Volume 1, No. 4, Nov-Dec 2010
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

An Approach to find Trustworthiness among Different Domains in a Grid Environment

Dolly Sharma*
Deptt. of IT, BUEST
Baddi, HP, India
dollysharma83@ymail.com

Sarbjeet Singh
CSE, UIET, Panjab University
Chandigarh, India
sarbjeet@pu.ac.in

Seema
Dept of CSE, Amity Unversity,
Gurgaon, Haryana, India
mudgil.seema@gmail.com

Abstract: The goal of Grid computing is to create illusion of a simple yet large and powerful self managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. Such an environment introduces challenging trust related issues as both service providers and users can come from mutually distrusted administrative domains and any of them can behave maliciously. The use of trust evaluation simplifies the security architecture and is evaluated on the basis of a number of parameters like trust decay, reputation, trust updation, transitivity etc. A number of models have been proposed by different researchers for the evaluation of trust but many of them have missed one or the another required parameters that are necessary for the evaluation of trust in a comprehensive way. In this paper a novel approach for evaluation of trust has been proposed that insists on the use of a number of imnportant parameters to calculate trust in a comprehensive way.

Keywords: Grid computing, trust, reputation, trust model, feedback, trust evaluation, trust decay.

I. INTRODUCTION

Grid technology brings together a set of resources distributed over wide-area networks and supports large-scale distributed applications by coordinating resource sharing and problem solving in dynamic, multi-institutional, virtual organizations [1]. Security requirements are fundamental to the grid design [2]. Rasmusson and Jansson [3] categorized security as hard security and soft security. Hard security is achieved through cryptographic mechanisms, encryption techniques etc. But it overshadows the essence of soft computing that the resources can be accessed directly. Integrating trust in grids is one of the ways to achieve soft security in grid environment.

Applying trust to grid computing provides a mechanism for entities to manage risk arising due to interactions taking place between different entities. Trust is a social phenomenon, and can be defined as a firm belief in the competence of an entity to behave as expected such that this firm belief is a dynamic value associated with the entity and is subject to the entity's behavior and applies only within a specific context at a given time [4].

A lot of attempts to evaluate trust in the field of distributed systems have been initiated. Important among them include [4, 5, 6, 7, 8, 9]. A survey of these trust models has been presented in [10]. Some of these models lack strong mathematical foundations whereas others have missed out one or the other required parameters for trust evaluation. Based on the literature survey a list essential trust related parameters has been identified. These parameters have been described in Section III. Finally a trust model has been proposed in section IV that insists on the use of all the trust related parameters identified in section III to calculate trust in a comprehensive way.

II. RELATED WORK

A number of models have been proposed by different researchers for the evaluation of trust in grid. A summary of classification of these models has been given in Table II.

A. Abdul-Rahman and S. Hailes proposed a Trust-Reputation Model [1] based on trust characteristics from social sciences. Trust is context-dependent and based on prior experiences. Trust supports negative and positive degrees of belief of an agent's trustworthiness. Trust is not transitive but subjective, dynamic and nonmonotonic. Trustworthiness is evaluated on the basis of experiences and reputation. An experience results from direct interaction. A reputation is an expectation about past behavior of an agent and is calculated from a trusted set of recommenders. F. Azzedin and Muthucumaru proposed a Trust model for Grid Computing Systems [4] which is an extension to [1] and [11]. They insist that direct trust weighs more than recommender trust. The model also lets a newcomer to build its trust from scratch by enforcing enhanced security. Here trust is dynamic, context specific, based on past experiences and spans over a set of values ranging from very trustworthy to very untrustworthy. Trust is evaluated on the basis of direct trust and reputation. A Recommender trust factor is introduced to prevent cheating via collusions among a group of domains. Farag Azzedin and Muthucumaru proposed a Trust Model [7] for peer to peer computing systems also. In [4], an accuracy measure has been associated with each

recommendation. Accuracy of each recommendation is difference between the recommendation provided by for entity and the true trust level of that entity. All the models discussed above view trust as a one dimensional quantity having value between 0 and 1. Chin Lin, V. Varadharajan, Yan and V. Paruthi proposed a Trust Management Architecture [6] for enhancing grid security that explored the three dimensional view of trust which includes belief, disbelief and uncertainty. This subjective logic based trust evaluation is based on Dempster-Shafer theory [12]. None of the above models includ risk assessment in Trust evaluation. Z.Liang and W.Shi proposed a PErsonalized Trust model [9] for peer-to peer resource sharing. Only PET has accommodated risk asses sment which has been done to perceive the suddenly spoiling peer.

III. TRUST SYSTEM PARAMETERS AND CLASSIFICATION

Based on the literature survey, a list of parameters essential in a trust model has been identified. Table II shows the list of these parameters.

A. Trust and Reputation

Trust has been defined in [7] as the firm belief in the competence of an entity to act as expected such that this firm belief is not a fixed value associated with the entity but rather it is subject to the entity's behavior and applies only within a specific context at a given time [7]. Trust has several characteristics. Trust is subjective, dynamic, context dependent and nonmonotonic [1]. It decays with time and can be classified into different levels. Trust broadly includes security, safety, reliability, timeliness, and maintainability.

Trust may be evaluated on the basis of identity trust, behavior trust or both. The reputation of an entity is an expectation of its behavior based on other entities' observation or information about the entity's past behavior within a specific context of a given time. If direct trust does not exist between two entities, then reputation is the only way to determine trustworthiness between those entities.

