Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 179 (2007) 59–73
www.elsevier.com/locate/entcs

A Scalable Evidence Based Self-Managing Framework for Trust Management

Mohammad Waseem Hassan[a,1], Richard McClatchey[a,2], Ian Willers[b,3]

[a] *Centre for Complex Cooperative Systems*
University of the West of England
Bristol, UK

[b] *PH Division, CERN*
CH-1211 Geneva 23, Switzerland

Abstract

Centrally managed, traditional security systems put limits on collaborative activities among huge number of entities in current open networks (such as Grids). This requires new approaches to handling security in large distributed systems and the need for new research especially in areas concerned with the provision of security through collaboration. This paper presents the design of a large-scale, self-managing Trust Management Framework (TMF) that makes efficient use of apparently invisible evidences that are scattered across potentially global networks. The TMF's design dictates a layered architecture for capturing evidence at the data layer of a network, transforming it into formed reputations in the information layer and utilizing these reputations to determine trustworthiness of an entity in the knowledge layer of the network. In essence, the main focus of the proposed work is to automate the acquisition of scattered evidence and the formulation, evolution and dissemination of reputations in a scalable way in order to make improved security decisions.

Keywords: Trust management, Scalability, Grid, Self-management, Reputation evolution

1 Introduction

"Grid" by definition promises to create effective "Virtual Organizations" (VOs) based on efficient, secure and trusted collaborations so as to establish the foundation for new forms of coalitions - for example amongst commercial, academic, government and international research and development organizations [4]. During the formation of a VO a group of participants with similar interests interacts in order to grab any upcoming opportunity for mutual benefit. Out of the dynamically selected

[1] Email: waseem.hassan@cern.ch
[2] Email: richard.mcclatchey@cern.ch
[3] Email: ian.willers@cern.ch

doi:10.1016/j.entcs.2006.08.031

participants there might be only a few who would be known to each other and others might be unknown, but in order to collaborate they should "trust" each other. In this context an important question that arises is: How can one judge the "trustworthiness" of these participants?

From social sciences we know that "trust" is a phenomenon which is build over time by having personal experiences with others. Then these collected experiences are used to judge how others will perform in an altogether new situation. However, when assessing our trust in someone with whom we have no direct personal experience, we often ask others about their experiences with this individual. This collective opinion of others regarding an individual is known as the individual's reputation and it is the reputation of this individual that we use to assess its trustworthiness, if we have no personal experience [9]. We advocate that this principle should be applied in a similar manner to collaboration between computing entities in dynamic VOs.

Trust-related information is important in making dependable online dynamic decisions in VOs. For example during the formation of a VO trust information would be essential. In general, *"Trust management is a mechanism that allows establishing mutual trust among participating entities"* [1]. A practical Trust Management Framework (TMF) should be capable of handling all the measures that are required in the trust establishment process in a scalable way. From [1], it is advocated that *" A powerful trust model is worthless if it cannot be implemented in a scalable way "*. This implies that the problems of "Trust Management" and "scalability" should be studied in tandem.

From the literature it is evident that attempts have been made to model trust and reputation such that each model represents the requirements of the domain to which they apply (see [7] for a general review of such models). In our case, for trust management in dynamic VOs, the requirements are summarized as follows:

(i) A TMF designed for a dynamic VO must be *scalable* in order to manage trust-related information for entities that are present on the Internet and which could become part of a VO readily. In this regard one of the most important requirements is to uniquely identify entities at a global level (see section 3.1 for a solution).

(ii) A TMF system should be *transparent*; by this it is meant that the availability of reputation information should be instantaneous. The users of such system should be free from the burden of searching the recommenders and gathering the evidence from recommenders to find the reputation of target entity. The *timely availability* of the reputation information at the desired location holds the key for this system.

(iii) *Evolution* in the reputation of an entity is a real time phenomenon. In order to automate this process there is a need to have *self-management* mechanisms which cater for the automatic and periodic collection of evidence and transforming them into reputations.

(iv) In the TMF's system the *trusting policies* should be governed at the node level

i.e. each node should decide its own policy and the trust decisions should be made accordingly. This is because of the fact that trusting attitudes vary between individuals so there should be provision in the TMF system for defining policies dynamically.

