

# Delegations and Trust

Henry Hexmoor\*, Rachil Chandran\*\*

\*Southern Illinois University, Carbondale, IL, 62901, hexmoor@cs.siu.edu

\*\*University of Arkansas, Fayetteville, AR 72703

**Keywords:** multiagent systems, trust, delegation.

**Abstract:** One of the fundamental notions in a multiagent system is that of delegation. Delegation forms the foundation for cooperation and collaboration among the members of a multiagent system. In diverse environments such as those formed by open multiagent systems, the various members constituting the environment are customarily alien to one another. Delegation decisions in such environments are necessarily of a nontrivial nature owing to the fact that there is a lack of strong basis on which such a decision can be predicated. Trust is a primary social notion providing a foundation for dealing with disparateness in virtual societies. Trust facilitates de-alienation of the otherwise mutually unfamiliar components of a virtual society. This work investigates and alleviates the problems associated with delegations in multiagent systems by using delegation decisions predicated on trust.

## 1. Introduction

A computer program capable of independent proactive behavior is known as an autonomous agent. A multitude of such agents coexisting together in a social setting form a multiagent system. A multiagent system targets a task in a distributed manner. By segregating the goal to be achieved, such systems often complete tasks that cannot be fulfilled by a single agent. Hence, one of the primary motivations for the existence of multiagent systems is the completion of objectives through coordination among members of the system.

Cooperation in a multiagent system amounts to the completion of different components or parts of a task by different agents leading to the effective completion of the overall task. This cooperation is commonly achieved through the delegation of tasks from one agent to another.

Delegation is the procedure by which one computing entity can validly (and securely) instruct another computing entity to perform some actions on its behalf (Hardjano et. al., 1993). Delegation involves two entities. The one that delegates or transfers an object is known as the delegator and the one that receives the order and completes the task delegated to it is known as the delegatee. The object of delegation that may vary in granularity is the work being delegated. It could be a task, a role, or even a complete goal (which in turn may mandate completion of multiple tasks).

Based on its constituents, a multiagent system can be categorized into two types: (a) a heterogeneous or open multiagent system, and (b) a homogeneous multiagent system. In order to understand the problem we address by this work, it is necessary to

understand the concept of an open heterogeneous multiagent system and the fundamental way it differs from a homogeneous system.

A heterogeneous or open multiagent system can be described as a system composed of varied entities. These agents are not designed as a team, but have their own individual goals and hence are sometimes referred to as self-interested agents. They cooperate with the other members only to achieve their own objective. They execute tasks delegated to them in order to gain the incentives associated with that task. In other words, the agents in such systems provide a delegation service to other agents for a price that incur a delegation cost for the delegator. Since each has her own goal, the members of such a system are said to be disparate and since they are designed by different users, members are strange to one another.

Heterogenous multiagent systems can be contrasted with the homogeneous multiagent system where the constituent members form part of a team with a common goal. These team members would be benevolent towards one another and complete delegated objects without requiring to receive a profit; they aim to contribute towards the common goal of the group.

An open heterogeneous multiagent system consists of disparate, self-interested agents. In such a system, there can be no assumption of benevolence among the agents (Griffiths, 2005). A malevolent agent may not complete a delegated task, may complete it unsatisfactorily, or may misuse privileges associated with that task. For instance, a delegated role often comprises of access rights associated with that role. A dishonest, self-interested agent may misuse these rights for her own gain.

The problem we address is a selection problem: the problem of “to whom should an object be delegated?” Given options (i.e. delegates), which option, when selected, would have the highest probability of completing the delegated object effectively and securely; i.e., with the least misuse of information or rights pertaining to the object. We term this selection problem as the *delegation decision problem*.

In an open system, the difficulty in selection arises from the fact that there can be no assumption about the benevolence of these prospective delegates towards the delegator. Although they may share protocols for communication, they are fundamentally unknown to each other; i.e., they have no intimate knowledge of each other’s benevolence or capability. Hence, there exists a need for some attribute based on which the various delegatee options can be compared and a delegation decision predicated. This attribute should serve to indicate an agent’s benevolence and ability.

The *delegation decision problem* is that of selecting a suitable delegatee from a set of available options. In an open multiagent system, the members are socially strange to each other and that makes delegation decisions in such systems a non-trivial process. To aid effective delegation decisions, various members of an open system should be made familiar with each other based on their interaction history.

We address the delegation decision problem by providing a protocol for such decisions. The protocol we propose is based on the social notion of *trust*.

It should be noted that there may be a number of different reasons for a delegator to choose a particular delegatee. For instance, a particular delegatee may be the only one that possesses the knowledge or resources required for the completion of a delegation object. Other considerations like cost of delegation can also play a role in the selection of a delegatee. However, here we limit our focus on delegation protocols solely on trust.

## 2. Related work

In this section we offer fairly comprehensive accounts of trust and delegation. Since it is difficult to navigate the literature on these topics, our two subsections will serve as detailed primers to the literature.

### 2.1 Trust

In recent years, considerable resources and research have focused on understanding, characterizing, categorizing, managing and modeling the concept of trust (Sabater and Sierra, 2005). In his work, Gambetta (Gambetta, 1990) exemplifies trust in the following manner:

“When I say that I trust Y, I mean that I believe that, put on test, Y would act in a way favorable to me, even though this choice would not be the most convenient for him at that moment”

Gambetta further describes trust as a level of subjective probability with which an agent will perform a particular action (in Gambetta, 2000). Castelfranchi and Falcone on the other hand describe trust as not only a mere subjective probability, but also as a mental state (Falcone and Castelfranchi, 1998). They contend that trust is a complex attitude exhibited by one entity towards another. This definition is further supplemented by Beavers and Hexmoor when they define trust as one agent’s belief in the intention of another agent with regards to the welfare of the former (Beavers and Hexmoor, 2003). Trust is a belief an agent has that the other party *will do as it says it will (or reciprocates)*, given an opportunity to defect (Dasgupta, 1998). Hence trust represents an agent’s estimate of the risk involved in entering into a potential partnership (Griffiths, 2005). Unique to our model of trust is a suggestion of a soft computing approach to mathematically modeling trust relationship among agents. If an agent believes in the potential partner’s benevolence towards itself (in other words trusts the latter), it can reason the partnership to be carrying a low degree of risk.

In a slight variant to these definitions, trust is considered to be the ability to believe in a subject despite uncertainty concerning its possible actions (Harris, 2002). This approach to define trust can be contrasted with previous approaches by its implication that trust is not necessarily based on knowledge. Therefore, here trust is described as resulting from an “assumption of benevolence”. Since this assumption is not a feasible one in the context of virtual societies, the corresponding definition of trust is also inadequate.

One of the foremost features of trust as reported by Castelfranchi and Tan is its ability to reduce the amount of risk involved in agent interactions (Castelfranchi and Tan, 2001). This ability is attributed to trust as a consequence of its effect on efficient partner selection (Schillo, 1999).

In their work, Abdul-Rahman and Hailes describe trust as a subjective degree of belief that ranges from complete trust to complete distrust (Abdul-Rahman and Hailes, 2000). Hence, as in our work, trust is not modeled to be binary in nature (i.e. only trust or distrust) but can be expressed in quantity. This value of trust can then be used to signify the extent or degree of trust (or Distrust) exhibited (Falcone and Castelfranchi, 1998). This in turn leads to quantification of trust which in itself is considered to be a separate field of study. Usually trust is represented as a real number along the positive axis of the

number line. It is to be noted that the number implies only comparative values and have no strong semantic meaning (Griffiths and Luck, 2003). For instance, while some models of trust might use a value of 1 to indicate *complete* and 0 to indicate *complete distrust*, some other model may use the values of 100 and 0 to indicate the same. To avoid such potential ambiguity, a stratified approach to trust is recommended (Abdul-Rahman and Hailes, 2000). Here the authors use four categories or ‘strata’ of trust - “very trustworthy”, “trustworthy”, “untrustworthy” and “very untrustworthy”.

Trust can also have multiple dimensions where each dimension represents an agent’s belief for a particular aspect of the environment (Griffiths, 2005). These dimensions are not constituents of trust itself (like belief), but are rather trust itself expressed with respect to some particular perspective.