B. Trust Asymmetry

If A $\longrightarrow$ B then,

Does B $\longrightarrow$ A ?

where A, B are entities and $\rightarrow$ denotes Trust.

The answer is not necessarily yes! This situation is called Trust asymmetry problem. The solution to this problem is Trust symmetry [6]. The user positions itself as the resource provider host to estimate trust on the user from user's point of view, i.e. to evaluate the *trust reflection*.

C. Historical Accumulation of Past Behavior

In [4, 5, 6, 7, 9], an entity stores the evaluated trust value, for future use, after interaction. Each entity maintains a data structure for direct trust (TB) as well as reputation (RB). Initially both are empty.

D. Weightage of Identity Trust and Reputation

Each entity trusts itself, so it trusts its own belief about some other entity, more than any other entity's belief about that entity. So an entity for which both direct trust and recommendations exist, direct trust is given a higher weightage than recommendation [4].

E. Trust Level

Trust may be categorized into levels [3, 15, 5]. For example, in our proposed model, Trust ranges from *ht* to *hu* as shown in Table I. Trust level *et* is not provided by any existing trust relationship.

Table I. Levels of Trust

Trust level	Significance
et	Extremely high trustworthy
ht	Highly trustworthy
t	trustworthy
u	untrustworthy
hu	Highly untrustworthy

F. Trust Inheritance

In a distributed environment, entities can join or leave a Virtual Organization at anytime. When an entity joins a domain, it inherits the recommendation trust table [4]. However, other domains might not trust this new entity to be as trustworthy, so, a member weight is associated with every entity to indicate if the entity is a new, recent or an old member with its domain [4].

G. Evolving Trust as a Newcomer

As a new entity joins a domain it has no trust relationships. Its data structures are initially empty. Each domain that wants to interact sets a required trust level (RTL) for its entities. So the RTL for a newcomer is set as the highest trust level so that the newcomer will make initial relationships only with highly trustworthy domains [4].

H. Trust Threshold

Trust threshold can be defined as the minimum value of trust that is required to establish a trust relationship between entities. In some models trust lies between 0 and 1 and threshold is 0.5. Some trust systems return relative ranking [5].

I. False Recommendation

Reputation is calculated from recommendations from several entities. Any entity can give malicious or fraudulent recommendation either intentionally or unintentionally. All these problems can be solved by aggregation of all the recommendations received for any entity [4, 5, 7, 9, 13].

J. Aggregation of Recommendations

Recommendations are provided by a number of recommenders so these are aggregated to solve the Recommendation network. Models that see trust as a single dimensional quantity generally use average of recommendations to calculate reputation. Other models that use three dimensional trust use Dempster Shafer rule for aggregations [6].

K. Transitive Recommendations

If entity A trusts entity B and B trusts entity C then A may also trust C. This is called transitivity.

L. Trust Decay and Reputation Decay

The positive effect of successful interaction on trust will reduce over time as will the negative effect of unsuccessful

interaction. So, a decay function is applied regardless of whether the trust value represents trust or distrust [4].

Table II. List of parameters andComparison of existing models

Classification Parameter	[5]	[4]	[7]	[8]	[13]	[6]	[9]	Proposed model
Identity Trust	yes	yes	yes	yes		yes		yes
Reputation	yes	yes	yes	yes		yes	yes	yes
Trust Asymmetry	yes	yes	yes					yes
Historical accumulation of past behavior	yes	yes	yes	yes	yes	yes	yes	yes
Weightage of identity trust & reputation		yes	yes				yes	yes
Trust levels	yes	yes	yes	yes				yes
Inheritance by new joining entity		yes						yes
Evolving trust as newcomer		yes	yes	yes	yes			yes
Trust threshold	yes	yes	yes	yes	yes	yes	yes	yes
False recommendation	yes	yes	yes	yes			yes	yes
Recommendation aggregation		yes	yes	yes			yes	yes
Transitive Recommendation	yes		yes			yes		yes
Trust Decay		yes	yes		yes		yes	yes
Reputation Decay		yes	yes				yes	yes
Feedback	yes	yes	yes	yes	yes	yes		yes
Intrusion detection	yes	yes	yes					yes
Trust & reputation updation	yes	yes	yes	yes		yes		yes
Risk assessment							yes	yes
Trust View	1-D	1-D	1-D	3-D	1-D	1-D	1-D	3-D

M. Feedback

After trust evaluation, entity decides to interact with a chosen entity. After the interaction, a feedback is taken, according to which the trust level in Trust base (TB) and Reputation base (RB) is updated. This feedback may be positive, negative or neutral. If the feedback is positive the trust level is raised and correspondingly if the feedback is negative, the trust level is lowered [5].

N. Intrusion Detection and Audit Trials Analysis

Intrusion Detection is a process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible violations of security policies [14]. For detecting authorized but abusive user activity, audit trails are an appropriate means. Audit trails establish accountability of users for their actions and provide evidence to establish the guilt or innocence of suspected individuals [15]. The ultimate goal of any Intrusion Detection System (IDS) is to obtain as high detection rate as possible and as low false alarm rate as possible [14]. These are helpful in finding the feedback.

O. Trust Update & Reputation Update

After the interaction, intrusion detection [6] and audit trials analysis is done. According to this feedback, TB and RB are updated [4, 5, 7]. The recommenders who gave recommendation about that entity also update their RB in a similar manner. Further, penalties may be imposed on the defaulters.

