УДК 52(15).003

KUSSUL O., NOVIKOV O.

UTILITY-BASED REPUTATION MODEL FOR VO IN GRIDS

В данной статье предложена модификация существующей модели репутаций для виртуальных организаций в Grid-системах, которая основана на оценке функции полезности. Модификация модели состоит в добавлении статистической модели поведения пользователя, которая ранее была разработана для компьютерных сетей и распределенных систем, а также компонентов, которые позволяют противостоять угрозам в области управления доверием и репутацией. К числу этих компонентов относятся: механизм присвоения начальной репутации для новых субъектов виртуальной организации; учет взаимосвязей между пользователями и ресурсами; функция учета времени; а также классификация предоставляемых сервисов в Grid-системе.

In this paper we extend the existing utility-based reputation model for VOs in Grids by incorporating a statistical model of user behaviour (SMUB) that was previously developed for computer networks and distributed systems, and different functions to address threats scenarios in the area of trust and reputation management. These modifications include: assigning initial reputation to a new entity in VO, capturing alliance between consumer and resource, time decay function, and score function.

Introduction

The key idea of Grid-systems is a coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations [1]. Therefore; mutual trust between users and resource providers is of high importance.

Paper [2] demonstrates that trust is enabling technology and its implementation can provide the possibility to secure electronic transactions. Meanwhile trust is described as an important and sophisticated object dealing with honesty, truthfulness and reliability of trusted person or service. Nevertheless there still there is no common definition of trust [3]. Two main definitions can be given:

— ”When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him. Correspondingly, when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain from doing so” [4]

— ”the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of a relative security, even though negative consequences are possible” [5]

A set of individuals and/or institutions defined by coordinated resource sharing rules form what we call a virtual organization (VO).

Virtual organization is a temporary or permanent union of geographically spread individuals and/or institutions defined by coordinated resource sharing rules, possibilities and information for reaching common goals [1].

VO’s are formed dynamically, exist for some time and then resolve. That is why the efficiency of their work depends on trust.

For monitoring and creation of such VO’s the reputation based trust management technologies are needed.

Currently, such trust models are built based on entity reputation assessment and mostly where created for other areas of distributed systems.

In common case under the reputation we understand the measure of reliability. With the help of reliability we can build a trust of one entity to another. According to [6], reputation is an assumption about the agent’s behavior based on existing information or observations about his behavior before.

In this paper we extended the existing utility-based reputation model [8] by incorporating a statistical model of user behaviour (SMUB) that was previously developed for computer networks and distributed systems [10, 11, 12, 13]. This model was originally used in intrusion detection systems to detect anomalous patterns of user actions. Other modifications include:

- *assigning initial reputation to a new entity in VO*: when organization provides a new resource to be integrated in VO there are no records from monitoring system to infer reputation value for this specific resource. One possible way of assigning initial reputation to a new

resource is to use the methodology of active experiment. There can be several benchmark tasks in the system to estimate the utility function and to provide initial reputation of the resource.

- *alliance between consumer and resource*: since reputation of resource is based on measure of satisfaction of a consumer in relation to this resource we should avoid cheating via collusions among a group of entities [14]. For this purpose, it is advisable to include into the model a factor that will reflect an alliance between the consumer and the resource.

- *time decay function*: reputation of resource is based on measuring average value of utility function over certain period of time [14]. But if VO exists for a considerable period of time (e.g. for years) reputation of resource may vary considerably. That is, it is unlikely to use, for example, two years data to estimate current resource reputation if more recent records are available. So, we propose to incorporate a time lag function into the model that will provide weights depending on the time of the transaction record between consumer and resource.

- *score function*: for different types of services offered by resource providers different reputation values will be used [15]. Namely, we will categorize services into categories, and a resource provider will get reputation value according to such a category. In Grid systems, tasks can be categorised by computational complexity. Successful execution of tasks with complex workflow and parallel programs (for example, environmental models like numerical weather prediction [16]) will provide to a resource provider higher reputation value.

We will also study the described reputation model against the existing attacks and how the model overcomes them. In particular, we will follow security threat scenarios presented in [15]. These scenarios include: individual malicious peers, malicious collectives, malicious collectives with camouflage, malicious spies, sybil attack, man in the middle attack, driving down the reputation of a reliable peer, partially malicious collectives, malicious pre-trusted peers.

