trust is finer grained and based on interactions in similar situations. In Marsh's model, when considering cooperation for a particular situation an agent determines a single trust value for another by using a weighted combination of the general trust placed in it, along with the situational trust for similar situations [14]. Our proposed MDT model also gives agents a finer grained model of others, but unlike Marsh's approach this granularity is according to the dimensions of trust (such as cost and quality), rather than for explicit situations. Our approach is based on general trust rather than situational trust, however it gives agents a finer grained model that a simple general trust value. We argue that our approach provides a useful (in terms of the practicality of implementation) refinement of a simple general trust model.

Maintaining MDT models is relatively simple: on delegating a task an agent has certain expectations according to the dimensions being used, and on receiving results the agent assesses whether its expectations were met. This avoids the main drawback of situational trust, namely, the computational overhead involved in identifying and maintaining trust values for specific similar situations. It is important to emphasise that MDT is complementary to situational trust and the two can be combined. For example, an agent might model the situational trustworthiness of others in different dimensions (e.g. to perform to a high quality in particular situations). In the remainder of this paper, however, we do not consider situational trust further.

The trust dimensions of quality and cost discussed above are merely illustrative, and agents can model trust along any number of dimensions according to their preferences and motivations. For the purposes of this paper, we model the trust of an agent α along the following dimensions:

- success (denoted T^s_α): the likelihood that α will successfully execute the task,
- cost (denoted T^c_α): the likelihood that the cost of α executing the task will be no more than expected,
- timeliness (denoted T^t_α): the likelihood that α will complete the task no later than expected, and
- quality (denoted T^q_α): the likelihood that the quality of results provided by α will meet expectations.

Each of these trust dimensions is updated after an interaction, and an agent is able to consider them all in delegating a task, as described in Section 4.

2.2 Representing Trust

We base our representational model of trust on Gambetta's theoretical work [9], Marsh's formalism [12], and our previous work [7, 10], and define the trust in an agent α, along a dimension d, to be a real number in the interval between 0 and 1: $T^d_\alpha \in [0, 1]$. The numbers merely represent comparative values, and have no strong semantic meaning in themselves. Values approaching 0 represent complete distrust, and those approaching 1 represent complete trust. There is an inverse relationship between trust and the perceived risk of an interaction: cooperating with a trusted agent has a low perceived risk of failure in the dimensions for which the agent is trusted, while there is a high risk associated with distrusted agents. Trust values represent the view of an individual agent, subjectively based on experience, and are not directly comparable across agents. Trust values are associated with a measure of confidence according to the breadth of experience on which they are based; as an agent gains experience its confidence increases.

Trust initially takes a value according to an agent's disposition on a continuum from optimistic to pessimistic, and is subsequently updated according to experience. Optimists ascribe high initial trust values (implying low perceived risk), and pessimists ascribe low values. Agents' dispositions also determine how trust is updated after interactions [13]. After interacting, optimists increase their trust more than pessimists for the dimensions in which the interaction met expectations and, conversely, pessimists decrease trust to a greater extent when expectations are not met. Since we are aiming to provide a general framework that can be easily incorporated into practical systems we must ensure that the overheads are kept to a minimum. For this reason, when updating trust values, agents only consider whether or not their expectations are met, rather than attempting to quantify the degree by which expectations are achieved (or otherwise). Although using the degree to which expectations are met would provide a more powerful mechanism, it requires more complex reasoning about the effect on trust of the extent to which expectations are met. Whilst this is relatively easy for a dimension such as cost, it is more difficult to quantify for dimensions such as timeliness and quality. Hence, we only consider simple update functions that distinguish binarily between achieved and unachieved expectations. Future work will investigate more complex approaches.