The details of the TMF design, based on the above identified requirements, are structured in the following sequence. In the next section some of the related works in the immediate area of research will be highlighted. Section 3 illustrates the concept of Trust Management in a scalable environment. Section 4 will focus on the operations of the working TMF and Section 5 provides evaluations and results. The last section presents the conclusion of the overall research work.

2 Related Research

Abdur Rahman et al. proposes a trust model in [10] based on the social aspects of trust and reputation. In our work we build upon some of their basic ideas, however their model has some limitations. For example, they make recommender's chain to get the recommendations due to which this model cannot be scaled in an environment where millions of entities exist. The work of Kwei-Jay Lin et al. [8] is similar to the work presented in this paper. They uses a network of brokers, but their approach could cause congestion in the network and in case if reputation is not found locally then the broker network is searched randomly through broker-broker protocols, which could be a time consuming activity, especially when scalability issues are an important consideration. In the approach presented in this paper the Reputation Servers collaborate in an offline setting to collect evidences from different domains and to update reputation values resulting in the availability of reputation in the proximity of the entity whose reputation is required to be determined. Aberer and Despotovic [1] present a scalable peer-to-peer (P2P) evidence locator based on their P-Grid data structure. In their approach they only consider complaints as behavioural data and assume only one context whereas our approach is more flexible as we consider the positive attributes and any number of contexts. In trust management the role of a "context" is extremely crucial as emphasized by many researchers [10,5]. It should be noted that none of the above approaches focus on the concept of "Global identities" that is very crucial when scalability is the main consideration ."Global identities" are required to recognize the entities uniquely on a global scale such that it should be clear that whose "reputation" is being asked, about whom evidence is being collected or in general whose "trustworthiness" is being seeked. However, David Ingram [6] presents a trust management solution for P2P systems and emphasizes the need for "Global identities". In his approach he proposes a "few in a lifetime identities" from organizations like Identity Providers (IPs). Furthermore, to discourage its frequent use he proposes that taking a new identity should be an expensive activity. But this approach cannot completely stops an agent having multiple unlinked identities. In contrast in our approach an agent can have multiple identities but these identities are linked together through the proposed TMF system.

3 Trust Management in a Scalable Environment

The research presented in this paper proposes a distributed large-scale, self-managing Trust Management Framework (TMF) that addresses the challenges of mining scattered evidences, which is apparently invisible, and combining these evidences to form reputation and most importantly catering for the evolution of the reputation information. According to [11], the concept of reputation evolution is central to trust management system but it is rarely discussed in a practical way. In order to provide scalability and ease of management, the proposed TMF is designed in layers. By this design it is intended to enhance the flexibility and extendibility of the overall system. The first layer is used for capturing the evidence at the data layer. In the second layer this evidence is used to form reputations in the information layer and finally these reputations are utilized to determine the trustworthiness of an entity in the knowledge layer.

Figure 1 provides a detailed view of the proposed Trust Management Framework. It consists of a network of Trust Domain Controllers (TDC), Reputation Servers (RS) and RSLocator (RSL). At the domain level entities are registered in TDCs and each TDC is registered with a RS of its domain and is under the control of a real world organization. A TDC is responsible for: 1) issuing and maintaining identities of the entities, 2) storing their experiences, 3) registering entities and their list of interactors with the Reputation Server (RS) of its locality, and 4) providing evidence once required. Here a TDC normally represents a physical organization.

Each RS is responsible for: 1) collecting evidences for its registered entities, both locally and remotely, 2) calculating reputation from these evidences, 3) providing reputation information once required. Generally RSs are considered trustworthy mainly because of two reasons: Firstly, they do not have any interest in skewing reputation information. Secondly, they run publicly known algorithm and hence are transparent to be judged by anyone on any decision. These RSs are registered with a central Registry service called RSLocator (RSL). It should be noted that the RSL is responsible for: 1) Registering the RS into its database, 2) Providing interface for any modification in the RS information, 3) Return the address of the RS upon request from outside. This registry is queried by RS(s) to locate another RS in order to obtain evidence or reputation information.