Related Works

Up to date, there have been proposed several approaches of reputation models in grid systems.

PathTrust [14] is a reputation system suggested for choosing members of the VO whiles its formation. The organization has to register with the enterprise network by providing some certificates to enter the VO. Besides of user management enterprise network provides centralized reputation service. When the VO is resolved each member gives feedback values for reputation server and other members he had interacted with. The proposed model feels the lack of dynamics, because the feedback value is collected only when the VO is resolving.

Paper [8] proposes utility-based reputation model that can be used for users, resources and resource providers' assessment. This model is based on the calculating of the utility function that expresses the satisfaction of the entity with its interaction with other entities with respect to the key features specific for the assessable entity. For resource reputation assessment this trust model demands the existence of monitoring system for the quality of service data collection. But this paper does not highlight the question of assigning initial reputation values for new users or resources.

Paper [9] is solving the task of assessment of trust level between the agents of multi-agent system using the direct observation data and reputation information. Unlike other existing approaches that are based on the usage of heuristics the proposed approach HABIT (Hierarchical And Bayesian Inferred Trust Model) is using statistical Bayesian methods. The advantage of this model is the possibility of assessing trust values for new agents by searching for correlation with known groups of agents. It is extremely important in cases when the truster has no previous experience or reputation data of trustee. In the model truster-trustee trust is sighted from the providing desirable quality of service point of view. It is obvious that the rational agent is functioning in purpose to gain the maximum of expected utility.

Utility-based Reputation Model for VOs in Grids

The reputation model described in this paper is based on the model proposed in [8]. The main additions are associated with reputation model for VO users, and extensions of three functions to resource provider model. In our model we will also use slightly modified notations as comparing to [8]. All proposed modifications are described in details in the following subsections.

Basic notations used in our paper

The central notion in the reputation model is the organisation [8]. The organisation provides resources; and there exist users associated with this organisation (Fig. 1).

Definition 1. Organization is a set:

$$Org = \left\{ o\_id, \bigcup_i r_i, \bigcup_j u_j \right\}, \quad (1)$$

where *o_id* is an organisation's identifier, $\bigcup_i r_i$ and $\bigcup_j u_j$ are resources and users associated with this organisation, respectively.

We will denote all existing organisations with $\bigcup_n Org_n$.

Virtual organization (VO) can be modelled as a set of organisations. Organisations integrate their resources on a temporary or permanent basis to achieve common goals [1]. It is to be noted that in general case organisation may provide to a VO only a subset of its resources, and the same resource can be used in different VOs. The same stands for users of the organisation.

Definition 2. VO is a set (Fig. 1):

$$VO = \left\{ vo\_id, \bigcup_k r_k, \bigcup_l u_l, f_{vo\_id}(), g_{vo\_id}() \right\}, \quad (2)$$

where *vo_id* is a VO's identifier, $\bigcup_k r_k$ and $\bigcup_l u_l$ are resources and users from multiple organisations that participate in *VO*, respectively, $f_{vo\_id}()$ and $g_{vo\_id}()$ are membership functions defined in the following way

$$f_{vo\_id} : \bigcup_k r_k \rightarrow \bigcup_n Org_n, \text{ i.e. } f_{vo\_id}(r_k) = o\_id \quad (3)$$

$$g_{vo\_id} : \bigcup_l u_l \rightarrow \bigcup_n Org_n, \text{ i.e. } g_{vo\_id}(u_l) = o\_id \quad (4)$$

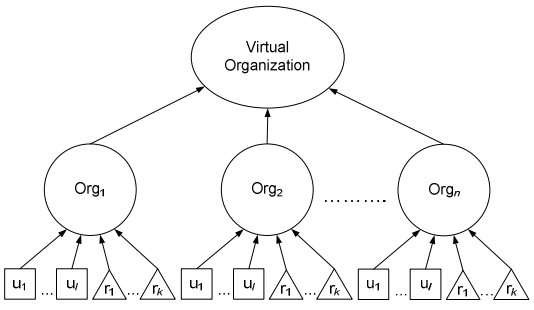

Fig. 1. Correspondence between virtual organization, organizations, resources and users

In general case, these functions can be maintained by the Virtual Organisation Membership Service (VOMS). Using these functions we can retrieve any required information on membership of organisations, resources and users in VOs. According to (3) a set of resources provided by the organisation with identifier *o_id* in a specific VO with identifier *vo_id* is given by

$$\{ r \in \bigcup_k r_k : f_{vo\_id}(r) = o\_id \} \quad (5)$$

We will denote this set as $f^{-1}{}_{vo\_id}(o\_id)$.