An agent's disposition comprises: the initial trust $T_{initial}$ ascribed to each trust dimension prior to interacting and functions for updating trust after successful and unsuccessful interactions, $update_{success}$ and $update_{fail}$ respectively. These functions are simple heuristics that apply to all trust dimensions, and there is no standard definition for them. Instead, it is the responsibility of the system designer to choose an appropriate heuristic. In this paper we use the following definitions to update the trust in agent α along dimension d:

$$update_{success}(T^d_\alpha) = T^d_\alpha + ((1 - T^d_\alpha) \times (\omega_s \times T^d_\alpha))$$

$$update_{fail}(T^d_\alpha) = T^d_\alpha - ((1 - T^d_\alpha) \times (\omega_f \times T^d_\alpha))$$

where ω_s and ω_f are weighting factors defined by the disposition. It is beyond the scope of this paper to discuss the impact of different update functions, however our experiments give similar results to those presented in Section 6 with different update functions of a similar form.

Over time trust values may become inaccurate and outdated if the experiences that gave rise to them are no longer relevant. Resources may change, and a resource that was trustworthy previously may no longer be so. To address this problem, we apply a decay function to converge each trust value to $T_{initial}$ in the lack of subsequent experience. Thus, unless reinforced by recent interactions, the positive effect of expectations being met reduces over time, as does the negative effect of failed expectations. The decay function for the trust in agent α along dimension d is defined as:

$$decay(T^d_\alpha) = T^d_\alpha - \frac{T^d_\alpha - T_{initial}}{\omega_d}$$

where the decay rate ω_d is defined by the disposition.

3. STRATIFIED TRUST FOR COMPARISONS

In our approach, trust in each dimension is represented as a numerical value, however some researchers note that the use of such values can introduce ambiguity since the semantics are hard to represent [1, 12]. One alternative, is to divide the trust continuum into labelled strata, and use these to represent trust values. Abdul-Rahman and Hailes, for example, take this approach and use four distinct trust strata ("very trustworthy", "trustworthy", "untrustworthy", and "very untrustworthy") that they argue provide a clear semantics [1]. The problem of defining the meaning of a numerical value, is avoided since "trustworthy" for one agent should correspond to "trustworthy" for another. However, these semantics are still subjective, and different agents may ascribe the same experiences to different strata; experiences that rate as highly trustworthy for one agent may rate as trustworthy for another. Furthermore, representing trust using strata gives a loss of sensitivity and accuracy, since comparisons become coarse grained with no way to distinguish between agents within a stratum. For this reason Marsh rejects the use of strata in favour of numerical values [12]. Our approach, to avoid loss of sensitivity and accuracy, is also to use numerical values to represent trust. Additionally, updating trust values is simple for a numeric representation, whilst stratified approaches often omit details of how agents determine trust strata from experience [1, 2].

The advantage of trust strata is that comparisons when selecting partners are simplified. In our approach, suppose that an agent must select between alternative partners with trust values 0.5 and 0.50001 for a particular trust dimension. The agent must either conclude that the numerical difference is insignificant and so the alternatives are equally trusted, or that there is a real difference in trust and the latter is more trustworthy. Although such small differences may arise from genuine trust variations, they may also arise from variations in the extent or recency of experience. There is, therefore, a risk of overfitting by drawing conclusions from trust values where differences arise from irrelevant artifacts of the data.

Using strata minimises overfitting, since there are no numerical values for consideration. Ideally, a trust model would have the sensitivity and accuracy of a numerical approach, combined with the comparison advantages of a stratified approach. To this end, we use a variable size stratifying of trust *at the time of trust comparisons*. Trust values are translated into strata immediately before comparison. The number of strata is not fixed, although typically an agent will use the same number of strata for each trust dimension and in each comparison. Fewer strata minimise the risk of overfitting but give the least precise comparison, while more strata retain precision, but at an increased risk of overfitting.

4. DELEGATING BY COMBINING TRUST DIMENSIONS

When choosing a cooperative partner, an agent must consider several factors, including the various dimensions of trust. An agent's preferences determine the emphasis given to each of these factors. For example, one agent may prefer to minimise the risk of failure and achieve the highest quality, while another may be concerned primarily with minimising cost. Some factors may have related trust dimensions, such as the expected quality of results, while others may not, such as the available communication bandwidth. Each of these factors, and associated trust values where appropriate, must be combined to determine which potential cooperative partner is the best choice according to the agent's preferences.