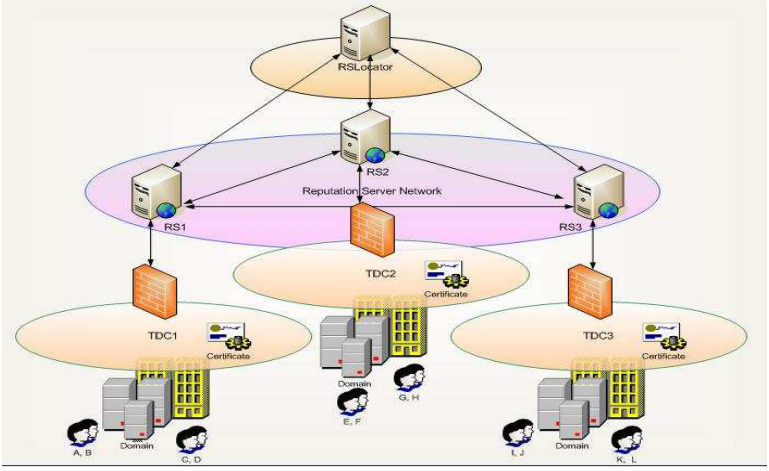
In the presented TMF the trustworthiness of an entity is calculated locally in accordance with the local policies of the domain and inline with the preferences of the trustor. This means that, as in the real life, in the decision making process the trusting attitude of the particular individual will be reflected who is involved in the transaction.

In the following the main issues resolved by the proposed design of the TMF are dispatched to satisfy the requirements set in section 1.

3.1 Global Identities

To solve this challenging problem we have adopted a very simple, yet effective, approach that does not cost extra effort. We propose global identities similar to an

Fig. 1. Trust Management Framework



"email address" to each entity on the inter-network. This means that the identity comprises of a user's id and its location information. For example `Jan@qau.edu.pk` has user id '`Jan`' and domain information i.e. '`qau.edu.pk`'. So these identities are uniquely identifiable in the same manner as an email address is always unique. For example, while sending email to somebody on the internet we are absolutely sure that our email will reach the intended recipient. We do not have to consider that our email address (if correct) will end up at an unknown location. In order to prevent Sybil attack [6], we propose the delegation of certain responsibilities to the physical organizations. In this regard when a new entity is added in an organization, the organization first verifies the entity's identity and other related information thoroughly. After verification process the organization issues a unique identity to the joining entity. Furthermore, an organization is also responsible for updating the records as soon as an entity acquires another identity or as soon as it leaves the organization. The organization manages all these tasks with the help of a Trust Domain Controller (TDC). In this way, once a user is registered with one organization, at least (s)he cannot acquire duplicate identities from his own organization and in case of acquiring another identity from another organization the user is required to inform the TMF system about the same. Consequently multiple identities of a single user are linked through the TMF system.

3.2 Reputation Availability

The availability of the reputation information is very important at a desired location in a timely manner when instantaneous decision making is being done in an online environment. To achieve this we have adopted a strategy in which the reputation of each entity is calculated and stored at the RS, where it is registered, in an offline setting. In this way the reputation information of each entity is present beforehand and is returned immediately upon querying. By following the design strategy employed in the TMF, the placement of a RS is very important, particularly

since this helps in later searches. Earlier it has been stated that a set of TDCs are associated with a RS, so if the namespace allotted to a single RS is such that in each country, for example, for each major DNS domain like 'edu', 'gov', 'org' etc. there is one RS then that RS will be responsible for holding reputation records related to that major DNS domain. This RS could be further made up of a hierarchy of RSs but from the top only the super-RS will be visible. Here for simplicity we only consider one RS per one major domain. Now the RS from each domain forms a network of RSs which is later used to disseminate the 'reputation' information efficiently. One important point that should be noted is that the address of the RS comes from this major DNS domain, e.g. 'reputation.edu.pk'. This strategy makes the searching of the desired RS efficient while to finding the reputation of an entity. For example, a user `Jan@qau.edu.pk` is registered with the reputation server '`reputation.edu.pk`'. When the reputation about Jan is required, then from his own identity his reputation server will be predicted and reached through the RLS and since the RS '`reputation.edu.pk`' is responsible for keeping reputation information of its registered user it can easily returns his reputation value.