P. Risk Assessment

Any entity may provide bad services, which may be ue to non-subjective factors. Bad service here implies that the service provided by the entity was not as satisfactory as was promised at the time of interaction. On the behalf of feedback, the actual trust level is calculated which is different from the expected trust level. This accounts for the Quality of Service provided by the entity. This service quality can also be used to calculate risk [9].

Q. Trust View

Some models consider trust as a single dimensional quantity. Other models have explored the three dimensional view of trust based on Dempster Shafer theory [12].

IV. PROPOSED TRUST MODEL

The proposed trust model computes trust based on combination of direct trust (ß), reputation (ж) and Risk (§). For this, the components Direct Trust Fetcher, Reputation collector engine and Risk evaluator respectively are used. They are shown in Trust evaluation cycle UML2 activity diagram in figure 1 and figure 2. Direct trust is fetched from entity's Trust Base (TB) which is a table that store trust values resulted from direct interaction with another entity. For a new entity, having no previous interactions, this TB is empty. In that case trust is evaluated on the basis of entity's reputation and the direct trust component is zero. Reputation is the summarized value of recommendations taken from different sets of recommenders. Recommendation is stored in entity's Reputation Base (RB). The third component for trust evaluation is Risk assessment, which considers the fact that any resource provider may provide unsatisfactory services for a small interval of time and then continue to provide satisfactory services. This kind of behavior can be taken into account in order to determine quality of service. The resultant trust values from Direct Trust Fetcher,

Reputation collector engine and Risk evaluator are given as input to the Trust Evaluator as shown in figure 1. Trust Evaluator evaluates total trust on the basis of its input.

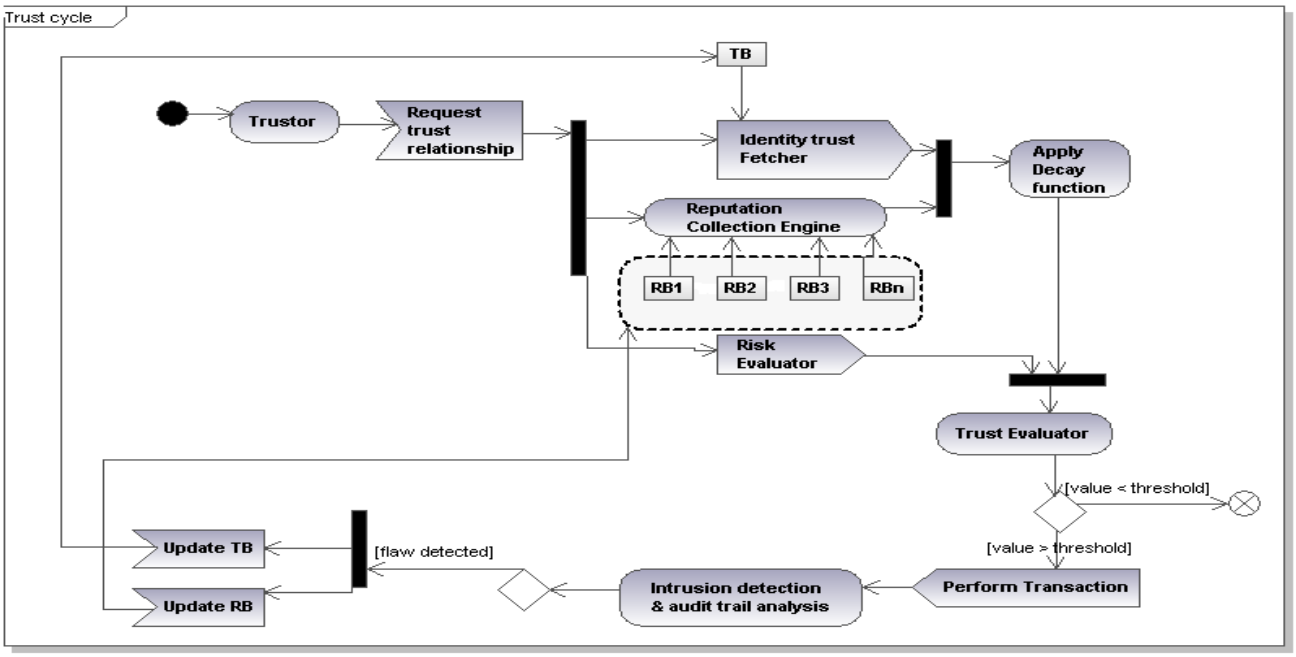

Figure 1 Trust evaluation cycle UML2 activity

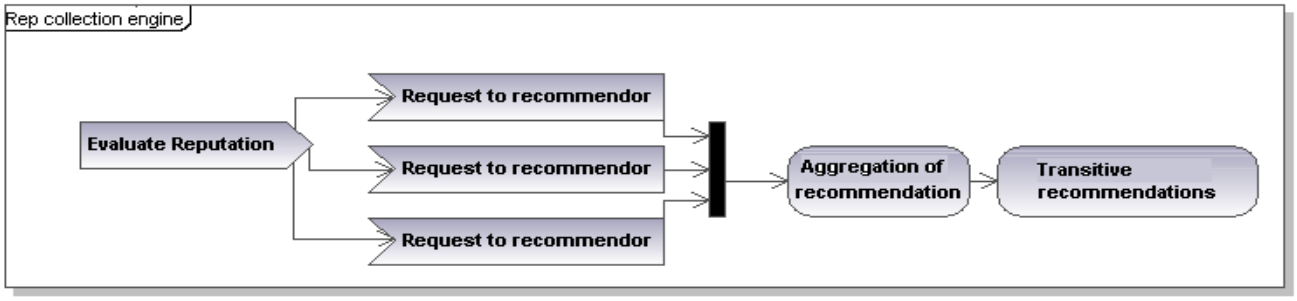

Figure 2. Reputation evaluation UML2 activity diagram

Weights are assigned to direct trust, reputation and risk so that the resultant trust lies between 0 and 1. Weightage of direct trust is always more than reputation. Weightage of reputation and risk depends on the quality of required service.