The nature of the source of trust is one important criterion on which to base a classification of trust. Accordingly, there are two forms of trust – experience-based and recommendation-based trust (Griffiths, 2005). In the context of one entity’s trust in another, experience-based trust stems from the former entity’s knowledge, familiarity and outcome of previous communications with the latter entity. As can be expected, this is the most widely understood and used form of trust. Recommendation-based trust on the other hand arises not from an entity’s own experiences and communications, but from shared information with other entities (Sabater and Sierra, 2002). For instance, an agent A’s trust in agent B can be a consequence of agents C, D, E etc. sharing information about B with A. As can be perceived, recommendation-trust has a strong relation with the notion of reputation in that it results from perception of one entity by all other subjects. Therefore, recommendation based trust requires active sharing of information between subjects as to how trustworthy another is perceived to be. This might not be feasible in all forms of virtual organizations, particularly ones in which each agent is a ‘self-interested’ entity (Griffiths, 2005). Hence, recommendation-based trust is not as popular as the experience-based trust. Some researchers are, however, investigating the use of recommendation-based trust (Huynh et. al., 2004, Ramchurn et. al., 2003, Yu and Singh, 2002).

Another well accepted factor of classification is the context in which a particular trust value is applicable. Marsh suggested two types of trust based on this factor; namely, general trust and situational trust (Marsh, 1994). General trust is the value of trust with no specific circumstance applied. Here trust may be compounded using a variety of attributes, consequently giving a generic affect to the nature of trust. In contrast, situational trust is a value that only has meaning in a particular context or situation. The situation can be a particular environment, a specific background or with respect to a unique goal. Although related to one another, the notion of specific trust and multi-dimensional trust are two distinct conceptions (Griffiths and Luck, 2003).

Besides categorizations outlined here, there are other ways in which trust has been classified. One such method, used by Falcone and Castelfranchi, is the distinction between different forms of trust based on the belief on which the trust is founded (Falcone and Castelfranchi, 1998). They categorize belief into competence, disposition, dependence and fulfillment belief. The trust that results from each of the unique beliefs is in turn unique and distinct in nature. Beavers and Hexmoor categorize trust as either active or passive depending on the necessity (or lack of it) as an explicit act on the part of the trusted entity (Beavers and Hexmoor, 2003). In the same work, the authors suggest another classification, voluntary/thick trust and forced/thin trust. Trust obtained due to

monitoring and/or control is termed as thin trust. Such ‘strong-arm treatment’ obtained trust is recognized as undependable by Shapiro and Sheppard (Shapiro, Sheppard, 1992). Thick trust, in contrast is more enduring as a consequence of its voluntary nature (Ring, 1996).

## 2.2 Delegation

Definitions and descriptions of the concept of delegation are numerous in the research literature. Not all of these descriptions are consistent. In this section, we primarily examine those definitions of delegation that are most relevant to the theme of our work and do not attempt to be exhaustive.

“Delegation concerns the method of how one computing entity can validly (and securely) instruct another computing entity to perform some actions on its behalf” (Hardjano et. al., 1993, page 1). Delegation is the procedure by which tasks (or other higher units like roles) can be assigned to a different, possibly disparate entity (Castelfranchi, and Falcone, 1999, Abadi et. al., 1991). Basically, through delegation, a user in a distributed system authorizes another system or user to access resources on her behalf (Gasser and McDermott, 1990).

An accurate, but more limited definition was provided by Ahsant. In this work, the authors describe delegation as an “act of transferring rights and privileges to another party (the delegatee)” (Ahsant et. al., 2004, page 1). Another definition that conforms to the aforementioned is provided in (Castelfranchi and Falcone, 2002). In this work, delegation is illustrated as an act of entrustment of some object.

Another notion introduced by Castelfranchi and Falcone is that of adoption (Castelfranchi and Falcone, 1999). Adoption is used as a concept that is very similar to delegation and is frequently misconstrued as an alternative form of delegation. Although similar in some respects, adoption is a concept that is separate from delegation. Castelfranchi and Falcone define adoption as a process by which one entity accepts or “adopts” an object from another entity (Castelfranchi and Falcone, 1999, Falcone and Castelfranchi, 2000). The difference between delegation and adoption stems from the fact that adoption is a delegatee (in this case adopter) initiated process as opposed to the delegator initiating the process of delegation. For instance, if A and B are the agents concerned in a collaboration, delegation occurs when instigated by A whereas adoption entails B to voluntarily proffer services to A. Also, the feasibility of adoption exists only if B has overlapping goals with A (Castelfranchi and Falcone, 1999). We base our work on pure delegations and do not consider adoption in our models.

Categorization of delegation is frequently founded on different levels or granularity of objects that are delegated. Based on this criterion, delegation is classified into two primary categories, Task delegation and role delegation (Castelfranchi and Falcone, 1999, Castelfranchi and Falcone, 1997). In task delegation, an agent transfers a task to another agent for completion. This represents the most basic type of delegation, in that, task is the object with minimum granularity that can be involved in cooperation among agents. Here, the outcome of the entire delegation process is dependant on the result of the task execution by the delegatee agent (Griffiths and Luck, 2003). In role delegation, the rights and obligation of a role are transferred to the receiving agent. Here, the delegatee agent is expected to perform several different tasks required to fulfill the particular responsibility that is associated with the delegated role. In this sense, roles are

of a higher granularity than tasks. Role delegation is also known as rights delegation (Hardjano, et. al., 1993). Role delegations are very commonly used in open systems for the purpose of Role Based Access Control or RBAC (Barka and Sandhu, 2004). Yet another level of granularity commonly related to delegation objects is that of a goal (Bergenti, et. al., 2003). In goal delegation, the delegator agent entrusts a goal to the delegatee agent. This type of delegation entails a high degree of trust and harmony between the agents involved in the delegation.

Another important attribute employed in the categorization of kinds of delegation is that of single or cascaded delegation. In single delegation, an object is delegated to a delegatee and the responsibility or obligation for that particular object is satisfied by the delegate itself. In cascaded delegation on the other hand, the delegated object (or a part of it) is re-delegated by the initial delegatee to further entities (Tamassia, et. al., 2004). Here, re delegations form a chain from the initiator to the final recipient. Hence, cascaded delegations are also known as chain delegations (Faulkner, et. al., 2005). It should be noted that the further recipients (excluding the initial delegatee) can potentially be unknown to the initiator of the chain (Tamassia, et. al., 2004). A chain delegation scenario is illustrated in the figure below.

Yet another classification of delegation based in its nature was presented by Faulkner et. al. (Faulkner et. al., 2005). Here, delegation is segregated into forced (or blind delegation) and free delegation. Forced delegation between two entities, as a norm, does not depend on trust between those entities (Falcone and Castelfranchi, 1998). Free delegation, however, mandates a minimum amount of trust between the participating entities. Having delineated on this particular categorization, we posit that forced delegations are relatively atypical in an environment composed of self-interested entities. Hence, for the scope of this work, we address only free delegation.

In their work on delegation, Castelfranchi and Falcone define various classification of delegation based on the level of agreement between the parties (Castelfranchi and Falcone, 1999). Weak delegation, according to the authors, occurs when there is a very limited level of belief and consequently agreement between the delegator and delegate. Mild delegation arises for inductive belief and string delegation for complete belief and agreement between the involved agents. Open and closed delegation is another form of classification introduced by the authors in the same work. In open delegation, the object of delegation is commended to the recipient without any specific plan on the completion of the object. In closed or specific delegation, the plan of execution of the object accompanies the delegation and the recipient is responsible to not only complete the object of delegation, but also to complete it in accordance with the executive plan.

### **2.3 Models for Delegations**

The importance and necessity of a secure means of delegation was long overlooked by the research communities. However, in recent times, the topic of delegation has drawn considerable attention from both the information assurance and multi-agent research communities. In this section, we review important relevant works.

In their early work on agent research, Jennings et al. recognized that

“For an individual (agent or otherwise) to be comfortable with the idea of delegating tasks to agents, they must first trust them” (Jennings, Sycara, Wooldridge, 1998, p. 31).