3.3 Reputation Evolution

Here we turn our attention towards an important social aspect which relates to the concept of reputation evolution. The proposed TMF system is designed to cater for this mechanism. As reputation evolves with time, a low fixed value is given to a newcomer 'entity'. When the number of interactors, for that entity reaches a threshold value (10 in our case), its actual reputation is calculated and its reputation is accordingly updated. In the TMF the process of reputation evolution is done through self-management and is described as follows. For a given entity, a process of random sampling is carried out periodically upon this entity's list of available interactors to select a set of four interactors. These interactors are then polled to acquire evidence about the desired entity. Then these mined evidences are used to find the reputation of an entity through a publicly known algorithm. The advantage gained by employing this strategy is two-fold. Firstly, through a self-management mechanism the reputation of an entity is evolved. Secondly, the possibility of collusion among opinion providers is put under check. It is because of this reason we have selected the threshold value for the interactor as 10 for the newcomers such that sampling could be done to select final four interactors to poll evidences. Lastly, if an entity commits a destructive action, then this action should be reported to its RS immediately through the TDC.

3.4 Opinion Provider's Reliability

Generally after an interaction an agent evaluates its opinion provider's reliability for future use. This phenomenon is termed as Meta Trust in [6]. In the proposed TMF the RS basically collects opinions but it itself is not involved in any kind of interactions and therefore the evaluation of opinions is not possible at the RS. To reduce the chances of collusions the above-mentioned strategy has been employed

where the RS randomly selects the interactors of an entity from the stored list in order to poll the evidence. With this strategy the chances of a single interactor skewing the reputation of an entity is low since it does not have the control to be selected in the random selection process. In the case that it is selected, it should anyway be in a minority of one.

3.5 Dynamic Policies at the Local Node

This section discusses the functioning of the TDC in the context of the decisions being made. In the TMF system the trusting policies are governed at the node level i.e. each node decides its own policy and the trust decisions are made accordingly. The policies that are used in TMF are as follows, and they have been adapted from the works of Abdur Rehman et al. [10] as mentioned in section 2. *1) Analyze Policy:* This policy determines how trust values should be calculated by inspecting a particular context and phase of the trust relationship. For instance, the trust value can be calculated by considering previous experience (i.e. direct trust), reputation (i.e. indirect trust) and dispositional trust or any combination of these. It should be noted that we have selected discrete trust levels and the trust relationship phases include: trusted, known, unknown and untrusted. *2) Decision Policy:* This policy determines whether a trustee is enough trustworthy to be granted access by inspecting trust and risk threshold values. *3) Context Experience Policy:* In case if dispositional trust is used to calculate the trust value, this policy determines how to aggregate the experience values to make a single trust value in a particular context. *4) Stereotyping Policy:* In case if dispositional trust is used to calculate the trust value, this policy determines: i) which attribute should be used for stereotyping and ii) how to aggregate the experience values to make a single trust value in a particular context. It should be noted that in the decision making process the trusting attitude of the particular individual, who is involved in the transaction, is incorporated.

4 TMF in Operation

This section provides a scenario based on dynamic VOs in which we demonstrate the use of our proposed TMF. This scenario is taken from MammoGrid [2] project's technical document [3]: Research laboratories continuously develop novel medical imaging diagnostic technologies that are well received among medical image analysis peers. Encouraged by these results, researchers naturally aim to bridge the gap to the medical community and seek publication of the results in medical journals to demonstrate the merit of the technology in clinical settings. A major hurdle is encountered when a scientifically acceptable clinical trial that fulfills the criteria of evidence-based medical research needs to be organized. In this regard access to the necessary quantities of medical data that can statistically prove the real applicability of the system is difficult. In order to fulfill this short-term need a dynamic VO could be formed which allows users to perform their desired task in a trusted secure environment. The formation of this VO requires several trust-based

Table 1
Previous experience

Interactor	Context	Phase	Experience Value
Jak@abc.co.uk	Data reliability	Trusted	2
Jones@xyz.co.uk	Data reliability	Known	0
green@pqr.com.pk	Confidentiality	Unknown	Not Available
brown@uvw.com.pk	Confidentiality	Unknown	Not Available

Table 2
Analysing Policy

Context	Phase	Risk	Trust Type
Data reliability	Trusted	Reliability	Direct Trust
Data reliability	Known	Reliability	Direct Trust, Reputation
Confidentiality	Unknown	Confidentiality	Reputation

decisions e.g. the service requestor (or medical researcher) would like to use trusted resources such that his novel algorithms are not compromised whereas the service providers (i.e. the data and computing resource providers) also want to offer their resources to trusted consumers. In the following we only consider the situation where the researcher has to find trustworthy resources with which he would like to form a VO. The case for service provider will be similar; due to space restrictions it will not be covered here.