According to this model, trust is a based on subjective logic [9] and is represented by a triplet (B(x), P(x), Q(x)). These triplets are stored in Trust Bases and Reputation Bases for future use. Here B(x) stands for belief which is the probability that an entity x can be trusted, P(x) stands plausibility which is the probability that an entity x cannot be trusted and Q(x) stands for commonality which accounts for uncertainty and fills the void in the absence of both belief and disbelief. All the three functions have value between 0 and 1.

$$0 \leq B(x) \leq 1 \quad (3)$$

$$0 \leq P(x) \leq 1 \quad (4)$$

$$0 \leq Q(x) \leq 1 \quad (5)$$

In general, $B(x) + P(x) \leq 1$. All the three functions are linked together such that the sum of these is 1.

$$B(x) + P(x) + Q(x) = 1 \quad \text{or} \quad (6)$$

$$Q(x) = 1 - B(x) - P(x) \quad (7)$$

The difference, Q(x) function accounts for uncertainty. As Dempster-Shafer theory [9] deals explicitly with uncertainty, so this component of trust triplet is very important while combining trust values. Trust values can be aggregated as well as transitively combined. Trust is context sensitive.

When an entity (Trustor) requests to interact with another entity (Trustee) for a particular context, trust is evaluated as shown in figure 1. This trust consists of direct trust, reputation and risk. Direct trust fetcher fetches trust value from Trust Base (TB). For calculating reputation, entity asks its set of recommenders for giving recommendations. This recommendation is fetched from each recommender's RB. Whether those recommenders have an entry in their RB or not, they further ask their set of recommenders to give recommendations and the process continues thereby forming a chain of recommendations. These chains form a recommendation network which is solved by both aggregation of recommendations and transitive recommendations. A reputation evaluation UML2 activity diagram is shown in figure 3. As trust and recommendations are 3-D quantities therefore, recommendations are aggregated using a modified form of

Dempster Shafer rule for combining independent elements of belief.

```
Procedure EvaluateTrust(Entity,Context)
{
call Evaluate_Identity_Trust();
// It returns Identity_Trust_After_Decay

call Evaluate_Recomendation();
// It returns Recommended_Trust

call Evaluate_Risk();
// It returns Risk_Trust

If <Identity_Trust_After_Decay != NULL>
{
   // Calculate total trust by applying weights to
   Identity_Trust_After_Decay,    Recommended_Trust, Risk_Trust

   Set Total_Trust = a * Identity_Trust_After_Decay  +
                     b *  Recommended_Trust +
                     c *  Risk_Trust;
  // a is always greater than b and a+b+c = 1
}
else
{
   //Calculate Total_Trust by taking a=0
   Set Total_Trust =  b *  Recommended_Trust + c *  Risk_Trust;
}

//Compare Evaluated_Trust with Specified Trust_Threshold
if < Total_Trust <= RTL>
{
   Insert Total_Trust , Time_of_Interaction , Entities , Context into
Risk_List
   call Update_trust();
   return Total_Trust;
}
else
{
   // Trust relation cannot be formed
   return NULL;
}
```

Figure 3. Trust Evaluation Algorithm

```
procedure Evaluate_Identity_Trust()
{
 Fetch Identity_Trust from Trust Base for a    particular Entity and
 Context

 Apply decay to Identity_Trust resulting in
 Identity_Trust_After_Decay

 Return Identity_Trust_After_Decay;
}
```

Figure 4. Identity Trust Evaluation Algorithm

$$B_x \Xi B_y = \begin{cases} B(x) = \dfrac{B(x)\cdot B(y) + Q(x)\cdot B(y) + B(x)\cdot Q(y)}{k - B(x)\cdot P(y) - P(x)\cdot B(y)} \\ P(x) = \dfrac{P(x)\cdot P(y) + Q(x)\cdot P(y) + P(x)\cdot Q(y)}{k - B(x)\cdot P(y) - P(x)\cdot B(y)} \\ Q(x) = \dfrac{Q(x)\cdot Q(y)}{k - B(x)\cdot P(y) - P(x)\cdot B(y)} \end{cases}$$

$$\text{where } k = \begin{cases} (B(x) + P(x) + Q(x)) \cdot (B(y) + P(y) + Q(y)), \; B(x) + (x) + Q(x) < 1, \\ B(y) + P(y) + Q(y) < 1 \\ 1, B(x) + P(x) + Q(x) = 1, \\ B(y) + P(y) + Q(y) = 1 \end{cases} \quad (8)$$

The factor 'k' is used to make the resultant trust in range 0 to 1, in case the trust values B_x and B_y are not in the range 0 to 1.