To select between agents we adopt a weighted product model for combining choice factors to obtain a single performance value for each agent [3, 17]. A weighted product model is a standard multi-criteria decision making technique; other more powerful approaches are available, but in accordance with our aim to ensure practical applicability by minimising the overheads we adopt this simple technique. Each factor is raised to the power equivalent to its relative weight according to the selecting agent's preferences. For each potential partner a performance value is calculated as:

$$PV(\alpha) = \prod_{i=1}^{n} (f_{\alpha_i})^{\mu_i}$$

where there are n factors and f_{α_i} is the value for agent α in terms of the i'th factor and μ_i is the weighting given to the i'th factor in the selecting agent's preferences. The values of the weightings μ_i defined by the selecting agent's preferences must be such that:

$$\sum_{i=1}^{n} \mu_i = 1$$

The best alternative is the agent α whose performance value $PV(\alpha)$ is greater than that of all other agents. Where several agents have equal performance values, one is selected arbitrarily.

Provided that the μ_i's sum to 1, individual weightings can take any value in the interval $[0 : 1]$. Thus, agents can select based on a single factor by giving that factor a weighting of 1. This flexibility is one of the key strengths of the MDT approach, since the trust information maintained by the agent is the same, regardless of its current preferences and factor weightings. An agent's preferences can be determined by its current goal, without needing additional trust modelling. For example, agents may give more weight to the likelihood of success for crucial goals, while for less important goals (or goals where there is time to re-delegate) the cheapest alternative might be preferred. A disadvantage of this approach is that weightings must be chosen judiciously. However, a developer has flexibility over these weights without affecting the underlying framework and trust values that are represented.

Factors such as quality can be used directly in calculating the performance value, provided that they are numerical and the agent wishes to maximise the value. Similarly, factors that should be minimised, such as cost, can be included by using the value

$$f_{\alpha_c} = max(\alpha_c \ldots \xi_c) + 1 - \alpha_c$$

where α_c represents the advertised cost from agent α, and $max(\alpha_c \ldots \xi_c)$ is the maximum advertised cost of all agents being considered, also denoted as max_c. (The addition of 1 ensures that for a maximal cost alternative, the factor still has a positive value.)

In order to include trust values they must first be stratified, as discussed above. Our approach is to divide the trust range into s equal strata such that each is given a value from 1 to s in order. Trust values are stratified by determining the value of the stratum they occupy. For a trust value t, its stratum is obtained by using:

$$\mathit{stratify}(t) = \lceil t \times s \rceil$$

For example, using 10 strata, a trust value of 0.35 is given a stratum value of $\lceil 0.35 \times 10 \rceil = 4$.

Recall that in this paper we are considering the trust dimensions of success (T^s_α), cost (T^c_α), timeliness (T^t_α), and quality (T^q_α), for an agent α. When delegating a task each of these dimensions should be considered, along with the advertised cost and quality of each alternative agent. Thus, an agent should calculate a performance value for each potential partner as:

$$\begin{aligned} PV(\alpha) \;=\; & (max_c + 1 - \alpha_c)^{\mu_c} \times {\alpha_q}^{\mu_q} \\ & \times \mathit{stratify}(T^s_\alpha)^{\mu_{ts}} \times \mathit{stratify}(T^c_\alpha)^{\mu_{tc}} \\ & \times \mathit{stratify}(T^t_\alpha)^{\mu_{tt}} \times \mathit{stratify}(T^q_\alpha)^{\mu_{tq}} \end{aligned}$$

where α_c and α_q are α's advertised cost and quality respectively, max_c is the maximum advertised cost of the agents being considered, μ_c and μ_q are the weightings given to advertised cost and quality, and μ_{ts}, μ_{tc}, μ_{tt}, μ_{tq} are the weightings for the trust dimensions of success, cost, timeliness, and quality respectively. Note that although there is clearly a relation between advertised cost and trust in the cost dimension, we consider them independently in the calculation of the performance value. This allows agents flexibility through the choice of weighting to determine which factors are of higher emphasis. For example, an agent can choose to emphasise trust rather than advertised values of cost and quality, i.e. it may be preferable for an agent to have a high degree of expectation about the actual cost and quality of task execution, regardless of whether other agents give better advertised values.