4.1 Calculating Trust and Establishing Reputation

Suppose the identity of the researcher who wants to test his algorithm is Jan@qau.edu.pk. In order to test his algorithm Jan needs: a) a real time medical imaging data from a reliable source and b) a computing resource on whose confidentiality he can trust. To achieve his goal Jan needs to form a VO with trustworthy service providers to obtain a) and b). Here we make some assumptions: that Jan has past experience with two entities that can provide data and that Jan has no experience in using computing resources. Jan's trust relationship phases and past experiences with these entities are shown in Table 1.

It should be noted that for this simple illustration we have considered just two entities and two contexts for this discussion. It is obvious that Jan would like to choose the most trustworthy entity available in each of the contexts described in Table 1. In the following it is described how Jan can make trust-based decisions to achieve his goal by using the TMF described in this paper.

Before we investigate which of the possible candidates are most trustworthy, it would be worthwhile to know the trusting policies of Jan which are available as shown in Table 2 and Table 3.

According to Jan's analyzing policy, if the trust relationship is in the "Trusted"

Table 3
Trust Decision Policy

Context	Phase	Risk Threshold	Trust Threshold
Data reliability	Trusted	0	0
Data reliability	Known	0.5	1
Confidentiality	Unknown	0.9	1

phase and the context is "Data reliability" then the Trust Type is "Direct Trust". (By "Trust Type" we mean how the trustworthiness of the trustee should be calculated).

In our example it means that only Direct Trust (i.e. previous experience) will be considered to determine the trust value and from Table 1 this value comes out to be 2. Furthermore, it would also be considered how much risk will be involved in this transaction. For example, Table 3 indicates that in the context of "Data reliability" and a "Trusted" phase relationship the Risk Threshold value has its minimum value and the trust value is more than the Trust Threshold value. The final trust decision will be made depending upon the values for trust and risk and based upon Jan's Trust Decision Policy. From Table 3 it is clear that in accordance with Jan's Trust Decision Policy when the context is Data reliability and the trust relationship is "Trusted" then Risk Threshold has its minimum value and the trust value is more than the Trust Threshold value and hence the resource is declared are "trustworthy".

In this example, in the case of computing resources Jan discovers two entities `green@pqr.com.pk` and `brown@uvw.com.pk` with "Unknown" trust relationships. According to his analyzing policy (see Table 2), if the trust relationship is in the "Unknown" phase and the context is "Confidentiality" then the Trust Type is "Reputation". This means that only the reputation value will be considered to determine the trust value. In order to find the reputation of the two discovered entities, the Reputation Network will be searched. Thus a request is made to Jan's RS, where he is registered, and since the domain of both Green and Brown is different, the cache in the Jan's RS will first be searched. If no reputation value is available then the RSLocator will be contacted to get the end point address of the RSs of Green and Brown. The addresses of the RSs of Green and Brown will be predicted from their identities. For example, suppose the reputation value for Green is found to be 2 and that of Brown 0 then these values are first sent to the TDC where Jan is registered. After this the trustworthiness of Green and Brown will be calculated, depending upon Jan's Trust Decision Policy, In this example scenario the calculation of trust and the finding of the reputation values have been described.

In the next section another important scenario in the TMF system will be presented which relates to offline opinion acquisition and hence the formation and evolution of reputation information at a given RS.

Table 4
Remote Opinion

User	Interactor	Context	Opinion	Timestamp
Jim@abc.co.uk	Jan@qau.edu.pk	Data reliability	2	25/02/2006

4.2 Opinion Acquisition

Once the new entity's registration process has been completed at the TDC the next steps involve the registration of the entity in its respective RS. It is the responsibility of the RS administrator to validate the request from the TDC. It should be noted that, along with the entity, the list of interactors is also registered in the RS. If the number of interactors is less than the required threshold (i.e. 10 in our TMF), the entity is given a fixed low reputation value. Once the number of interactors reaches the threshold, its reputation is calculated by getting opinions from his interactors. To explain this process consider an example. Jan@qau.edu.pk is registered in a RS of domain "edu.pk" and his list of interactors from three different domains are as follows: {Jim,Roy,Don}@abc.co.uk, {Jak,Hic,Seb}@xyz.co.uk , {Gur,Raj,mik,bob}@pqr.com.pk. We assume that the context is "Data reliability". Next we calculate the reputation of Jan. First of all 4 interactors are randomly selected through the process of sampling, this being carried out in the RS. These randomly selected interactors are as follows: Jim@abc.co.uk, Jak@xyz.co.uk, Gur@prq.com.pk, bob@prq.com.pk.