According to Wong and Sycara, delegations are one of the four main sources for security threats in a multi-agent environment (Wong, Sycara, 1999). They propose a solution by means of an authentication mechanism. The delegatee agent is queried for an answer known only to the delegator agent. This is a form of authentication mechanism to ascertain the identity of a certain individual entity and does not address the delegation problem at a policy level. The solution proposed in the RETSINA model “is not satisfactory as nothing prevents a delegatee from misusing the secret” (Sycara, et. al., 2003). In another work, Hu recommends the usage of Public Key Infrastructure (PKI) mechanism from the X.509 standard (Hu, 2001). Here again the emphasis is on how to delegate effectively and securely rather than to whom to delegate. Hu advocates a centralized server as a certification authority, but does not take into consideration complexities that arise when the notion of trusting the server arises. Hence, though providing us with an excellent insight of general delegation mechanisms In another attempt to devise a secure delegation model, Tian-chi and Shan-ping propose an exchange server that implements ‘security instance’ which is an authorization mechanism with respect to code segments of an agent (Tian-chi and Shan-ping, 2005). Like Hu’s model, this solution depends completely on a centralized server. Moreover, the proposed model is dependent on the participant agents’ ability to transfer code from one machine to another (i.e., as in the case of mobile agents).

Faulkner proposed a generic model of delegation which considers only the ontological aspects of delegation (Faulkner et. al., 2005). The mechanisms proposed in the work consider different schema attributes for different types of delegation. Although the work does mention trust as an inherent component of delegations, the authors make no attempt to incorporate the notion into the working of their system. Yet another model in this category was put forth by Barka and Sandhu in their work recommending delegation as an implementation methodology role based access control (Barka and Sandhu, 2000). Here the authors attempt to model computational delegations based on characteristics of human to human delegations. Delegation mechanisms are designed using cases generated from these characteristics. This work was extended by the authors by considering the semantics that can impact the delegate scenarios (Barka and Sandhu, 2004).

In one of the earliest works on delegation systems, there appears a formulation of the delegation/revocation mechanism that is independent of any trusted third party servers or authorities (Gasser and McDermott, 1990). The system incorporates concepts such as life of delegation to ensure security. Delegation keys are another novel concept proposed in this work. Delegation keys ensure that there is no threat in the event of a compromise on a previously trusted (and hence previous delegate). The model uses a token based approach to delegations.

Similar token-based delegation models are recommended in other works too (Sollins, 1988, Christianson, 1994 and Ding et. al., 1996). In these models, delegation mechanisms are dependent on a third party issued token. These heuristics mandate the need for both the delegator and delegatee to explicitly trust the token issuing third party. Parallel, but distinct from this approach is one described by Abadi et al. (Abadi et. al.,

1990). The authors here propose the use of smart cards for managing a delegation based authentication system.

Another concept employed in task-delegation is a two phase certification server or 2PCS (Hardjono, 1992). In this approach, public key cryptography is used. Each entity involved in a delegation has a permanent pair of keys and a number of delegation keys, one per each delegation. A Secure Key Management Center (SKMC), is a system which applies its own key to a delegation. This work is another example of a delegation mechanism that focuses on alleviating issues like non-repudiation and accountability, not on delegation decision support.

An attribute-based delegation model (ABDM) was put forth by Ye et. al. (Ye, et. al., 2004). The work is an extension of the role-based delegation model proposed in (Barka and Sandhu, 2000). The ADBM is more security oriented than the RBDM (Role-Based Delegation Model) in that it has more constraints with attribute-based values that have to be satisfied for a particular delegation to materialize. However, the model does permit “undecided delegations” that are not strictly based on a predicate.

Griffiths introduced a trust-based delegation model in (Griffiths, 2005). The work is based on experience-based trust as opposed to recommendation-based trust. It also considers different *dimensions* on which to base the trust value. The dimensions here are different from the concept of context. Dimensions are basically criterion like cost, quality, etc. In another trust based approach to delegation, a Grid Delegation Protocol (GrDP) was developed by Ahsant et al. (Ahsant, Basney, and Mulmo, 2004). The GrDP is based on the Web Services Trust (WS-Trust) specification. The model presented provides flexibility in terms of the underlying mechanisms that may be used for delegation implementation. The model is particularly useful for grid-based applications that require delegation for their effective operation.

A delegation model based on action and states is found in (Norman, and Reed, 2002). A language was developed to express the various states and actions. Actions would transform the system (or a component of it) from one state to another. Consequently, subjects in the system are associated with the actions and are held “responsible” for them. Therefore executers of delegations (delegates) can be mapped to their responsibilities.

### **3. A Model of Delegation Founded on Trust**

#### **3.1 Definitions**

The concepts proposed here will lead to a superior model on which delegation decisions can be founded. In this section, we define and describe components of our model.

*Delegation group:* A delegation group is a set of agents co-existing together in a given environment. A delegation group represents an open environment in that, the agents constituting the group are autonomous, self interested and disparate components. We define the characteristics of the agents composing the group in the following manner:

- The agents forming the delegation group may or may not have a common goal. They represent possibly disparate entities and hence work towards the goal of the entities. Hence, we term the agents as self interested.

- Due to their unique goal property, the collaboration among the agents in the group takes the form of delegations. More precisely, an agent collaborates through delegation of an object that is a part of its goal. The completion of this object is undertaken by another agent in the group (the delegatee). Since the delegatee's goal need not overlap with that of the delegator, the delegate offers the completion of the delegated object for some gain (Griffiths, 2005).
- Another attribute we associate with the members of the delegation group is the notion of autonomy. A formal definition for autonomous agents was proposed by Franklin and Graesser (Franklin and Graesser, 1996). They define an autonomous agent as entities capable of independent, directed and purposeful action in the real world. In the context of the delegation group, the autonomy of its members is manifested in delegation decisions, i.e. agents can decide to which entities to delegate an object.

Cohesive force that binds agents together is another property that is distinct in the delegation group. Since the agents may or may not be working toward a common goal, the cohesion among the agents in the delegation group is not as strong as the one that exists among the members of a team. The agents composing the delegation group are bound together by the objective of delegating or receiving delegations. For this reason, the cohesive force among the agents is not as weak as that found in crowds (Silverman, et. al., 2001). Therefore, the collection of agents outlined in the above paragraphs form a group.

*Delegation harmony*: Delegation harmony (i.e., DelHarmony) serves as an indicator of the degree of accord or concurrence between two given entities (Chandran and Hexmoor, 2007). We compute the value of DelHarmony based on the previous interaction history of the concerned agents. The formula employed to quantify the notion of DelHarmony is as follows:

$$DH(A, B) = \frac{\text{Number of honored delegations}}{DR(A, B)} \times \log_{10} \{DR(A, B)\}$$

DR(A,B) represents the total number of delegation requests between A and B.

For instance, when agent A delegates an object to agent B, the delegation is considered to be honored if and only if agent B accepts the obligation of the delegation and fulfills the obligation to the satisfaction of agent A.

An important attribute of DelHarmony is its asymmetry. In other words, DH(A,B) need not necessarily have the same value as DH(B,A). For instance, if agent A has delegated a substantial number of tasks to agent B, and agent B fulfills all the delegations, agent A's DelHarmony towards B can be significantly higher than that of B towards A (assuming B has not delegated to A).

As will be seen in the succeeding sections, delegation harmony is an important concept and helps us to devise a dynamic approach to trust updating and maintenance. Due to the fact that the value of DelHarmony is directly proportional not only to the number of honored delegations, but also to a function of the total number of delegations

that have transpired between the entities, DelHarmony reflects, reasonably accurately, the degree or extent of accord.

We suggest that DelHarmony augments reliability and accuracy of trust values.

Trust Value: Delegations in our model are predicated on trust. The value of trust is a numerical entity based on which we make a delegation decision. It is an indication of the amount or degree of belief that one subject has in the ability and benevolence of another.

The trust value is always stated between two given subjects. We represent trust in our system as  $T(A,B)$  where  $A$  is one agent and  $B$  is another.  $T(A,B)$ , it should be noted, is the trust that agent  $A$  has in agent  $B$  and not vice versa. This point is of considerable significance due to the fact that the trust value is an asymmetric one i.e.  $T(A,B)$  need not be the same as  $T(B,A)$ . This value of trust is updated according to policies outlined in the succeeding sections. An agent in the delegation group maintains trust values for all other agents in the group.

Trust incentives and penalties: To keep a record of a particular agent's performance with respect to delegations, we need parameters that update the trust value to account for the result of a delegation.

Trust incentives and penalties are parameters that help with updating of the trust value based on the different outcomes of a delegation. In the event of a successfully completed delegation, the trust value of the delegator towards the delegatee is incremented to reflect the success of the interaction. We term the value, by which augmentation to the trust is computed, as trust incentive. Similarly, the delegatee is penalized for its inability to fulfill the delegation it accepts. In such circumstances, the value of trust penalty is deducted from the trust value that the delegator currently has towards the delegatee. This provides further motivation for the delegatee to fulfill the obligation of delegation completely.

