# Supporting Peer-To-Peer Collaboration Through Trust

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

## Abstract

*Distributed systems generally require their component parts to interact cooperatively, in order for the system as a whole to function effectively. For any given activity, several alternative components may have the required capabilities. However, these components may be unreliable or dishonest, and are typically locally controlled. We view such systems as multi-agent systems, comprising autonomous agents that must cooperate for the system to be effective. In this paper we propose a mechanism, called MDT-R, for agents to delegate activities appropriately, using trust and the recommendations of their peers to meet preferences such as minimising risk and maximizing quality.*

**Keywords:** Agents, Trust, Cooperation, Peer-to-Peer.

## 1. Introduction

Complex distributed systems such as cooperative design processes, supply chains, and Grid computing can be viewed as multi-agent systems. Each component is considered to be an autonomous agent, and for the system to function effectively these agents must cooperate. The individual computational resources, knowledge bases, and human actors in a system are represented by agents with appropriate goals and preferences [11, 12]. Agents have individual sets of capabilities, knowledge and resources that are made available to others (typically) in return for imposing some cost. Using their knowledge of others' capabilities agents can delegate the activities that must be performed in order to achieve their own goals. Agents are equal peers, and there is no overarching system control.

Agents have varying degrees of reliability, quality and honesty, and activities may fail, produce substandard results, or may cost more and take longer than expected. On delegating an activity, agents enter into an uncertain interaction since a high quality, timely, and on budget outcome is not guaranteed. Furthermore, since agents are autonomous peers they have no control over how others cooperate [10]. Thus, agents determine for themselves when to delegate activities or provide assistance, when to cease cooperating, and how to conduct themselves; agents can change the nature of cooperation, or even cease to cooperate, at any time. For example, an agent may choose to delay execution, reduce the quality of execution, or simply fail to complete an activity. To function effectively, agents must manage the risk of activity failure (or reduced performance).

In this paper we propose a mechanism, called *multi-dimensional trust with recommendations* (MDT-R), in which agents use trust and the recommendations of their peers to manage the risks of cooperating. Agents build models of their peers' trustworthiness along several dimensions, based on their experiences. When delegating an activity, a cooperative partner can be selected by combining these trust dimensions with peer recommendations and other decision factors (such as cost).

## 2. Cooperative design

In this paper, we describe our proposed MDT-R mechanism within the domain of cooperative design. We view the workflow of the design process as a directed graph, whose nodes correspond to activities and edges show dependencies between these activities. An edge from activity $A_i$ to $A_j$ indicates that $A_i$ is dependent on $A_j$ being performed. Furthermore, nodes may represent complex activities, and can be decomposed into corresponding primitive activities. For example, the activity of validating a design might be decomposed into configuring and running a simulation, followed by analysis of the results. Figure 1 illustrates an example fragment of workflow (shown at the bottom of the figure), where the sub-graph contained in the circle represents the decomposition of the indicated node. Thus, the workflow fragment at the bottom of the figure might represent the steps involved in validating a design, while the steps within the circle correspond to the lower-level tasks involved in executing one of these steps, such as the detailed steps that are required to run a simulation.

Individual activities might be specific design tasks or computational tasks such performing a simulation, however for our purposes we abstract out the details. An activity has certain requirements in terms of the capabilities and resources (e.g. knowledge or network bandwidth) needed for completion. For an agent to successfully perform an activity, these requirements must be met. An activity is (typically) dependent on other precondition activities (its children), and is a prerequisite for some activity (its parent).

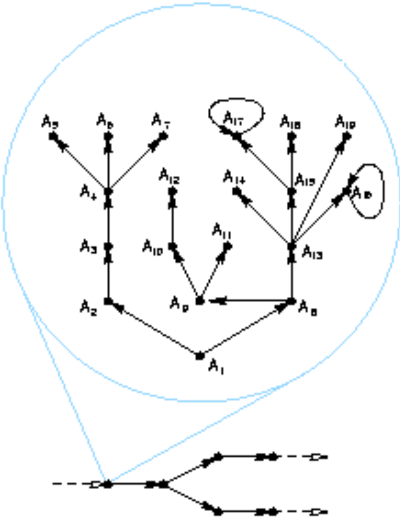


**Figure 1. An example workflow fragment.**

For each activity, a corresponding set of agents will meet the capability and resource requirements. The agent responsible for an activity must determine which of the capable agents to delegate its children to. For example, if an edge exists from $A_i$ to $A_j$ then the agent responsible for $A_i$ should delegate $A_j$. Agents that represent individual humans or design teams are also responsible for delegating child activities (although a human can typically override this choice if required). The root activity is allocated to an agent by the person (or agent) that instigates the design process. It is important to note that edges only represent dependencies, and *not* control. Agents are *equal peers* and delegation does not imply control over how the delegate performs an activity.