Transitive recommendations are combined by Recommendation Combination [16]. Suppose $R_x(y)$ denote the trust of entity $€_x$ on entity $€_y$ and $R_y(z)$ denotes the trust of entity $€_y$ on $€_z$. The trust of entity $€_x$ on entity $€_z$ is calculated by the Recommendation rule, $R_x(z) = R_x(y) \otimes R_y(z)$ where,

$$B_x(z) = B_x(y)\cdot B_y(z) \quad (9)$$

$$P_x(z) = P_x(y)\cdot P_y(z) \quad (10)$$

$$Q_x(z) = P_x(y) + Q_x(y) + B_x(y)\cdot Q_y(z) \quad (11)$$

Here, $B_x(y)$, $P_x(y)$, $Q_x(y)$ is the belief, plausibility and commonality respectively of entity x on entity y. $B_y(z)$, $P_y(z)$, $Q_y(z)$ is the belief, plausibility and commonality respectively of entity y on entity z. $B_x(z)$, $P_x(z)$, $Q_x(z)$ is the belief, plausibility and commonality respectively of entity x on entity z.

Risk is evaluated by taking the ratios of all bad services received in a given time interval over the worst services received in this interval.

After the evaluation of identity trust, reputation and risk, a decay function $\zeta(t_o,c)$ is applied to both identity trust as well as reputation, regardless of whether the trust value represents trust or distrust.

$$ß(Tr,Te,c,Ŕ,t) = TB(Tr,Te,c) \; \mathbf{X} \; \zeta(t_o,c) \quad (12)$$

$$ж(Te,c,Ŕ,t) = RB(Tr,Te,c) \; \mathbf{X} \; \zeta(t_o,c) \quad (13)$$

where,

$$t_o = t_{current} - t_{interaction} \quad (14)$$

```
procedure Evaluate_Recommendation(E)
{
//Find set of  Recomenders for Entity E
Recommender_Set = {R1, R2, ….. Rn};
for each Recommender Ri in Recomender_Set{ R1, R2, ….. Rn }
whose Ri.flag=false
//(false for those entities that have not given //recommendation yet)
   {
    Fetch Recommendation from Ri into Recommender_Trust
    Set Ri.flag= True;
  //call Evaluate_Recommendation(E) module of Entity Ri
    call Ri.Evaluate_Recommendation(Ri);
  //This module will return aggregated //recommendations from
  Entity Ri.
   }
//call Aggregate_Recommendations by passing all //recommender''s
Trust.
  call Aggregate_Recommendations(Recomender_Set);
// It returns Recommendation_Aggregated_Result
 return recommended_Trust;
}
```

Figure 5 Recommendation Trust Evaluation Algorithm

```
procedure aggregate_recommendation (recommender_set)
{
find number of recommenders n from recommender_set
//aggregate all the recommendations
 while(i<=n)
 {
   aggregate r_i and r_i+1 resulting in R where
   k = (ri.b + ri.p + ri.q) * (r_i+1.b + ri+1.p + ri+1.q)

   R.b = (ri.b * r_i+1.b + ri.q * r_i+1.b + r_i.b * r_i+1.q) /
    (k - r_i.b * r_i+1.p - r_i.p * r_i+1.b)

   R.p = (ri.p * r_i+1.p + ri.q * r_i+1.p + r_i.p * r_i+1.q) /
    (k - r_i.b * r_i+1.p - r_i.p * r_i+1.b)

   R.q = (ri.q * r_i+1.q) / (k - r_i.b * r_i+1.p - r_i.p * r_i+1.b)
   Now aggregate R and r_i+2 and so on…
 }
return aggregated_trust;
}
```

Figure 6 Recommendation Aggregation Algorithm

The time when last interaction took place, should be subtracted from the current time [4]. Here, t_o is the difference between the current time and the time of last

interaction and c is the context for which the entities want to interact.

Three of these evaluated components are fed to the Trust Evaluator Engine after applying a decay function to identity trust and reputation as shown in figure 1. This engine evaluates trust by combining these components and applying weights to them.

$$\text{ə}(Tr, Te, c, \acute{I}, t) = a \mathbf{X} \text{ ß}(Tr,Te,c,\acute{I},t) + b \mathbf{X} \text{ ж}(Te,c,\acute{I},t) + c \mathbf{X} \text{ §}(Tr,Te,c,t) \quad (15)$$

$$a+b+c=1,\ 0\leq a \leq 1,\ 0\leq b \leq 1,\ 0\leq c \leq 1,\ a>b$$

```
procedure Evaluate_Risk(E_Id, Context)
{
 Fetch Trust from Risk_List where Entity_Id = E_Id   and
 Cntxt=Context
 //  These trust values resulted from poor //behavior of that entity

  Calculate Risk_Trust by averaging the fetched Trust values.
 Return Risk_Trust;
}
```

Figure 7. Risk Evaluation Algorithm

```
procedure update_trust()
{
 for each recommender r_i whose r_i.flag = true
 {
   Update Trust Base and Reputation Bases        with evaluated
   trust value
 }
}
```

Figure 8. Trust Update Algorithm

where ə(Tr, Te, c, Í, t) is the total evaluated trust of Trustor Tr on Trustee Te for context c, at time t with trust level Í. Similarly ß is the Identity trust, ж is the Reputation of Trustee Te and § is the Risk. Here a,b,c are the weights assigned to Identity trust, Reputation and Risk respectively.

The resultant trust value ə(Tr, Te, c, Í, t), is compared with the trust threshold. For comparison an Trust comparison operator ($\geq op$) [16] is used. Trust value ə(Tr1,Te1,c,Í, t) = (B(Tr1),P(Tr1),Q(Tr1)) is over the trust threshold RTL(B(Th),P(Th),Q(Th)) i.e.

$$\text{ə}(Tr1, Te1, c, \acute{I}, t) \geq op \text{ RTL}(B(Th),P(Th),Q(Th)) \quad (16)$$

if B(Tr1) > B(Th); P(Tr1) < P(Th) and Q(Tr1) < Q(Th). And we say that opinion *OA* is over a threshold presented by *OB*.