We formulate the augmentation to and deduction of trust based on delegation outcomes as follows,

Trust increment:

$$T(A,B) = T(A,B) + \alpha (A,B)$$

Trust decrement:

$$T(A,B) = T(A,B) - \beta (A,B)$$

where  $\alpha (A,B)$  and  $\beta (A,B)$  are the trust incentive and trust penalty values for agent  $A$ , the delegator, with respect to agent  $B$ , in this case, the delegatee.

It should be noted that the value of  $\alpha$  and  $\beta$  are not fixed, static values but are rather, dependent on the two entities involved. In other words, the trust incentive value of agent  $A$  with respect to agent  $B$  need not be the same as that of  $A$  to  $C$ . Also, the value of trust incentive and penalty are asymmetrical i.e.  $\alpha (A,B)$  may not be the same as  $\alpha (B, A)$  and  $\beta(A,B)$  may not be the same as  $\beta(B,A)$ . The computation of the  $\alpha$  and  $\beta$  can be represented as given below,

$$\alpha(A,B) = \omega \times DH(A,B)$$

and

$$\beta(A,B) = \omega / DH(A,B)$$

where,  $\omega$  is the weight based on the importance of the object being delegated and  $DH(A,B)$  is the DelHarmony between agent A and agent B.

We propose a stratified approach to the importance of delegation object. Consequently,  $\omega$  can take only four values - 0.25, 0.5, 0.75 and 1.0 with 0.25 being assigned to the objects of least importance (like actions having less threat) and 1.0 being assigned to object of highest importance (goals which may entail a high risk of breach). Hence, the incentive or conversely the penalty incurred by the delegatee is directly proportional to the significance of the object involved in the delegation. Hence the revised value of trust after a delegation reflects a more accurate indication of the delegatee's dependability with the decisiveness of the delegation in prespective.

The trust incentive value is also directly proportional to the value of DelHarmony between the concerned agents. Hence, a high degree of accord and positive interaction history between the delegator and delgatee would lead to a greater value of incentive for further positive interactions. Similarly, the value of trust penalty is inversely proportional to the value of DelHarmony. We opine that this technique leads to a dynamic approach of updating trust in that inconsistencies arising due to infrequent failures in a long interaction history do not adversely affect the trust value.

*Elimination threshold:* One of the primary goals of the delegation group, as explicated earlier, is to provide a forum for agents that intend to cooperate through delegation. However, the notion of delegation group also aspires to maintain a level of reliability among it's members. A malicious agent with a record of frequent breach of delegation obligation should not contend to offer it's delegation services in the group. This can be achieved by expelling persistently malicious agents from the delegation group. Hence, frequently 'lying' agents are proscribed from continuing as members of the group. This in turn leads to members of a delegation group representing a minimum level of performance and benevolence in the context of delegations. Also, potential delegator agents have lesser risks as they do not consider the expelled agents as possible delegate.

The chief parameter that governs the elimination of temporally malicious agents is what we term the elimination threshold. As indicated by it's name, the elimination threshold is the minimum level of trust required for an agent to continue to be a part of the delegation group. We also introduce a cycle based elimination threshold. A cycle based elimination threshold means the value of the elimination threshold is not a constant. The value of the elimination threshold is dependant on the number of delegations an agent has accepted. For instance, the value of the threshold for an agent that has accepted only 10 delegations is different from the one that has had 1000 delegations. We provide a simulation in the next chapter to illustrate the concept.

Since elimination of an agent affects the entire group both in terms of delegatee availability and potential elimination, we conceptualize the elimination threshold as a group level attribute. Later in this chapter, we describe a voting based policy wherein the elimination of an agent is initiated by another and other member entities use their autonomy to decide upon the same. When a particular subject finds its trust value of another member agent below the elimination threshold, it initiates the elimination of the latter agent. We represent elimination threshold by the symbol  $\delta_E$ .

*Depreciation of Trust:* Trust is a notion that continuously undergoes change in its value. The reason for this temporal nature of trust is that trust denotes the level of belief

that subjects in a community have towards each other. When these entities have an autonomous and subjective nature as agents in a multiagent system do, the outlook of one agent towards another and hence the corresponding trust values are predisposed to change over time.

Therefore, time is a critical element with respect to trust computation. As described so far, the trust value in our model changes over time due to the outcomes of the delegations that the concerned agents were involved. We also elucidated the role played by the notion of DelHarmony in the computation of trust. Although DelHarmony reflects the length and nature of interactions between any two given agents, it does not capture the time dependant aspect of frequency of these interactions. Consequently, gaps of time interleaving the interactions between two agents go undetected. We contend that these gaps should reflect on the trust value as the performance and benevolence of agents with respect to delegation or with respect to particular entities can undergo a change in the period that interactions cease between the concerned entities.

To capture the aforementioned breaks in interaction and reflect the same in the trust value, we introduce the concept of trust depreciation. Taking the depreciation of trust into account, the formula to update the trust can be given as,

$DT(A,B) = \tau(A,B) \times \text{time units elapsed since the last interaction between A and B.}$  where  $DT(A,B)$  is the depreciation of trust between A and B,  $T(A,B)$  is the current trust value between A and B, and  $\tau(A,B)$  is the rate of depreciation between the agents involved in the computation. Since  $DT(A,B)$  is directly proportional to units of time that have elapsed agent A and B interacted, the value of trust depreciates more with increase in time.

The rate of depreciation  $\tau$  can be defined as the amount by which the value of trust decreases per unit time in the absence of interaction. We compute the value of  $\tau$  as follows,

$$\tau(A,B) = 1 / DH(A,B)$$

where  $DH(A,B)$  is the value of DelHarmony between agents A and B. Therefore, more the harmony between any two agents, lesser is the depreciation of trust between them. This is another formulation where the concept of DelHarmony lends a dynamic approach as an alternative to a more static approach to trust computation. When an agent has had a long and positive interaction history with another agent, the probability of a change in the benevolence of the latter is relatively lesser as compared to an agent with whom the length of the interaction history has been limited.

Once the rate of depreciation and the time that has elapsed since the last interaction is known, the depreciation of trust or  $DT$  can be computed. When the value of  $DT$  is obtained, the current value of trust is updated to reflect the depreciation using,

$$T(A,B) = T(A,B) - DT(A,B)$$

*The Subscriber Model:* The trust updating policy we have proposed thus far is one that is a purely experience based. The subscriber model introduces another unique approach to the trust updating policy. Here, an agent requests another one to enlist it as the latter's subscriber. If the request is accepted, we refer to the former as a subscriber

agent and the latter as the publisher agent. Whenever it updates its own trust value due a new delegation outcome, the publisher agent publishes the importance ‘ $\omega$ ’ of the object of delegation and the result of the delegation. The agents that have subscribed to this particular publisher agent receive this information and in turn update their own trust values based on the same. For instance, consider a publisher agent A and a subscriber agent B. Assume agent A delegates an object to another agent C. Upon completion of the delegation agent A updates its own trust based on the outcome and then ‘publishes’ the importance of the object and the outcome of the delegation. Since agent B is a subscriber of agent A, B receives this information and updates its trust value of agent C to reflect this latest consequence.

Furthermore, an agent requests a publisher agent for subscription only if it has a minimum value of trust towards that agent. We formally term this minimum value as the publisher threshold. The publisher threshold is an agent level parameter. We denote the publisher threshold as  $\delta_p$ . The subscriber model enhances the accuracy of trust without the actual risk of an incomplete or breached delegation through information sharing among trusted entities.

### 3.2 Algorithms

This section describes the algorithms that implement the policies and protocols of the model using the notions that we have defined and described earlier.

*Delegation cycle:* The delegation cycle is the fundamental set of operations that compose our delegation protocol. It starts with a delegation request and concludes with the updating of trust and publishing the results of the delegation. Table 1 lists major steps in our algorithm.

|   |
|---|
| <ol style="list-style-type: none"> <li>1: Compute depreciation of trust and update trust.</li> <li>2: Choose the delegatee.</li> <li>3: Delegate.</li> <li>4: Update the DelHarmony based on result.</li> <li>5: Update the trust value based on the result.</li> <li>6: Publish the result</li> <li>7: Check if updated trust value &lt; elimination threshold<br/>if yes. initiate the elimination</li> </ol> |
|---|

**Table 1.** The basic algorithm for delegation cycle

Step 1 of the algorithm computes the depreciation of trust and then updates the trust value to reflect the same. This process is performed to update the trust values of all the agents that the subject agent has had interactions with, i.e. for whom the subject agent has some trust value to consider.