There is no central or hierarchical control. Agents are considered equal, and have no control over others, with the exception of delegating activities. Thus, the cooperative design process is viewed as a peer-to-peer multi-agent system, where agents themselves manage the allocation of activities. If substandard, erroneous or late results occur at one stage, these problems can propagate to subsequent activities, leading to errors, lateness, or complete failure. Furthermore, if failure occurs in an activity we assume that the agent responsible for the parent activity re-delegates execution to an alternative agent, i.e. all activities must be executed for the process as a whole to be successful.

## 3. Trust

Trust is a well recognised mechanism for assessing the potential risk associated with cooperating with autonomous agents [5, 6, 13]. Specifically, trust represents an agent's estimate of how likely another is to fulfil its commitments. Trust is divided into two categories: *experience-based* and *recommendation-based*. In the former, trust is based on individual experience, while in the latter it is based on information provided by others. Experience-based trust is simplest, where agents delegate activities and update their trust models according to the outcome. Recommendation-based trust is more complex, requiring agents to share information (based on their experiences) about the perceived trustworthiness of another. Our approach combines these two categories: agents maintain individual experience-based trust assessments of others, and combine these with information provided by others.

An obstacle to using recommendation-based trust is the subjectivity of trust. Agents build trust models based on individual experiences, and such subjective information may not be directly useful when shared with another agent. Thus, an agreed trust semantics is needed to enable recommendation-based trust. Agents must be able to interpret the information provided others. Our solution to this problem is described in Section 5.

### 3.1. Multi-dimensional trust

We take a multi-dimensional view of trust, decomposing it into beliefs about the different dimensions of an interaction, such as quality and timeliness. Activities are typically more than simple succeed or fail interactions. Agents cooperate with an expectation of successful performance, to a given quality and for some anticipated cost. In addition to possible failure, activities 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*. MDT-R does not prescribe specific trust dimensions, and agents can model trust along any number of dimensions. However, for the purposes of this paper the trust of an agent $\alpha$ is modelled in the dimensions of:

- success (denoted $T_\alpha^s$): the likelihood that $\alpha$ will successfully execute the task,
- cost (denoted $T_\alpha^c$): the likelihood that the cost of $\alpha$ executing the task will be no more than expected,
- timeliness (denoted $T_\alpha^t$): the likelihood that $\alpha$ will complete the task no later than expected, and
- quality (denoted $T_\alpha^q$): the likelihood that the quality of results provided by $\alpha$ will meet expectations.

Trust in any given dimension encompasses beliefs about competence, disposition, dependence, and fulfillment [4]. For example, if an agent is trusted in the quality dimension, then it is believed to be capable of performing an activity to a high quality (competence), actually doing so (disposition), being the preferred agent to do it (dependence), and being the means for activity achievement (fulfilment).

### 3.2 Representing trust

Trust initially takes a value according to an agent's disposition (optimistic or pessimistic). Optimists ascribe high initial values (implying low perceived risk), and pessimists ascribe low values. An agent's disposition also determines how trust is updated [14]. After interacting, optimists increase their trust more than pessimists in the dimensions where expectations were met and, conversely, pessimists decrease trust to a greater extent when expectations are not met. An agent's disposition comprises: the initial trust $T_{initial}$ ascribed in a trust dimension prior to interacting, and functions for updating trust after successful and unsuccessful interactions, $update_{success}$ and $update_{fail}$ respectively. These functions are heuristics applying to all trust dimensions, and they have no standard definition. 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 $\alpha$ along dimension $d$:

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

$$update_{fail}(T_{\alpha}^{d}) = T_{\alpha}^{d} - ((1 - T_{\alpha}^{d}) \times (\omega_f \times T_{\alpha}^{d}))$$

where $\omega_s$ and $\omega_f$ are weighting factors defined by the disposition. The trust for an agent $\alpha$ in a dimension $d$ has an associated numerical confidence level, $C_{\alpha}^{d}$, that is incremented on each application a trust update function.