Now let us see how Jim@abc.co.uk is contacted by Jan's RS to fetch the evidence. The RS parses Jim's identity (Jim@abc.co.uk) to find his domain (co.uk). Then the first RS looks for an entry of the remote RS from "co.uk" domain in its cache. If it does not find it in the cache then it queries the RSLocator to get the address of the same. Once the required address has been found, Jan's RS contacts Jim's RS to get the desired evidence. On the other end Jim's RS forwards this request to the TDC in which Jim is registered. This TDC in turn queries its database to find Jim's opinion about Jan in the requested context. The result of the query is shown in Table 4.

Here it is worth noting that one entity could have multiple interactions with another entity. However this entity has only one opinion about the other entity. Furthermore, this opinion gets updated after every new interaction with that entity. To incorporate this fact from real life, the opinion and trust relationship phase are updated in the TDC after every interaction and the timestamp for the said opinions is also recorded. Returning to our example, the TDC will return the required opinion to Jan's RS. Table 5 shows the evidence values returned from the RS of each of the interactors.

Once all the evidence is collected then the next job is to evaluate this evidence in the light of how old each of the evidences values is. In a manner similar to [10], we select a weighting factor and call this the "time weight" t_w. This time weight is used to give preference to the latest evidence. The summation of evidence after

Table 5
Evidence Values

Interactor	Evidence Value	TimeStamp
Jim@abc.co.uk	2	25/02/2006
Jak@xyz.co.uk	1	15/12/2005
Gur@pqr.com.pk	1	01/04/2004
bob@pqr.com.pk	2	20/09/2005

Table 6
Evidence weights

Evidence timestamp	Current Year(CY)	CY-1	CY-2	CY-3
Weight	4	3	2	1

incorporating time weight value can be described as follows:

$$Sum_e = \sum_{i=1}^{n} t_{wi} \tag{1}$$

Table 6 shows the Evidence weights for our example. From Table 5, it is clear that there are two different evidence values. By using formula (1) the total time weight for value 2 is 7 (4 + 3) and for value 1 is 5 (3 + 2). This means that the final reputation value for Jan in the context of Data reliability is 2. As a final step this reputation is stored in the RS and evolved with time on a periodic basis.

Here it is worth mentioning that the evidences gathered in order to form reputation are deleted from the system so as to avoid any privacy concerns. Hence the only information available in the RS is reputations which anyway is a public information.

5 Evaluations & Discussion

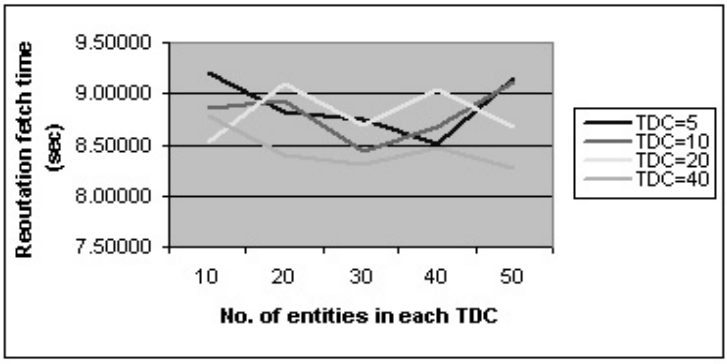
In this section the TMF system is evaluated in terms of its performance and an investigation is carried out to determine how the presented design impacts the scalability of the adopted approach. In this regard we have carried out a set of tests assisted by some simulations. The tests are conducted in multiple batches at different times to determine average responses. The purpose of the simulations is to evaluate, in realistically larger settings, the scalability of our approach. In this regard it should be noted that for simulations a set of pre-conceived situations are considered to record data from different 'simulation runs'. In general, the results along with associated discussion provide a view of the expected outcome of the TMF, as presented earlier.