Step 2 forms the most important part of the cycle. The delegation decision of ‘whom’ to delegate the object to is made during this phase. Here, the agent decides on potential delegates based on the delegation threshold and then chooses a final delegate.

Once the delegation is made in step 3 is completed and the outcome is known, the delegator agent updates her own DelHarmony value to include the result of the latest delegation (see step 4). This is the primary operation of trust updating in our model.

Step 5 includes the computation of the trust incentive or penalty and the actual updating of trust value. The values of incentive or penalty and thus the new modified value of trust are all dependant on the DelHarmony value calculated in the previous step.

The sixth step of the algorithm is part of the subscriber model. Here, the delegator agent publishes the results and the importance of the object that was delegated.

In the final step in the delegation cycle, the delegator agent checksto determine if the updated trust value of her delegatee satisfies the elimination threshold for the given cycle. If not, the delegator initiates the elimination algorithm described next.

In addition to the above steps, all agents that form part of the delegation group update their trust values based on their respective publishers' results.

Elimination cycle: In the previous section, we described the importance of maintaining the benevolence of agents in a group by expelling or eliminating consistently lying agents. We also introduced the notion of a group level elimination threshold,  $\delta_E$ . Elimination cycle is series of steps that is executed by the group as a whole and results in either the agent in question being eliminated or reinstated as a member of the group.

Since all the agents in the group have their individual trust values for all other agents, reaching a consensus on eliminating an agent is non-trivial. Furthermore, the possibility of existence of malicious agents that could cause discord on an otherwise easily obtained consensus complicates the matter. We propose a voting based protocol to solve the aforementioned problem. The steps employed in the protocol are outlined in Table 2 and are executed immediately after initiation. These steps areapplied to all agents in the group except the initiator and the agent to be eliminated.

|   |
|---|
| 1: Decide on the initiation based on the current trust value towards the candidate for elimination.   |
| 2: If no trust has been established with the candidate,<br>Check the trust of the initiator<br>If a high trust value is perceived, vote in affirmative. |
| 3: If no trust is established with the initiator<br>View the published result and vote in accordance to majority.                                       |

**Table 2.** The elimination cycle.

In the algorithm illustrated in Table 2, the initiator is the agent that requests the elimination, and the agent whose elimination is under question is denoted as the *candidate* for elimination.

Each agent first attempts to decide on the initiation based on their own trust value of the concerned agent (or candidate for elimination). In case there is not enough trust established between the voter agent and the agent to be eliminated, the voter agent

predicates the decision on its trust value of the initiator agent. A high value of trust can provide sufficient motivation for the voter agent to confirm the initiation. If here again there exists no dependable basis on which to decide, the voter agent depends on the information that it requests from its publisher agents. If the majority of its publisher agents have affirmed the initiation, the voter agent too confirms the initiation and vice-versa. Here again, the agent explicitly depends on its trust in the publisher agent

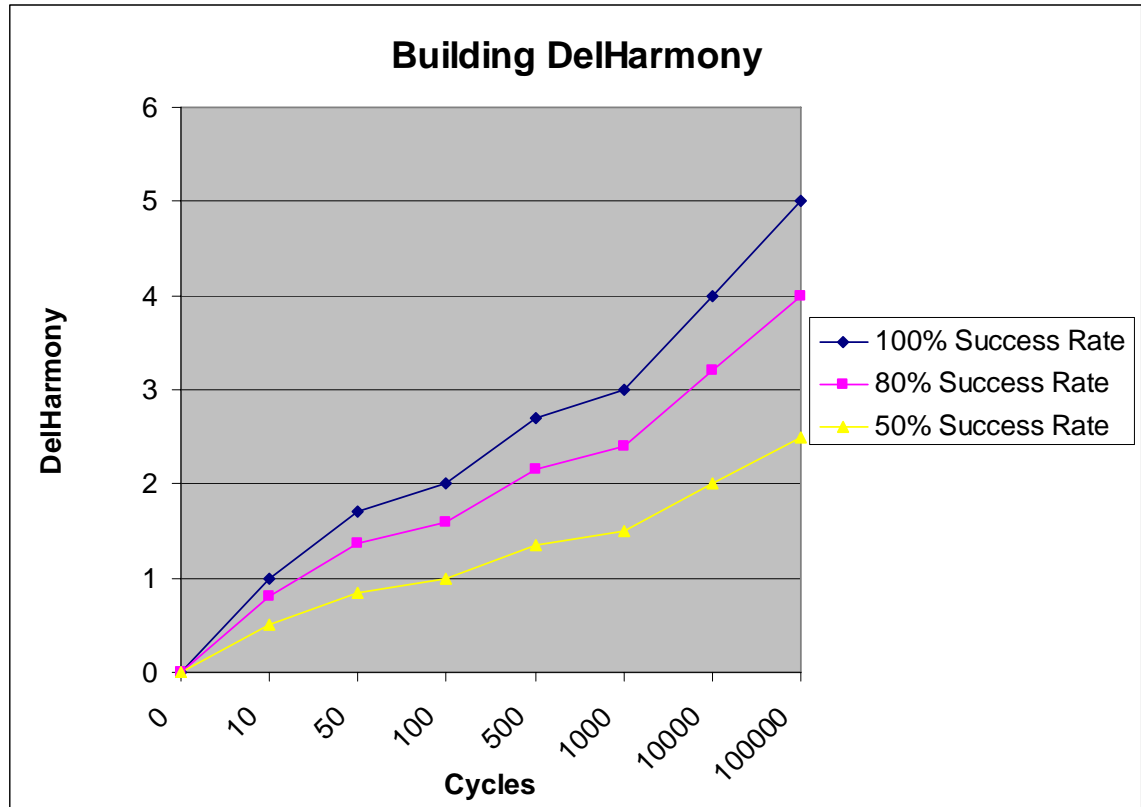
Therefore, the elimination protocol we propose does not entail a mandatory established trust between all the agents forming a part of the delegation group. In the next section we outline validation of our model using a simulation testbed.

## 4. Simulation

In order to illustrate the working of the model we introduced in previous sections a series of simulations are implemented and experiments are reported. The primary aim of our simulations is to illustrate the progression of the values for DelHarmony and trust under different conditions. The simulations were implemented using the Java 1.4 programming language. Values resulting from the simulations were captured and the salient experimental results are reported.