Over time, trust values may become outdated if the experiences that gave rise to them are no longer relevant. To address this, we apply a decay function to converge trust values 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 $\alpha$ along dimension $d$ is defined as:

$$decay_{trust}(T_{\alpha}^{d}) = T_{\alpha}^{d} - ((T_{\alpha}^{d} - T_{initial}) / \omega_{td})$$

where the trust decay rate $\omega_{td}$ is defined by the disposition. As trust values become outdated, the confidence in them reduces. Thus, when applying $decay_{trust}$ the corresponding confidence level must be reduced. We define a decay function for the confidence level as:

$$decay_{confidence}(C_{\alpha}^{d}) = C_{\alpha}^{d} - (C_{\alpha}^{d} / \omega_{cd})$$

where the confidence decay rate $\omega_{cd}$ is defined by the disposition.

## 4. Stratified trust

We represent trust values numerically, however some researchers note that this can introduce ambiguity since the semantics are hard to represent [1, 13]. This is problematic when using recommendation-based trust, where agents share trust information. One solution is to divide the trust continuum into labelled strata, and use these to represent trust values [1]. However, the resulting semantics remain subjective, with different agents ascribing the same experiences to different strata. Furthermore, this representation reduces sensitivity and accuracy, and comparisons become coarse grained since the trust of agents with a stratum is indistinguishable. For this reason, we concur with Marsh in rejecting the use of strata in favour of numerical values [13]. Additionally, updating numerical trust is straightforward, yet stratified approaches often omit details of how strata are related to experience [1, 2].

However, the use of strata minimises overfitting; numerical values are not considered and so insignificant numerical differences in trust are not misinterpreted as important. An ideal trust model has the sensitivity and accuracy of a numerical approach, combined with the minimal risk of overfitting 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.

## 5. Interaction summaries

To enable sharing of trust information, there must be a clear semantics to the information provided. As discussed above, although stratified trust provides a possible solution, it still suffers from subjectivity. Several alternative approaches to recommendation-based trust have been developed, however each of them suffer from potential problems due to the subjectivity of information provided [9, 15, 17]. In MDT-R we avoid the subjectivity problem since, rather than attempting to provide an explicit assessment of trust, agents provide a summary of their relevant previous interactions. Thus, when agent $\alpha$ shares information with $\beta$ about $\gamma$, rather than communicating its trust value in a given dimension $T_{\gamma}^{d}$, it communicates a summary of the experiences that led to $T_{\gamma}^{d}$ (but not the value itself). As described in Section 3, trust is determined by whether an agent's expectations are met in a given dimension. For example, trust in the quality dimension $T_{\gamma}^{q}$ is determined by whether previous interactions were of suitable quality. When sharing information, $\alpha$ can communicate to $\beta$ the number of previous interactions with $\gamma$ in which quality expectations were met, $I_{\alpha\gamma}^{q+}$, and the number in which the quality was below that which was expected, $I_{\alpha\gamma}^{q-}$. The receiving agent β therefore obtains a summary of $\alpha$'s interactions with $\gamma$, and an indication of the extent of $\alpha$'s relevant experience (since $\alpha$ has had $I_{\alpha\gamma}^{q+} + I_{\alpha\gamma}^{q-}$ relevant interactions).

The first step in delegating an activity, is for the delegating agent to ask its trusted peers, where the general trust is above a minimum threshold, to provide information about each potential delegate. (General trust combines all trust dimensions and is based on Marsh's notion of general trust [13].) Only trusted peers should be asked for recommendations, to minimise the risk of dishonest or misleading information. Thus, each trusted peer $\alpha$ will provide $I_{\alpha\gamma}^{d+}$ and $I_{\alpha\gamma}^{d-}$, for each potential delegate $\gamma$ in each of the trust dimensions $d$. The delegating agent can then combine this information into a single recommendation value for each trust dimension. For each dimension $d$ the set of responses from the trusted peers about a potential delegate $\gamma$ are combined by summing the proportions of interactions where expectations are met, weighted by the extent of the peer's experience. Thus, the recommendation for a dimension, $R_{\gamma}^{d}$, is defined as:

$$R_{\gamma}^{d} = \sum_{i=\alpha}^{\xi} \left( \frac{I_{i\gamma}^{d+}}{I_{i\gamma}^{d+} + I_{i\gamma}^{d-}} \times \frac{I_{i\gamma}^{d+} + I_{i\gamma}^{d-}}{totalInteractions} \right)$$

where $\alpha$, $\beta$, …, $\xi$ are the set of trusted peers, and *totalInteractions* is the total number of interactions across all trusted peers with agent $\gamma$ in dimension $d$, defined as:

$$totalInteractions = \sum_{i=\alpha}^{\xi} \left( I_{i\gamma}^{d+} + I_{i\gamma}^{d-} \right)$$