4.1 Example Performance Values

By way of example, suppose that an agent must choose between two alternatives, α and β, such that the factors being considered have the following values.

factor	α	β
advertised cost (units per second)	8	10
advertised quality (range 1 to 10)	9	8
trust (success dimension)	0.93	0.42
trust (cost dimension)	0.63	0.95
trust (timeliness dimension)	0.81	0.77
trust (quality dimension)	0.42	0.71

Suppose that the selecting agent uses the following factor weightings (i.e. each is considered equal with the exception of success, which is given a higher weighting).

μ_c	μ_q	μ_{ts}	μ_{tc}	μ_{tt}	μ_{tq}
0.16	0.16	0.2	0.16	0.16	0.16

The agent should calculate the performance value of each of the alternative partners. Thus, applying $PV()$ to agent α gives:

$$\begin{aligned} PV(\alpha) &= (10+1-8)^{0.16} \times 9^{0.16} \times \mathit{stratify}(0.93)^{0.2} \\ &\quad \times \mathit{stratify}(0.63)^{0.16} \times \mathit{stratify}(0.81)^{0.16} \\ &\quad \times \mathit{stratify}(0.42)^{0.16} \\ &= 3^{0.16} \times 9^{0.16} \times 10^{0.2} \times 7^{0.16} \\ &\quad \times 9^{0.16} \times 5^{0.16} \\ &= 6.741 \end{aligned}$$

Similarly, for agent β we get $PV(\beta) = 5.411$. Therefore, based on the given weightings, agent α is the alternative that best balances the factors considered.

To demonstrate how the factor weightings allow agents to balance their preferences, suppose that the following weights are used emphasising the quality and cost of results (in terms of advertised values and the perceived trustworthiness of potential partners to return those values).

μ_c	μ_q	μ_{ts}	μ_{tc}	μ_{tt}	μ_{tq}
0.15	0.15	0.05	0.3	0.05	0.3

In this case we get performance values of $PV(\alpha) = 5.965$ and $PV(\beta) = 6.116$. Thus, where greater emphasis is placed on quality and cost, agent β is considered the best alternative.

5. EXPERIMENTAL SCENARIO

Our proposed MDT model is generally applicable, and can be utilised in a variety of situations. In order to demonstrate its use, however, we use a Grid-based scenario. Grid computing aims to allow heterogeneous computational resources to be shared and utilised globally. These resources and their users can be viewed as autonomous agents, each having individual preferences and objectives. For the purposes of this paper, we consider a multi-agent system in which each agent represents a combined Grid resource and user. Thus, each agent has a set of capabilities that are available to others, and a set of tasks for which it needs to find cooperative partners. Each agent has individual characteristics that determine its success rate and the cost, quality and timeliness with which results are returned. Agents use MDT to determine which of the potential partners to delegate tasks to, according to their preferences. These preferences are defined by the factor weightings, and determine the emphasis given to success, cost, quality etc.

We have investigated this scenario using an extension to the GridSim simulation toolkit [4]. In GridSim, an agent's capabilities comprise a set of machines that in turn contain a set of processing elements (PEs). Each agent has certain capabilities, defined by its communication bandwidth and the configuration and processor architecture of its PEs. Our extension to GridSim gives agents additional characteristics defining their failure rate, cost and quality variations etc. To test the validity of our proposed MDT approach a series of experiments were performed, using a range of factor weightings in various environmental contexts (in terms of the reliability or otherwise of agents). Several sizes of system were experimented with, however the results presented below are for a system comprising 30 agents, each of which generates 500 random tasks to be completed with varying lengths, PE and bandwidth requirements, and priorities. For each experimental configuration we performed 10 runs, and the results shown below are the average values across those runs.