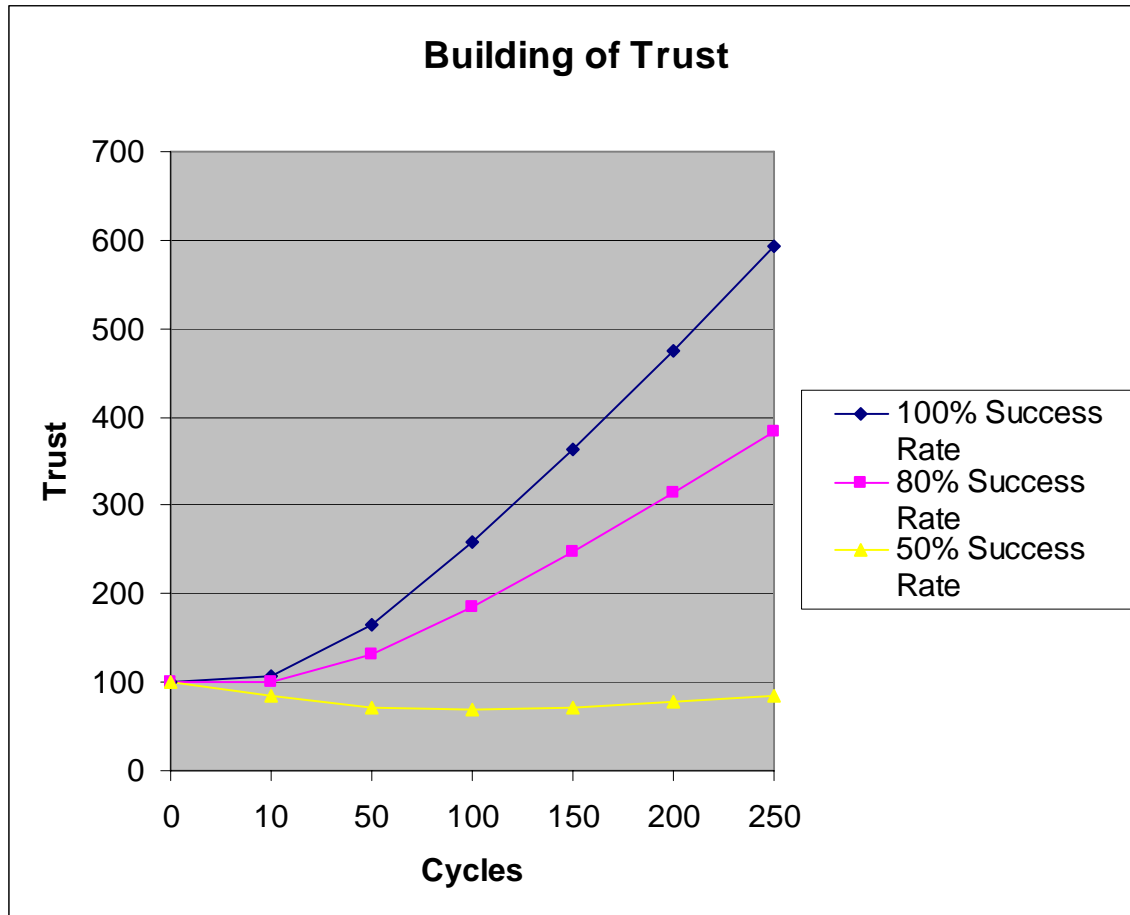
Success Rate: To appreciate the significance of the results of our simulations, it is important to explain the meaning of success rate. We define success rate as the total percentage of delegations that are successful. The success rate is specified at an agent level. For instance, an agent with 100% success rate can be deemed completely reliable and one with a rate of 50% can be expected to succeed in at least half of the total tasks delegated to it. In the experiments that follow, we compare and contrast values of agents with different success rates. The first part of the simulation illustrates the manner in which DelHarmony varies along with rates of delegations in a given group.

We employed three distinct sets of agents working at 100%, 80% and 50% success rates. The success rate is assumed to be constant for the duration of simulation. The simulation is run for a total of 100,000 cycles. Each cycle represents a delegation and agents have the potential to succeed or fail proportional on their respective designated success rates.



**Figure 1.** Building of DelHarmony

As seen in Figure 1, the DelHarmony for the agent with highest success rate is also high over all cycles. The result for the agent with lowest success rate on the other hand is the lowest of the three agents at any given point. Next, our simulation illustrates variation of trust with delegations. Three agents with the same three levels of success rates were employed as in the previous runs of the simulation that were carried out for 250 cycles.



**Figure 2.** Building trust

The results shown in Figure 2 illustrate that the value of trust increases with the increase in success rate. Hence, trust is a good predictor of success rate.

Next, we illustrate the concept of trust depreciation. The simulation was run with three agents maintaining a constant DelHarmony quantity throughout their inactivity periods. Iterations were used to represent time elapsed since the last interaction among agents.

Agents with a high DelHarmony should have their trust depreciate at a slower pace than those with low values of DelHarmony. The reason for this observation is the fact that DelHarmony represents accord and consistency over long periods of interaction among agents concerned. Therefore, an agent with a higher value of DelHarmony would be one that is consistent and consequently, one whose ability and benevolence does not undergo a drastic change in the period of inactivity during which trust depreciates.

The experimental results shown in Figure 3 confirm the fact that the trust for an agent with a higher value of DelHarmony depreciates at a slower rate than that of an agent with a relatively lower value of DelHarmony.

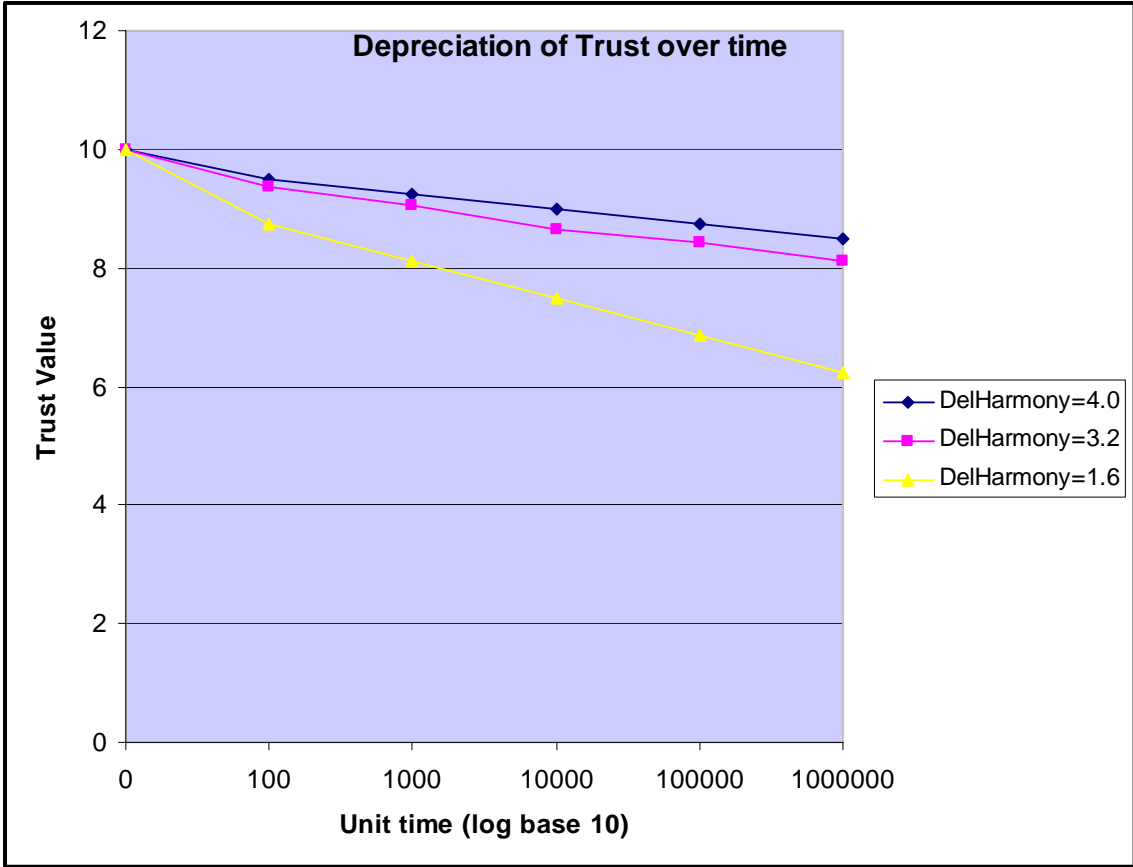
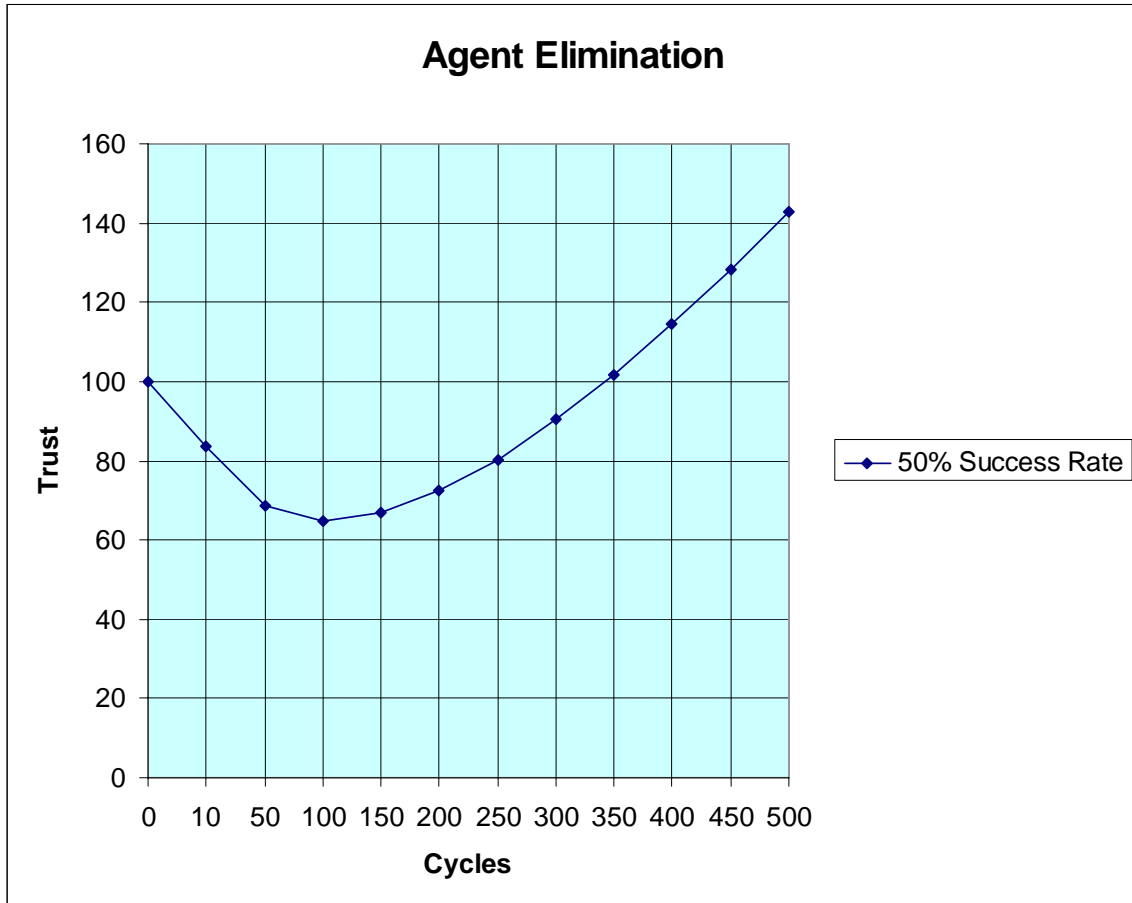