Applying this approach, the delegating agent can determined recommendations in each of the trust dimensions of success, cost, timeliness and quality, denoted $R_{\gamma}^{s}$, $R_{\gamma}^{c}$, $R_{\gamma}^{t}$ and $R_{\gamma}^{q}$ respectively.

## 6. Delegating by combining trust dimensions

When delegating an activity, the various dimensions of trust, the recommendations from trusted peers, and any other relevant decision factors (such as advertised cost and quality) must be considered. An agent's preferences and its confidence in its trust models determine the emphasis given to these factors. For example, one agent may prefer to minimise the risk of failure and achieve the highest quality, while another may prefer to minimise cost. Similarly, if an agent has relatively low confidence in its own trust models, it may place more emphasis on peer recommendations.

To select between agents we use a weighted product model to combine choice factors and give a single performance value for each agent [3, 16]. 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:

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

where there are $n$ factors and $f_{\alpha_i}$ is the value for agent $\alpha$ in terms of the $i$'th factor and $\mu_i$ is the weighting given to the $i$'th factor in the selecting agent's preferences. The values of the weightings $\mu_i$ are defined by the selecting agent's preferences such that:

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

The best delegate is the agent $\alpha$ 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 $\mu_i$'s sum to 1, individual weightings can take any value in the interval [0:1]. This flexibility is a key strength of MDT-R, since the information maintained by an agent is the same, regardless of its current preferences and factor weightings. Furthermore, agents can use different weightings in different situations. For example, if an agent is relatively inexperienced more emphasis can be given to others' recommendations, and as experience is gained the emphasis can move toward its own trust models.

Factors such as quality can be used directly in calculating the performance value, provided that they are numerical and should be maximised. Similarly, peer recommendations can be used directly since they are numerical and to be maximised. Factors that should be minimised, such as cost, can be included by using:

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

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

Trust values must be stratified before inclusion, as discussed above. The trust range is divided 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:

$$stratify(t) = \lceil t \times s \rceil$$

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, cost, timeliness, and quality along with the corresponding peer recommendations in these dimensions. When delegating an activity 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:

$$\begin{aligned} PV(\alpha) = & \left(\max_c + 1 - \alpha_c\right)^{\mu_c} \times (\alpha_q)^{\mu_q} \\ & \times stratify(T_\alpha^s)^{\mu_{ts}} \times stratify(T_\alpha^c)^{\mu_{tc}} \\ & \times stratify(T_\alpha^t)^{\mu_{tt}} \times stratify(T_\alpha^q)^{\mu_{tq}} \\ & \times (R_\gamma^s)^{\mu_{rs}} \times (R_\gamma^c)^{\mu_{rc}} \\ & \times (R_\gamma^t)^{\mu_{rt}} \times (R_\gamma^q)^{\mu_{rq}} \end{aligned}$$

where $\alpha_c$ and $\alpha_q$ are $\alpha$'s advertised cost and quality respectively; $\max_c$ is the maximum advertised cost of the agents being considered; the weightings given to advertised cost and quality are denoted as $\mu_c$ and $\mu_q$; $\mu_{ts}$, $\mu_{tc}$, $\mu_{tt}$ and $\mu_{tq}$ are the weightings for the trust dimensions of success, cost, timeliness, and quality respectively; and $\mu_{rs}$, $\mu_{rc}$, $\mu_{rt}$, and $\mu_{rq}$ are the corresponding weightings for recommendations. This approach allows an agent to balance the relevant decision factors when selecting a peer for delegation

## 7. Example performance value calculation

By way of example, consider an agent selecting between two alternatives, $\phi$ and $\psi$, who requests recommendations from three trusted peers, $\alpha$, $\beta$ and $\gamma$. Suppose that these peers give the following information about their interactions with $\phi$ and $\psi$, with respect to the trust dimension of success (where $i$ takes the values $\alpha$, $\beta$ and $\gamma$ in turn).