6. RESULTS AND EVALUATION

In this section we present results obtained from using MDT for task delegation, focusing upon three main aspects: failure rate, execution cost, and execution quality. Execution cost refers to the raw cost of performing a task as charged by the executing agent. The actual overall cost to the delegating agent may be much higher once the issues of quality and failures have been factored in. We begin by comparing the effectiveness of MDT using various strata sizes against cost-based, quality-based and random delegation. We also consider a simple general trust approach, where agents maintain a single trust value representing overall trustworthiness and use a strict numerical comparison [12]. Finally, we consider the impact of the factor weightings in various resource settings.

Figure 1 shows the failure rate of delegated tasks for varying strata sizes obtained in a "mixed" environment, i.e. there is a mix of reliable and unreliable agents[1]. The failure rates for the simple general trust and random delegation approaches are also shown, however for clarity the cost and quality delegation methods are omitted from the graph, since they give very high failure rates (3902 and 1313 respectively). An equal weighting is given to factor weightings (with a slight emphasis on success), as follows.

[1]It should be noted that the effect of the number of strata is broadly the same regardless of the mix of agents.

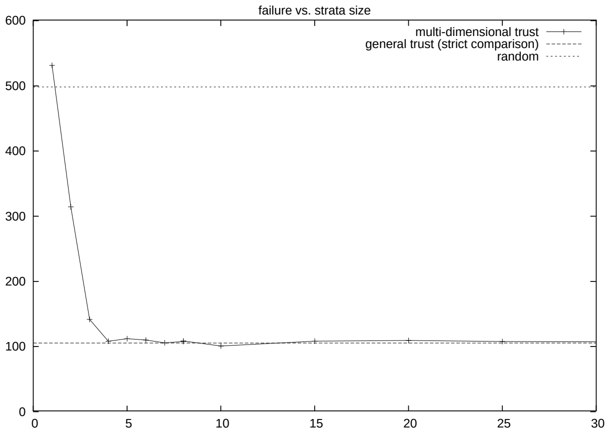

Figure 1: Failure rate for MDT versus strata.

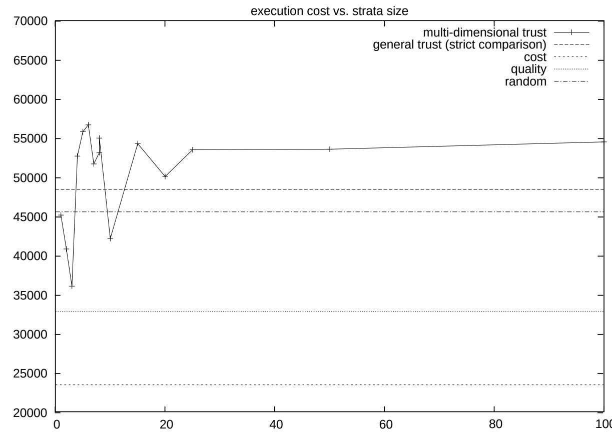

Figure 2: Execution cost for MDT versus strata.

μ_c	μ_q	μ_{ts}	μ_{tc}	μ_{tt}	μ_{tq}
0.16	0.16	0.2	0.16	0.16	0.16

It can be seen that, provided that more than 10 strata are used, the lowest failure rate is given jointly by MDT and general trust. Between 2 and 10 strata MDT improves from slightly worse than random to being equal to simple general trust. The random delegation method performs consistently poorly (with cost-based and quality-based being even worse).

The (raw) execution cost for the MDT approach is shown in Figure 2, along with the general trust, cost-based, quality-based, and random delegation methods. These results represent the cost only for *successfully executed* tasks, and so low cost does not necessarily correspond to the best approach (due to potentially high failure rates, meaning that several failed attempts occur before delegation to a successful agent). It can be seen that the cost-based approach gives the lowest execution cost, followed by quality-based, random, and simple general trust respectively. Quality-based is close to cost-based simply because in our environment, low cost tended to correlate with high advertised quality. The execution cost for the MDT approach fluctuates for less than 25 strata, and then stabilises as being the highest cost approach. For less than 25 strata it performs in the region of the general trust and random methods. If execution cost is the sole preference of the selecting agent, then a cost-based approach is best, but as discussed above this results in a very high failure rate (3902 in this example). It is important to note that the high execution cost for MDT shown in Figure 2 represents the raw cost of task execution, and when the costs incurred from failures and low quality executions are considered the overall cost is typically much improved over the alternative approaches. The fluctuations in execution cost for MDT with less than 25 strata are largely due to our environment (and remain present even if the number of runs is increased), since agents have a larger variance for execution cost than for failure, and so agents within the same trust strata may differ greatly according to execution cost. This fluctuation reduces significantly if the factor weightings favour cost, but this results in poor performance in terms of failure rate and quality.