If the resultant trust is greater than threshold, transaction is performed. After that intrusion detection tool or audit trial analysis tool is used to evaluate feedback. The trust bases and reputation bases are updated according to positive or negative feedback.

V. RESULTS AND DISCUSSIONS

A Grid Environment is set up using ASP.NET with C# Web Services. A few entities are created, out of which some are known to perform good and others are known to perform bad. Database is created in Microsoft SQL Server 2000. The proposed trust algorithm has been tested against various conditions. The weightage of trust functions, Identity Trust, Recommendation Trust and Risk Trust etc, are varied. Trust model is analysed without considering the Risk component and then compared with the Model that includes risk component. The performance of Trust model is measured in terms of number of interactions that took place. The number of good known entities is gradually increased to measure performance. Similarly, the number of bad known entities is also gradually increased. Trust model is also tested using decay component and without using decay component.

A. Performance with increasing Good known entities

Few of the entities are known to be good. They give higher trust values before interaction and positive feedback after interaction. Weightage of trust functions is varied as shown in Table III.

Here a, b and c is the weightage given to Identity Trust, Recommendation and Risk Trust respectively. Interaction can take place only when the evaluated trust value is greater than trust threshold. Trust threshold has been taken as 0.7.

Table III Performance with increasing Good known entities

S No	Weightage of Trust Functions		Description
1	E1	a = 0.7 , b = 0.2, c = 0.1	Emphasizing Identity Trust
2	E2	a = 0.3 , b = 0.5, c = 0.2	Emphasizing Recommendation Trust
3	E3	a = 0.3 , b = 0.2, c = 0.5	Emphasizing Risk Trust and Identity Trust
1	Trust Threshold = 0.7		Minimum Required Trust level for interaction

As shown in figure 9, the number of good known entities is increased gradually. As the number of good known entities increases, number of recommenders increase. With the increasing number of recommendations, trust value increases raising the number of interactions. Three different weightage conditions are considered. Results obtained are categorized as:

(a) *Emphasizing Identity trust :*Emphasizing Identity Trust gives almost constant number of interactions even if the number of good known entities increases. Trust value

comes majorly from Trust Base, so number of increasing recommenders' donot changes the trust value.

(b) *Emphasizing Recommendation trust:* Recommendation plays a vital role in evaluating trust value. As the number of good known entities increases, number of interactions also increases.

(c) *Emphasizing Risk Trust and Identity Trust:* Emphasizing Risk and Identity Trust leads to lesser number of interactions as compared to when recommendations are emphasized. Risk assessment leads to detection of bad behavior of the entity. It accounts for the trust values when the entity behaved poorer then expected.

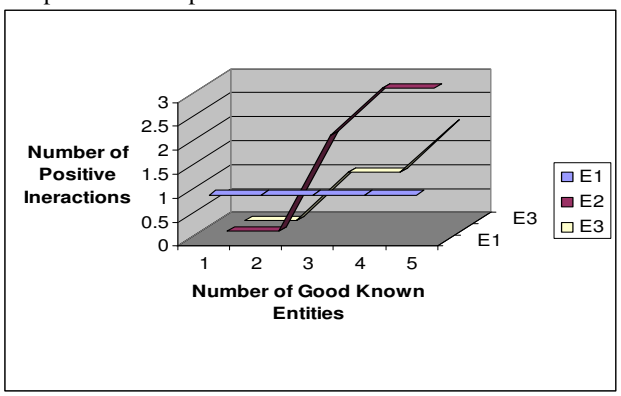

Figure 9 Performance of Trust Model with increasing Good known entities

B. Performance with increasing bad known entities

As the number of bad known entities increases, the evaluated trust level decreases. Bad entities are those that produce a minimal trust level. They may also fail to produce positive feedback. As the number of entities increases, number of recommenders also increases. These recommenders produce lesser value of recommendation trust because the entity is bad. So the total trust value decreases as shown in figure 10.

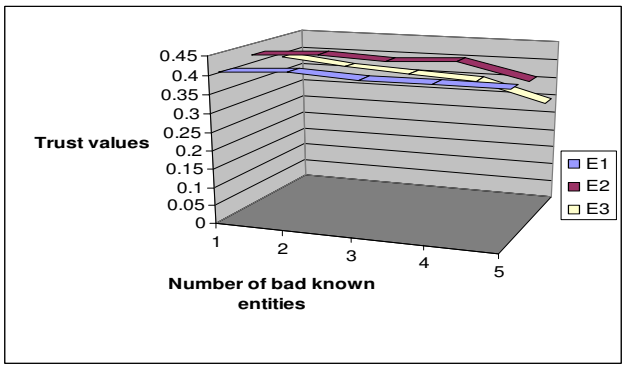

Figure 10 Performance of Trust Model with increasing Bad known entities

C. Performance using Risk Assessment

Trust model has been evaluated for number of cooperations while considering risk and without considering risk. The trust level of an entity is lowered when its past bad behavior (if any) is taken into account as shown in figure 11. So lesser number of entities qualifies the trust threshold test, thereby, decreasing the number of positive interactions. Considering risk trust while evaluating total trust avoids the danger of interacting with any entity that behaves at times bad**.**