| | $I_{i\phi}^{s+}$ | $I_{i\phi}^{s-}$ | $I_{i\psi}^{s+}$ | $I_{i\psi}^{s-}$ |
|---|---|---|---|---|
| $\alpha$ | 32 | 17 | 12 | 18 |
| $\beta$ | 65 | 32 | 79 | 13 |
| $\gamma$ | 3 | 7 | 48 | 42 |

The delegating agent must calculate recommendation values for both alternatives. Firstly, the total number of interactions with agent $\phi$ is calculated, using:

$$totalInteractions = \sum_{i=\alpha}^{\gamma} \left(I_{i\phi}^{s+} + I_{i\phi}^{s-}\right)$$

which gives a value of 156. The recommendation for $\phi$ in the success trust dimension, can now be calculated as follows:

$$\begin{aligned} R_\phi^s = & (32/(32+17) \times (32+17)/156) \\ & + (65/(65+32) \times (65+32)/156) \\ & + (3/(3+7) \times (3+7)/156) = 0.64 \end{aligned}$$

Similarly, for alternative $\psi$ we get $R_\psi^s$=0.66.

Suppose that the information received from the trusted peers regarding the other trust dimensions is such that the delegating agent calculates the recommendation values given below, and suppose that the other decision factors being considered have the following values.

| factor | $\phi$ | $\psi$ |
|---|---|---|
| advertised cost (units per second) | 10 | 9 |
| advertised quality (range 1 to 10) | 9 | 8 |
| trust (success dimension) | 0.43 | 0.81 |
| trust (cost dimension) | 0.90 | 0.64 |
| trust (timeliness dimension) | 0.71 | 0.87 |
| trust (quality dimension) | 0.78 | 0.42 |
| recommendation (success dimension) | 0.64 | 0.66 |
| recommendation (cost dimension) | 0.43 | 0.57 |
| recommendation (timeliness dimension) | 0.77 | 0.66 |
| recommendation (quality dimension) | 0.51 | 0.66 |

Furthermore, suppose that the following factor weightings are used, i.e. quality is given more emphasis that cost, trust is given the most emphasis, and recommendations are given the least emphasis.

| | | | | | |
|---|---|---|---|---|---|
| $\mu_c$ | 0.08 | $\mu_{ts}$ | 0.15 | $\mu_{rt}$ | 0.05 |
| $\mu_q$ | 0.12 | $\mu_{tc}$ | 0.15 | $\mu_{rq}$ | 0.05 |
| | | $\mu_{tt}$ | 0.15 | $\mu_{rs}$ | 0.05 |
| | | $\mu_{tq}$ | 0.15 | $\mu_{rc}$ | 0.05 |

The agent must calculate the performance value of each of the alternative partners. Thus, applying *PV()* to agent $\alpha$ gives:

$$\begin{aligned} PV(\alpha) = & \left(10 + 1 - 10\right)^{0.08} \times 9^{0.12} \times stratify(0.43)^{0.15} \\ & \times stratify(0.90)^{0.15} \times stratify(0.71)^{0.15} \\ & \times stratify(0.78)^{0.15} \times 0.64^{0.05} \times 0.43^{0.05} \\ & \times 0.77^{0.05} \times 0.51^{0.05} \\ = & 1^{0.08} \times 9^{0.12} \times 5^{0.15} \times 10^{0.15} \times 8^{0.15} \times 8^{0.15} \\ & \times 8^{0.15} \times 8^{0.15} = 8.15 \end{aligned}$$

Similarly, for agent $\psi$ we get $PV(\psi)$ = 6.53. Therefore, based on these weightings, agent $\phi$ is the alternative that best balances the factors considered.

To demonstrate how factor weightings allow agents to balance their preferences, suppose instead that the following weightings 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).

| | | | | | |
|---|---|---|---|---|---|
| $\mu_c$ | 0.15 | $\mu_{ts}$ | 0.025 | $\mu_{rt}$ | 0.025 |
| $\mu_q$ | 0.15 | $\mu_{tc}$ | 0.15 | $\mu_{rq}$ | 0.15 |
| | | $\mu_{tt}$ | 0.025 | $\mu_{rs}$ | 0.025 |
| | | $\mu_{tq}$ | 0.15 | $\mu_{rc}$ | 0.15 |

In this case we get performance values of $PV(\phi)$ = 5.42 and $PV(\psi)$ =5.57. Thus, where greater emphasis is places on quality and cost, $\psi$ is considered the best alternative. In general, since each peer is autonomous we cannot guarantee that an agent makes the absolute optimal choice. However, our proposed approach makes the best expected choice given the agent's experiences so far.