One significant disadvantage of using general trust (with strict numerical comparison) for task delegation is that a very narrow set of agents is interacted with (due to small numerical differences being treated as significant). In the experiments illustrated in Figures 1 and 2, the general trust approach typically led to a single agent being interacted with, according to the task requirements. The MDT approach led to a wider set of agents being delegated to, where reduced strata gave wider sets (a single strata leads to all capable agents being cooperated with equally). For this reason, in addition to the difficulty of tailoring delegation to the agent's current preferences (e.g. quality, cost or success) we do not consider a general trust approach further.

Figure 3 shows the execution quality (again for successfully executed tasks) for the MDT approach, along with alternative approaches for control purposes. Contrary to what might be expected, the quality-based approach actually performs the worst. This is because it is based on *advertised* quality rather than actual or expected (purely based on trust) quality, and in our environment unreliable agents tended to advertise that they were high quality, but then perform poorly at execution time. The MDT approach (for more than 3 strata) gave the highest execution quality, and for above around 20 strata consistently gave 30% higher quality.

The final set of results that we present here illustrates the effect of factor weightings on the failure rate, execution cost and quality of execution, in different environments. We have performed experiments with several different weighting sets in a reliable, mixed, and unreliable environment. For the purposes of these experiments, reliable, mixed, and unreliable environments are defined by the proportion of honest (in terms of advertised cost and quality) and trustworthy (in terms of adhering to advertised information and successfully returning a timely result) agents. Our experiments are based on the following three configurations.

	Highly reliable	Reliable	Marginal	Unreliable	Highly unreliable	Very highly unreliable
reliable	60%	20%	10%	4%	3%	3%
mixed	20%	20%	20%	20%	10%	10%
unreliable	5%	10%	30%	20%	20%	15%

Due to space constraints we omit the details of the exact

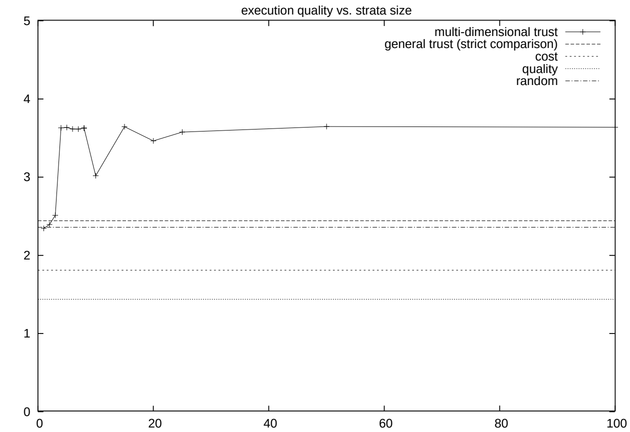

Figure 3: Execution quality for MDT versus strata.

definitions of these categories, but suffice to say that "highly reliable" etc. have their obvious meanings in terms of agent honesty and reliability (e.g. highly reliable corresponds to a failure rate of below 3%). (Our example data has a higher resolution at the unreliable end of the reliability scale, since it is the unreliable agents that have the most effect on the success of an interaction.) A range of factor weightings were investigated, including the following.