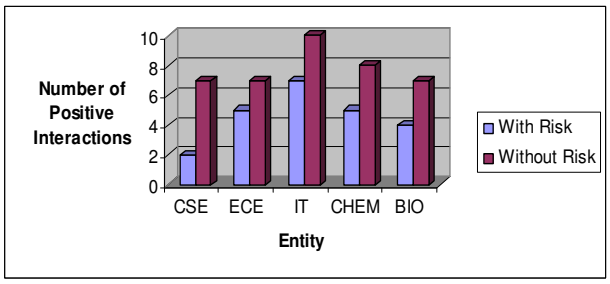

Figure 11 Performance of Trust Model using Risk Assessment

D. Performance using different weightages of Trust functions

The adaptability of the Trust model is tested in different scenarios using different weightage criteria of trust functions. The weightage criteria are I1, I2, I3, I4 and I5 as shown in Table IV. Identity trust is the major focus in I1. Here recommendations are given very less importance and risk is almost ignored. So interaction takes place only with the previously interacted entities. This is the situation where recommendations are not trusted much and it is believed that entities give almost similar feedback whenever they are interacted with.

In second scenario I2, entities are believed not to give a similar behavior every time they are interacted with. So sometimes the entities behave badly. In this situation weightage of Risk Trust is kept more. The resultant trust value is lowered for the entities that have previously behaved badly. The average number of interactions that take place is therefore less.

In the third scenario I3, risk is ignored. It is believed here that entities perform almost same every time they are interacted with. Identity trust and Reputation accounts for the resultant trust. This scenario produces the maximum number of interactions.

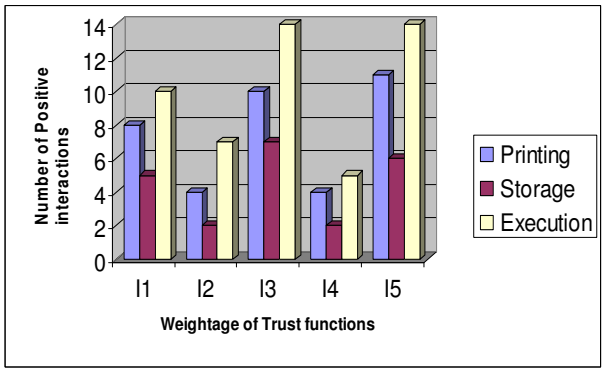

Figure 12 Performance using different weightages of Trust functions

In fourth scenario I4, Identity trust, Risk trust is considered and reputation is ignored. In this case recommendations from other entities cannot be trusted, so they are ignored. Entities may perform badly in some cases so risk trust is considered. Because of not considering the

Table IV Weightage of Trust functions

S No	Weightage of Trust Functions		Description
1	E1	a = 0.7 , b = 0.2, c = 0.1	Emphasizing Identity Trust
2	E2	a = 0.3 , b = 0.5, c = 0.2	Emphasizing Recommendation Trust
3	E3	a = 0.3 , b = 0.2, c = 0.5	Emphasizing Risk Trust and Identity Trust
1	Trust Threshold = 0.7		Minimum Required Trust level for interaction

Because of not considering the recommendations, the number of interactions is lowered significantly as shown in figure 12.

Fifth scenario is the general case when Identity trust, reputation and Risk are considered. Entities may perform badly at times, so risk assessment is done. Entities' recommendations are considered as well. This scenario produces the maximum number of interactions after I3.

E. Adaptability of Trust model

The most crucial part of the model is Reputation evaluation. This process consists of evaluating first the set of recommenders and then getting recommendation from them. The evaluated set of recommenders also further evaluates their set of recommenders for getting recommendations. This process continues up to a given level keeping in mind the fact that any recommenders is not asked for recommendation twice. So reputation evaluation module is the most time consuming module.

In some time critical applications like, military applications time is one of the major issues. This model is adaptable to these situations too. This model works even if recommendations are ignored. The graph in figure 13 shows the time which is saved if recommendations are not considered.

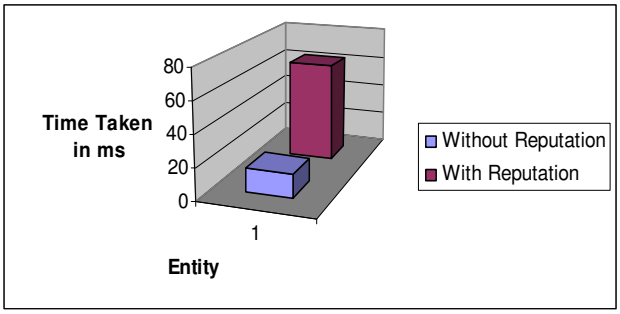

Figure 13 Adaptability of Trust model

F. Trust Decay parameter

The proposed trust model supports trust decay. With time the importance of trust decreases, so the value of trust should decay. As shown in figure 14, the average trust value without using trust decay is 0.7 for IT entity. The average trust value by using trust decay for IT entity is 0.6. With the decay of trust, the number of interactions becomes less as compared to when the trust doesn't decay.

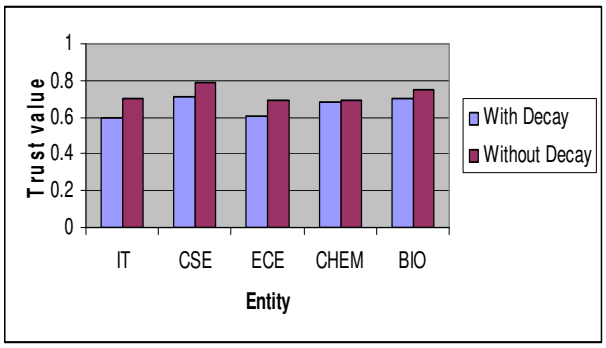

Figure 14 Trust Decay parameter

G. Trust as 3-D quantity

This model sees trust as a three dimensional quantity. The results of this model have also been compared with that of model with trust as one dimensional quantity as shown in figure 15.