In section 3 we identified the need for "Global Identities" in order to achieve a global and scalable trust management solution. In our solution we emphasized the use of "email address"-like identities. The advantage of this approach is twofold: firstly it is simple to apply and secondly it is an agreed standard among the overall

population of information technology users. However, the use of email addresses alone will not suffice but an infrastructure like TMF, involving both physical and virtual organizations is necessary for the success of the overall concept. In our approach TDCs are only responsible for introducing the entities in the RS so as to reduce the chances of entry of malicious agents in the system.

In the context of scalability an important test has been conducted where the reputation fetching time has been recorded by increasing the overall population of the TMF system. Figure 2 shows how the system responds as the number of entities increases in the overall system. The variations in the TMF system settings are described in the following: the total number of RS = 10, TDCs in a single RS vary as 5, 10, 20, 40, Entities in each TDC vary as 10, 20, 30, 40, 50 and the number of contexts varies from 1 to 3.

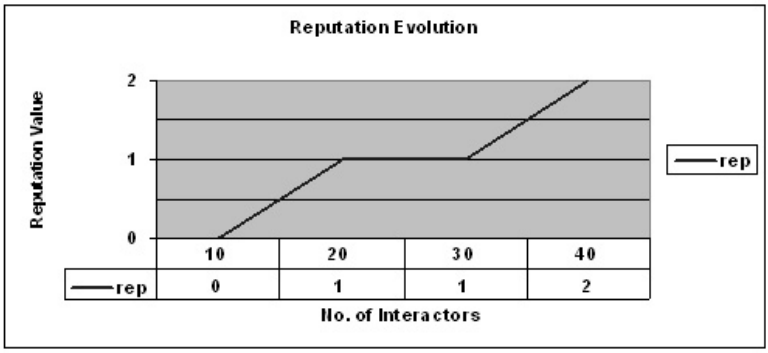
Fig. 2. Reputation time vs. No. of available entities



The maximum number of entities in the TMF system for these tests is 10*40*50 = 20,000. With the increase in the contexts the number of reputation record varies as: Context * No. of Entities e.g. for 3 contexts and 20,000 entities the total records in one RS is 3 * 20,000 = 60,000. Figure 2 shows the trend in the graph and it is evident that the variation in all of these cases is symmetrical i.e. within the same range. Similarly, if the number of RS is increased to "any" number then the response time of the TMF system will remain same as for each "RS search" same number of network call will be required. This is an important factor and makes our system *scalable*.

Another important aspect of any TMF system is *Transparency* i.e. instantaneous availability of the reputation information. No matter how sound is the reputation calculation algorithm in a system if this information is not available at the correct time then it will be of no use. In our approach, we are forming, evolving and managing reputations in the proximity of the user in an offline setting. By employing this strategy a significant number of online network calls are avoided which makes our approach scalable. Secondly, our strategy is also very effective and efficient in terms of finding the right RS in a minimum time. The caching mechanism at each RS helps in avoiding network congestion. Each RS keeps a record of the most frequently accessed remote RSs and therefore in most of the cases the RS search

Fig. 3. Reputation vs. No. of Interactors



finishes at the local RS.

Furthermore, we have estimated that roughly 10 RSs per country will be sufficient to represent major domains, so if we consider 100 countries then the total number of entries in the RS Locator will be about 1000, which is not very large to manage in a registry. In terms of reputation availability we have carried out simulations. In each simulated run we generated a set of queries to find reputation of different entities. The result shows that during all the simulated runs the queries hit the right RS to get the reputation information instantaneously. Figure 2, as discussed previously, also demonstrates the instantaneous availability of the Reputation information. In essence, in our proposed TMF one of the main idea is to make sure that Reputation value should be instantaneously available through the network of RSs.

Reputation Evolution is another important aspect of the TMF system. Reputation does not instantly evolve immediately after one interactor has reported its experience, with the only exception that if for example, simultaneously 5 independent interactors report the same problem then the system explicitly checks for the reputation information of this particular entity out of its routine activities. Otherwise, in general, the TMF system caters for reputation evolution periodically. The simulations regarding Reputation Evolution are depicted in Figure 3, which shows the variation in the Reputation values as the number of interactor increases. The entity, as a new comer, has a minimum reputation value as expected but as its interactors increases its reputation changes accordingly. In general, reputation evolution has a direct impact on the overall trust based decisions in the system.