	μ_c	μ_q	μ_{ts}	μ_{tc}	μ_{tt}	μ_{tq}
Equal	0.16	0.16	0.2	0.16	0.16	0.16
Quality	0.0	0.5	0.0	0.0	0.0	0.5
Cost	0.5	0.0	0.0	0.5	0.0	0.0
Success	0.0	0.0	1.0	0.0	0.0	0.0
Success and Quality	0.03	0.3	0.31	0.03	0.03	0.3
Success and Cost	0.3	0.03	0.31	0.3	0.03	0.3

The effect of the factor weightings is shown in Figure 4, for the three environmental contexts introduced above. The upper graph (a) shows the failure rate, the middle graph (b) gives the execution cost, and the execution quality is shown in the lower graph (c). The weighting sets are indicated along the x-axis. It can be seen that the best results are obtained in a reliable environment, while agents in an unreliable context fare the worst. Furthermore, the effect of the factor weightings is reduced in reliable contexts. As the context becomes less reliable, the influence of the weightings increases. In an unreliable context, the *Success* weighting gives the least failure rate, whilst the *Quality* and *Cost* weightings give the highest. The highest execution quality is given by the *Quality* weighting, and the *Success* weighting performs noticeably worse. Similarly, for unreliable contexts the *Cost* weightings give the least cost. However, in a reliable context, the *Quality* and *Success* weightings actually give a lower execution cost. The perturbations that can be seen, for example in failure rate and execution cost using the *Quality* and *Success* weightings in an unreliable context are a result of the configuration of our example. In these cases, where agents focus on quality they tended to delegate to agents that had a high failure rate (due to the unreliable context), and similarly when focusing on success they tended to delegate to agents with a high variance in cost.

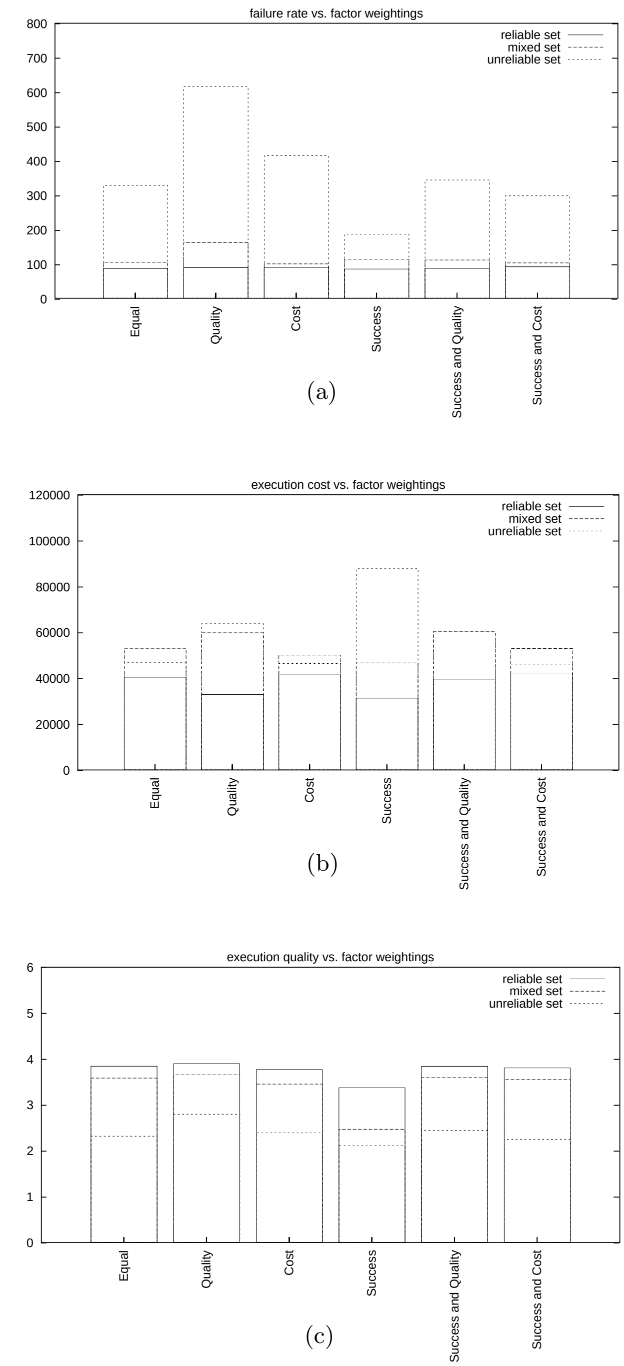

Figure 4: Failure rate (a), execution cost (b) and execution quality (c) for selected factor weightings in reliable, mixed, and unreliable contexts.