Figure 3. Depreciation of trust

The purpose of the next simulation with results shown in Figures 4 and 5 is to demonstrate agent elimination. The agents in the simulation are assumed to have a success rate that is a constant. The results show the variation of trust for the given agent and the minimum trust (for a given cycle) that ensures that an agent is not eliminated from delegations.



**Figure 4.** Agents at 50% success rate

The two results presented herein illustrate the variation of the trust value over delegation cycles for two agents with success rates of 50% and 20%. The purpose of this simulation is to compare the cycle based elimination threshold that should be adopted by a group. If a minimum success rate of 50% is desired in a group, the values of trust for every cycle in Figure 8 can be used as the threshold. This would eliminate the agent with the 20% success rate at an early cycle. Similar results can be obtained for agents with any given success rate and used as the basis for the cycle based elimination threshold. For instance, if the threshold is based on the agent with a 50% success rate (which has a value of 100 in the 300<sup>th</sup> cycle), an agent failing to maintain a trust value of 100 by the 300<sup>th</sup> delegation would be subject to elimination.

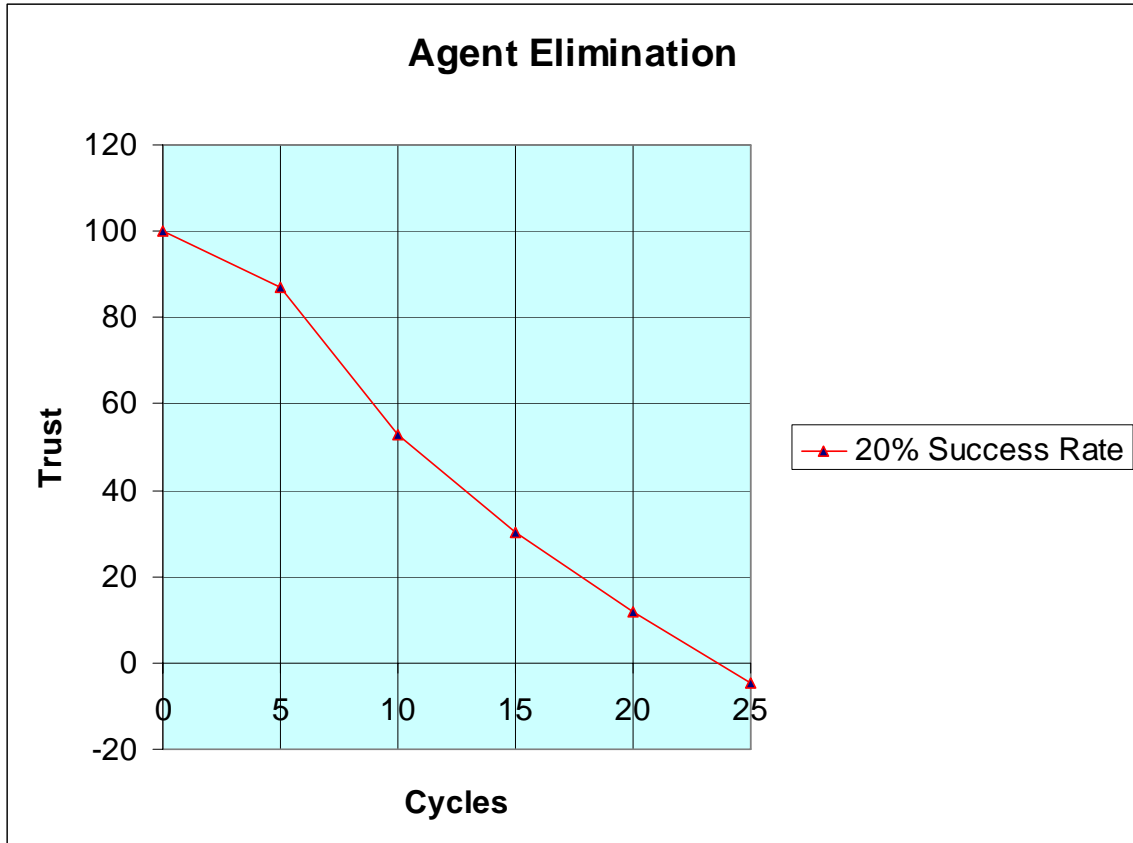


Figure 5. Agent with 20% success rate

## 6. Conclusions

Delegation represents an intuitive approach to collaboration among entities in a heterogeneous system. The open nature of multiagent systems, when coupled with its disparate properties, create challenges to the overall effectiveness of such delegations. Though apparent, the problem is uniquely addressed in the manner we have proposed. Most investigations conducted by the research community thus far have focused entirely on mechanisms involved in delegation. Literature in related fields has recently claimed the importance of a trust based approach to solving decision related problems in delegation (Castelfranchi and Falcone, 2000, Faulkner et. al., 2005, Griffiths, 2005).

By combining the notion of trust with the novel concept of delegation harmony, the model presented herein proposes an innovative approach. We discovered that using DelHarmony, a unique dynamic trust updating techniques can be developed. We posit that these techniques augment the accuracy of trust management. DelHarmony contributes to formulation of trust depreciation.

Our study sheds light on alternatives to experience based trust. The subscriber model described in our work takes a new and improved approach to the recommendation based trust found in majority of literature on trust. Cycle based agent elimination threshold shows how a dynamic approach can be adopted for maintaining a minimum level of trust quality in a given virtual community.

We have brought closer the two chief notions of trust and delegation. Furthermore, research based on this work leads to improvements and refinements resulting in an effective delegation system based on the notion of trust. Although the concepts we have recommended are novel, there remains considerable scope for extending and improving our protocols.