## 8. Discussion

To investigate the effectiveness of our proposed MDT-R model, we have constructed a simulation prototype. We have investigated the performance of MDT-R using several, relatively small-scale, process graph fragments containing around 500 activities, each of which has specific capability and resource requirements. Our simulation contains 100 peer agents, each with specific capabilities, that use MDT-R to delegate activities. Our initial results are promising, showing that MDT-R provides a significant advantage over alternative methods (e.g. delegation based on advertised cost and quality, general trust, and multi-dimensional trust without recommendations). For example, in a mixed environment (i.e. one where there is a uniform mix of reliable and unreliable/honest and dishonest agents) MDT-R provides an improvement of up to 30% in achieved quality, and up to a 20% decrease in failure rate, over simple advertised cost and quality delegation methods.

Furthermore, MDT-R allows agents to delegate activities according to their current preferences, by selecting appropriate weightings. Thus, an agent that is concerned about cost can place more emphasis on advertised cost, and trust and recommendations in the cost dimension. Similarly, an agent that has relatively little experience, and so low confidence in its trust models, can emphasise peer recommendations. This is a very flexible approach, allowing the delegation process to be tailored to the priorities of an agent at any given node of the process graph.

## 9. Conclusions

In this paper we have proposed the notion of multi-dimensional trust and provided a mechanism for peers to share information about their experiences. Our proposed model allows agents to model the various facets of trust, and combine these with information provided by peers (along with other decision factors) when delegating an activity. The MDT-R model is highly flexible, and system designers have full control of the trust dimensions modelled and the relative weightings given to the decision factors.

Currently, the weightings for decision factors are specified by the system designer. Although the designer may specify different weightings for different situations, agents cannot determine appropriate weightings for themselves. Future work involves exploring mechanisms, such as learning and genetic algorithms, to enable agents to tailor the weightings according to their preferences (e.g. maximising quality or minimising failures). Although we have validated MDT-R in a simulation prototype, we are performing ongoing experimentation, and aim incorporate MDT-R into a real-world peer-to-peer system.

## 10. References

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

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

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

[4] C. Castelfranchi, *"Trust mediation in knowledge management and sharing"*, Proceedings of the 2nd International Conference on Trust Management, pp. 304-318, 2004.

[5] C. Castelfranchi and R. Falcone, "Principles of trust for MAS: Cognitive anatomy, social importance, and quantification.", Proceedings of the 3rd International Conference on Multiagent Systems, pp. 72-79, 1998.

[6] D. Gambetta, *"Can we trust trust?",* Trust: Making and Breaking Cooperative Relations, D. Gambetta (ed), pp. 213-237, Basil Blackwell, 1988.

[7] N. Griffiths, *"Task delegation using experience-based multi-dimensional trust"*, Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems, 2005, to appear.

[8] N. Griffiths and M. Luck, *"Coalition formation through motivation and trust"*, Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems, pp. 17-24, 2003.

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

[10] M. d'Inverno and M. Luck, *"Understanding Autonomous Interaction"*, Proceedings of the 12th European Conference on Artificial Intelligence, pp. 529-533, 1996.

[11] N. R. Jennings, *"Cooperation in industrial multi-agent systems"*, World Scientific, 1994.

[12] M. Luck, P. McBurney and C. Preist, *Agent Technology: Enabling Next Generation Computing*, AgentLink II, 2003.

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

[14] S. Marsh, *"Optimism and pessimism in trust"*, Proc. of the Ibero-American Conference on Artificial Intelligence, 1994.

[15] J. Sabater and C. Sierra, *"REGRET: A reputation model for gregarious societies"*, Proceedings of the 1st International Conference On Autonomous Agents in Multi-Agent Systems, pp. 475-482, 2002.

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

[17] B. Yu and M. P. Singh, "An evidential model of reputation management", *Proceedings. of the 1st International Conference on Autonomous Agents in Multi-Agent Systems*, pp. 295-300, 2002.