Overall, as shown in Figures 1 and 3, MDT offers clear improvements in failure rate and and execution quality compared to other approaches. Furthermore, as shown in Figure 4 MDT allows agents to balance execution cost against these and other factors.

7. CONCLUSIONS

In this paper we proposed the notion of multi-dimensional trust, and demonstrated its use for task delegation. MDT provides a mechanism for agents to model the various facets of trust, and combine these with other factors when selecting a cooperative partner. Factor weightings enable agents to combine decision factors according to their current preferences. We have illustrated MDT using the trust dimensions of success, cost, quality and timeliness, although many others are possible. The validity and effectiveness of the MDT approach has been demonstrated in a Grid environment.

There are several areas of ongoing work. The primary area is continued empirical evaluation, both of the effect of factor weightings and of alternative update functions. A second specific issue to be considered is to make explicit use of the confidence in trust values when delegating a task. The current model simply ensures that there is sufficient confidence in trust models. However, it may be beneficial to explicitly incorporate confidence into the calculation of performance values. More general ongoing work is concerned with investigating further approaches to combining trust factors, in particular the use of fuzzy logic and alternative multi-criteria decision making methods (e.g. those discussed in [17]). Future work also includes the incorporation of recommendation-based trust, and the development of a semantics for trust to allow sharing of trust information.

8. ACKNOWLEDGEMENTS

The author would like to thank the anonymous reviewers for their useful comments, which have identified several clear areas for ongoing consideration.

9. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000.

[2] F. Azzedin and M. Maheswaran. Integrating trust into Grid resource management systems. In *Proceedings of the International Conference on Parallel Processing*, pages 47–54, 2002.

[3] P. W. Bridgeman. *Dimensional Analysis*. Yale University Press, 1922.

[4] R. Buyya and M. Murshed. GridSim: A toolkit for the modelling and simulation of distributed resource management and scheduling for Grid computing. *Journal of Concurrency and Computation: Practice and Experience*, 14(13–15):1–32, 2002.

[5] C. Castelfranchi. Trust mediation in knowledge management and sharing. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proceedings of the Second International Conference on Trust Management (iTrust 2004)*, pages 304–318, 2004.

[6] 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.

[7] J. R. D. Dyson, N. Griffiths, H. N. Lim Choi Jeung, S. A. Jarvis, and G. R. Nudd. Trusting agents for Grid computing. In *Proceedings of the IEEE SMC 2004 International Conference on Systems, Man and Cybernetics*, pages 3187–3192, 2004.

[8] R. Falcone and C. Castelfranchi. Social trust: A cognitive approach. In C. Castelfranchi and Y.-H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer Academic Publishers, 2001.

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

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

[11] T. D. Huynh, N. R. Jennings, and S. Shadbolt. Developing an integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 7th International Workshop on Trust in Agent Societies*, pages 65–74, 2004.

[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] S. Marsh. Trust in distributed artificial intelligence. In C. Castelfranchi and E. Werner, editors, *Artificial Social Systems*, pages 94–112. Springer-Verlag, 1994.

[15] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. A computational trust model for multi-agent interactions based on confidence and reputation. In *Proceedings of the 6th International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69–75, 2003.

[16] J. Sabater and C. Sierra. REGRET: A reputation model for gregarious societies. In *Proceedings of the First International Joint Conference on Autonomous Agents in Multi-Agent Systems (AAMAS-02)*, pages 475–482, 2002.

[17] E. Triantaphyllou. *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Publishers, 2000.

[18] B. Yu and S. M. P. An evidential model of reputation management. In *Proceedings of the First International Joint Conference on Autonomous Agents in Multi-Agent Systems (AAMAS-02)*, pages 295–300, 2002.