## References

- Abadi, M., Burrows, M., Kaufman, C. and Lampson, B. (1990). "Authentication and Delegation with Smart Cards." *Technical Report 67*, Digital Systems Research Center, October, 1990.
- Abadi M., Burrows, M., Lampson, B., and Plotkin, G. (1991) "A Calculus for Access Control in Distributed Systems." *Technical Report 70*, Digital Systems Research Center, February, 1991.
- Abdul-Rahman, A. and Hailes, S. (2000). "Supporting Trust in Virtual Communities." In *IEEE Proceedings of the Hawaii International Conference on System Sciences*, Maui, Hawaii, January 4-7, 2000.
- Ahsant, M., Basney, J. and Mulmo, O. (2004). "Grid Delegation Protocol", *UK Workshop on Grid Security Experiences*, Oxford.
- Barka, E., Sandhu, R., (2000). "Framework for Role-based Delegation Models." *16th Annual Computer Security Applications Conference (ACSAC'00)*, pg 168.
- Barka, E.; Ravi Sandhu; (2004) "Role-based Delegation Model/Hierarchical Roles (RBDM1)", *20th Annual Computer Security Applications Conference*, 6-10 Dec. 2004 Page(s):396 – 404.
- Beavers, G. and Hexmoor, H. (2003) "Understanding Agent Trust." In *Proceedings of The International Conference on Artificial Intelligence (IC-AI 2003)*. pp: 769-775, 2003.
- Bergenti F., Botelho L. M., Rimassa G., Somacher M. (2002) "A FIPA Compliant Goal Delegation Protocol." In *Proceedings of Workshop on Agent Communication Languages and Conversation Policies*, Autonomous Agents and MultiAgent Systems, Bologna, Italy.
- Castelfranchi, C., (2000) "Engineering Social Order." In *Proceedings of Engineering Societies in Agents' World (ESAW00)*, Berlin, 2000.
- Castelfranchi, C. and Falcone, R. (1997) "From Task Delegation to Role Delegation." In *Proceedings of the 5th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence*.
- Castelfranchi, C. and Falcone, R. (1999). "Towards a Theory of Delegation for Agent-based Systems." *Robotics and Autonomous Systems* 24 , pp: 141-157.
- Castelfranchi C., Falcone R. (2000). "Trust and Control: A Dialectic Link." *Applied Artificial Intelligence Journal*, pp 799-823.
- Castelfranchi, C. and Tan, Y. (2001). "Introduction: Why Trust and Deception are Essential for Virtual Societies." In C. Castelfranchi & Y. Tan (Eds.), *Trust and Deception in Virtual Societies*, Kluwer Academic Publishers.
- Chandran, R. and Hexmoor, H. (2007). "Delegation Protocols Founded on Trust." In the *Proceedings of the Knowledge Intensive Multiagent Systems (KIMAS 07)*, Boston, MA, USA.

- Dasgupta, P. (1998) "Trust as a Commodity." In Gambetta, D. (Ed.), *Trust: Making and Breaking Cooperative Relations*, Oxford: Basil Blackwell, pp 49-72.
- Ding, Y., Horster, P., and Petersen, H. (1996). "A New Approach for Delegation Using Hierarchical Delegation Token." In *Proceedings of the 2<sup>nd</sup> Conference on Computer and Communication Security*.
- Falcone R., Castelfranchi C. (1998). "Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification." *Proceedings of the International Conference on Multi Agent Systems (ICMAS'98)*. pp. 72-79, Paris, France.
- Falcone,R., Castelfranchi,C. (2002). "Tuning the Agent Autonomy: The Relationships Between Trust and Control." In *AAAI-02 Workshop on Autonomy, Delegation, and Control: From Inter-agent to Groups*.
- Faulkner, S., Dehousse, Kolp, M., Mouratidis, H. and Giorgini, P. (2005) "Delegation Mechanisms for Agent Architectural Design." In *Proceedings of the IEEE Intelligent Agent Technology (IAT), IEEE Computer Society Press, 2005*.
- Franklin, S. and Graesser, A., (1996). "Is It an Agent, or Just a Program? A Taxonomy of Autonomous agents." In the *Proceedings of the Third International workshop on Agent Theories, Architectures and Languages*, Springer-Verlag.
- Gambetta, D. (1990). *Trust*, Basil Blackwell, Oxford, UK.
- Gambetta, D. (2000). "Can We Trust Trust?" In Diego Gambetta (Ed.), *Trust: Making and Breaking Cooperative Relations*, Oxford: Basil Blackwell 2000, Chapter 13, pages 213-237.
- Gasser, M., McDermott, E. (1990) "An Architecture for Practical Delegation in a Distributed System." In *Proceedings of IEEE Symposium on Security and Privacy*.
- Griffiths, N., Luck, M., (2003). "Coalition Formation Through Motivation and Trust." In *Proceedings of the Second International Conference on Autonomous Agents and Multi agent Systems (AAMAS-03)*, pp 17-24.
- Griffiths, N. (2005) "Task Delegation using Experience-Based Multi-Dimensional Trust." In *Proceedings of the Fourth International Conference on Autonomous Agents and Multiagent Systems*, 489-496, ACM Press, 2005.
- Hardjono, T., Chikaraishi, T. and Ohta, T. (1993) "Secure Delegation of Tasks in Distributed Systems." In *Proceedings of the 10<sup>th</sup> International Symposium on the TRON Project*, Los Alamitos, California, USA.
- Harris, J. (2002) "On Trust, and Trust in Indian Business: Ethnographic Explorations." London School of Economics and Political Science, London.
- Hu, J. H. (2001) "Some Thoughts on Agent Trust and Delegation." In *Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, Quebec, Canada, 2001, pp 489-496.
- Huynh, T. D., Jennings, N. R., and Shadbolt, S. (2004) "Developing an Integrated Trust and Reputation Model for Open Multiagent Systems." In *Proceedings of the 7<sup>th</sup> International Workshop on Trust in Agent Societies*, pages 65-74, 2004.
- Jennings, N. R., Sycara, K. and Wooldridge, M. (1998) "A Roadmap of Agent Research and Development." *Journal of Autonomous Agents and Multi Agent Systems*, 1:275--306, 1998.
- Low, M and Christianson, B. (1994) "Self-Authenticating Proxies." *Computer Journal*, 37:422-428,

Norman, T. J. and Reed, C. A. (2002). "A Model of Delegation for Multi Agent Systems." In M. d'Inverno, M. M. Luck, M. Fisher & C. Preist (editors), *Foundations and Applications of Multi Agent Systems*, volume 2403 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pages 185-204.

Ramchurn, S. D., Sierra, C., Godo, L., and Jennings, N. R. (2003) "A Computational Trust Model for Multiagent Interactions Based on Confidence and Reputation." In *Proceedings of the 6<sup>th</sup> International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69-75, 2003.

Ring, P.S., (1996). "Fragile and Resilient Trust and Their Role in Economic Exchange." In *Business and Sociology* 35, pages 148-175.

Sabater, J., Sierra, C. (2002). "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.

Sabater, J., Sierra, C. (2005). "Review on Computational Trust and Reputation Models." In *Artificial Intelligence Review*(24). pp 33-60.

Schillo, M. (1999). "Trust and Deceit in Multiagent Systems." *Diploma Thesis, Department of Computer Science, Saarland University, Germany.*

Shapiro, D., Sheppard, B.H., Cheraskin, L., 1992. "Business on a Handshake." *Negotiation Journal* 8(4), pages 365-377.

Silverman, B., Johns, M., Weaver, R., O'Brien, K., and Silverman, R., (2001). "Human Behavior Models for Game-Theoretic Agents: Case of Crowd Tipping." In *10th Conference on Computer SISO*, 2001.

Sollins, K. (1988). "Cascaded Authentication." In *Proceedings of IEEE Conference on Security and Privacy*.

Sycara, K., Paolucci, M., Van Velsen, M., Giampapa, J. (2003) "The RETSINA MAS Infrastructure." In *Proceedings of Autonomous Agents and Multi Agent Systems*, Volume 7, 2003, pp 29-48.

Tamassia, R., Yao, D. and Winsborough, W. H. (2004) "Role-based Cascaded Delegation." In *Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT '04)*, pages 146 – 155, ACM Press.

Thompson, C. (2004) "Agents, Grids and Middleware." In *Architectural Perspective Column, IEEE Internet Computing*, September/October 2004.

Tian-chi, M. and Shan-ping, L. (2005) "Instance-Oriented Delegation: A Solution for Providing Security to Grid-based Mobile Agent Middleware." In *Journal of Zhejiang university*. Vol 6A, No. 5, pp 405-413.

Wong, H. C., Sycara, K., (1999) "Adding Security and Trust to Multi Agent Systems." In *Proceedings of Autonomous Agents'99 (Workshop on Deception, Fraud and Trust in Agent Societies)*.

Ye, C., Fu, Y. and Wu, Z. (2004). "An Attribute-Based-Delegation-model." In the *Proceedings of the 3<sup>rd</sup> International Conference on Information Security, ACM International Conference Proceedings Series: Vol. 85*.

Yu, B. and Singh, M. P. (2002) "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.