Lastly, our TMF is flexible enough to incorporate different trusting attitude of each entity by customizing *dynamic policies* at each node. In this manner the decision making process at a particular node reflects the trusting attitude of the particular individual who is involved in the transaction.

Having presented our results, allows us to provide a more detailed comparison with the related approaches. The work presented in [10] is of theoretical nature and same is the case in [6]. However, the works described by Kwei-Jay Lin et al. [8] suggests an approach where reputation information is searched online whereas

in our case this is an off-line process. Therefore in terms of the availability of the reputation information our results are more efficient than theirs. Apart from this they make a big assumption that agents will not cheat in providing information and they also do not consider the aspect of trusting dispositions. Moveover, in their case reputation evolves after every interaction, no matter it is between the same pair of entities. This gives rise to the chances of collusion as same pair of entities could potentially involve in fake interactions with the aim to raise their reputations. Note that our main emphasis is not on the number of interactions but the number of interactors. The work presented in [1] is comparable to ours but it is mainly meant for P2P environments. Furthermore, they do not consider the important aspect of 'context' which has a direct impact on the scalability of the approach because as discussed in this section the total number of records in the system is proportional to the number of contexts.

6 Conclusions

Trust Management plays a crucial role in a society which is becoming increasingly dependent on networked information systems. In this paper the design of a large-scale, self-managing Trust Management Framework (TMF) has been presented. The salient features of this design include: 1) the provision of an efficient and reliable mechanism for collecting evidence from the global inter-network, 2) the capability for disseminating reputation information very effectively in the distributed systems scattered over the globe and 3) the ability to cater for the social phenomenon of reputation evolution in computation. The TMF provides a platform where the vision of dynamic virtual organizations involving any number of either known or unknown global partners can be realized. Our results determine that the adopted approach is scalable and very efficient. In essence, the focus of our work revolves around the practical aspects of building a scalable Trust Management System. The results from our research would be very beneficial for the ongoing research in the field of Trust Management in many domains e.g. ecommerce, mobile agents platforms and P2P computing.

References

[1] Aberer, K., et al, *Managing Trust in a Peer-2-Peer Information System*, in: *Proceedings of the 10th Intl. Conference on Information and Knowledge Management*, 2001.

[2] Amendolia, S. R., et al, *MammoGrid: A Service Oriented Architecture based Medical Grid Application*, Lecture Notes in Computer Science **3251** (2004), pp. 939-942, ISBN 3-540-23564-7 Springer-Verlag. (Proceedings of the 3rd International Conference on Grid and Cooperative Computing (GCC 2004). Wuhan, China. October 2004).

[3] Annex 1 - "Description of Work" of Mammogrid Project. Proposal number: IST-2001-37614.

[4] Foster, I., et al, *The Anatomy of the Grid - Enabling Scalable Virtual Organizations*, International Journal of Supercomputer Applications (2001).

[5] Grandison, T., et al, *Trust Management Tool for Internet Applications*, in: *Proc. 1st Int'l Conf. of Trust Management*, LNCS 2692, Spinger-Verlag, 2003, pp. 91-107

[6] Ingram, D., *An Evidence Based Architecture for Efficient, Attack-Resistant Computational Trust Dissemination in Peer-to-Peer Networks*, in: *Proceedings of the 3rd Int. Conf. iTrust 2005*, Paris, France, May 2005.

[7] Josang, A., et al, A Survey of Trust and Reputation Systems for Online Service Provision, Conference on Decision Support Systems, 2005.

[8] Lin, K. J., et al, *A Reputation and Trust Management Broker Framework for Web Applications*, in: *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, pp. 262-269, March 2005

[9] Patel, J., et al ,*A Probabilistic Trust Model for Handling Inaccurate Reputation Sources*, in: *Proceedings of the 3rd Int. Conf. iTrust 2005*, Paris, France, May 2005.

[10] Rahman, A., et al, *Supporting trust in virtual communities*, in: *Proceedings of the 33rd Hawaii Intl. Conference on System Sciences*, Maui Hawaii, 2000.

[11] Ruohomaa, S., et al, *Trust Management Survey*, in: *Proceedings of the 3rd Int. Conf. iTrust 2005*, Paris, France, May 2005.