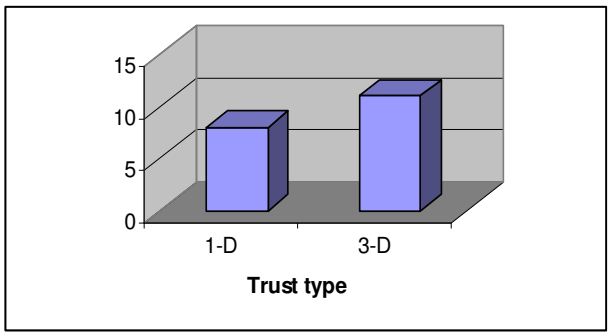

Figure 15 Trust as 3-D quantity

VI. CONCLUSION AND FUTURE SCOPE

The proposed model includes all the parameters identified in Section III and has strong mathematical foundations. Trust is evaluated on the basis of three components, Identity Trust, Reputation and Risk Trust. The weightage of Identity Trust is always more than that of Reputation. The three dimensional view of trust is explored through Dempster Shafer theory as it handles uncertainty component of trust explicitly. The model is realized by a number of equations. Also algorithms for Trust evaluation, Identity Trust Evaluation, Recommendation trust evaluation, Risk trust evaluation and Trust updation are given.

The proposed trust model is adaptive to numerous situations. For example, in some scenarios recommendations can not be trusted. Number of interactions increases as the number of good known entities increases. Number of

interactions decreases as the number of bad known entities increases.

In the current grid scenarios, trust evaluation alone is not sufficient for security. Though trust evaluation is important, but it should be supplemented by other secure methods like cryptographic based security mechanisms that can enhance security. This model can be implemented in a practical application as a separate layer of Grid security architecture. Being implemented as a soft trust, this model is not specific to a particular Grid Environment so it can be easily intergated in different Grid Computing Platforms. This model enhances the domain level security. An extension to this work can be a combination of the hard trust and soft trust.

VII. REFERENCES

[1] C. K. I. Foster and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," in International Journal of Supercomputer Applications, 2001.

[2] Sarbjeet Singh and S. Bawa,"A framework for handling security Issues in Grid Environment using Web Services Security Specifications",Second International Conference on semantics, knowledge & Grid(SKG2006) Nov06,Guilin,China,pp 68-68.

[3] Lars Rasmusson and Sverker Jansson, "Simulated Social Control for Secure Internet Commerce", in Catherine Meadows, editor, Proceedings of the 1996 New Security Paradigms Workshop. ACM, 1996.

[4] F. A. Maheswaran and M., "Evolving and managing trust in grid computing systems," in Proceedings of the 2002 IEEE Canadian Conference on Electrical Computer Engineering, 2002,pp 1424-1429

[5] A. Abdul-Rahman and S.Hailes, "Supporting trust in virtual communities," Hawaii Intl Conferrnce on System Sciences, 2000.

[6] Ching Lin, Vijay Varadharajan, Yan Wang and V. Pruthi,"Enhancing Grid Security with Trust management", Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04),0-7695-2225-4/04 $ 20.00 IEEE.

[7] Farag Azzedin, Muthucumaru Maheswaran, "Trust Modeling for Peer-to-Peer based Computing Systems", In Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03),2003, 0-7695-1926-1/03/$17.00 (C) 2003 IEEE.

[8] S. Lee, R. Sherwood, and B. Bhattacharjee. "Cooperative Peer Groups in NICE", In IEEE Infocom, San Francisco, CA, Apr. 2003, 0-7803-7753-2/03/$17.00 (C) 2003 IEEE.

[9] Zhengqiang Liang and Weisong Shi, PET: "A PErsonalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing", In Proceedings of the 38th Hawaii International Conference on System Sciences – 2005, 0-7695-2268-8/05 $ 20.00(C) 2005 IEEE, pp 1-11.

[10] Dolly Sharma, Sarbjeet Singh, Amandeep Verma, Seema, "A Comprehensive approach to Trust Management in Grid Computing", IEEE International Conference IACC in Thapar University,Patiala. Pp 3105-3110.

[11] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The Ponder policy specification language," Wortshop on Policies for Distributed systems and Networks, 2001.

[12] Jurg kohlas Paul-Andre Monney, "Theory of evidence - a survey of its mathematical foundations, applications and computational anaylsis" (1994) ,Institute of automation & Research, University of Fribourg, Switzerland. Research partly supported by Grants no 21-30186.90 and 21-32660.91 of the Swiss National Foundation of Scientific Research.

[13] Justin R.D. Dyson, Nathan E. Griffiths, Hélène N. Lim Choi Keung, Stephen A. Jarvis, Graham R. Nudd, "Trusting Agents for Grid Computing", 2004 IEEE, pp 3187-3192.

[14] Michal Witold Jarmo lkowicz, "A Grid-aware Intrusion Detection System", Kongens Lyngby 2007,IMM-THESIS-2007-109.

[15] Teresa F.lunt, "Detecting intruders in Computer System", Computer Science laboratory, SRI International. Menlo Park,California, 94025[24] Adams C. and Farrell S., RFC2510 - Internet X.509 Public Key Infrastructure Certificate management Protocols, 1999.

[16] A. Josang, "A logic for uncertain probabilities," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems", vol. 9, no. 3,pp. 279–311, 2001.