In the same, way according to (4), we can list all users from organisation with identifier *o_id* participating in VO with identifier *vo_id*:

$$\{ u \in \bigcup_l u_l : g_{vo\_id}(u) = o\_id \} \equiv g^{-1}{}_{vo\_id}(o\_id) \quad (6)$$

Suppose we want to list all organisations from $\bigcup_n Org_n$ that provide resources within specific VO with identifier *vo_id*, or whose users participate in this VO. According to (3), (4) such sets are given respectively by

$$\{ Org \in \bigcup_n Org_n : \text{if } \exists\ r \in \bigcup_k r_k \text{ that } f_{vo\_id}(r) = o\_id \} \quad (7)$$

$$\{ Org \in \bigcup_n Org_n : \text{if } \exists\ u \in \bigcup_l u_l \text{ that } g_{vo\_id}(u) = o\_id \}. \quad (8)$$

Let us denote with $\bigcup_m VO_m$ all existing VOs. Suppose we want to retrieve all VOs where a resource *r* from a specific organisation *Org* is used, or where user *u* from specific *Org* with identifier *o_id* participates in. According to (2), (3), (4) these sets are given respectively by

$$\{ VO \in \bigcup_m VO_m : f_{vo\_id}(r) = o\_id \} \equiv VO|_r, \quad (9)$$

$$\{ VO \in \bigcup_m VO_m : g_{vo\_id}(u) = o\_id \} \equiv VO|_u. \quad (10)$$

These basic notions are used in the following subsections to describe reputation models for resource providers and users.

Reputation model for resource providers

The reputation model is based on the utility function that measures the level of satisfaction of a user in relation to resource provider. In order to define utility function an auxiliary function that indicates the service level agreement (SLA) accorded between a VO user and a resource provider for a particular resource within a VO is implemented [8]

$$SLA : \bigcup_l u_l \times \bigcup_k r_k \times \bigcup_m VO_m \rightarrow \mathrm{R} \quad (11)$$

where R denotes the set of real numbers.

SLA value represents quality of resource provider as expected by user [8]. Quality of Service (QoS) metrics that can be used to measure the level of satisfaction is described in [17, 18]. Some papers also propose mechanisms for QoS negotiation and handling [19].

Definition 3. Event is a set:

$$Event = T \times \bigcup_l u_l \times \bigcup_k r_k \times \bigcup_m VO_m \times \{\text{QoS name}\} \times \mathrm{R} \quad (12)$$

where T is a time domain.

Therefore, the event is characterised by the following attributes

$\{t, u, r, vo\_id, QoS, v\}$

where t indicates time, QoS is a name indicating QoS of interest, and v is a real QoS value measured by Grid monitoring system after user-resource interaction.

Definition 4. Trace is the sequence of events (12):

$$Trace = \bigcup_p Event_p = \bigcup_p \{t, u, r, vo\_id, QoS, v\}_p . \quad (13)$$

Before defining utility function and reputation we will introduce three functions: the first one will characterise possible alliance between consumer and resource in order to avoid cheating, the second one will account for a time when utility was estimated, and the third one will provide different scores depending on the type of the provided service. These functions provide extensions to utility function and reputation originally proposed in [8].

Function $h(u, r)$ will take value between 0 and 1 and will show the level of alliance between user u and resource r. If there is no such alliance between targets, $h(u, r)$ will have a higher value. For example, one possible way of defining $h(u, r)$ is as follows

$$h(u,r) = \begin{cases} 1, \text{ if } f_{vo\_id}(r) \neq g_{vo\_id}(u) \\ \theta, \text{if } f_{vo\_id}(r) = g_{vo\_id}(u) \end{cases} \quad (14)$$

where θ is a parameter.

Function $z(t, t_c)$ will show what past records on user-resources interactions should be taken into consideration to estimate reputation of specific resource. Here t is the time, and t_c is a parameter. In a simplest form $z(t, t_c)$ could be a stepwise function

$$z(t,t_c) = \begin{cases} 1, t \geq t_c \\ 0, t < t_c \end{cases} \quad (15)$$

Function $s(type(r))$ will provide different values for different types of services provided by the resource r (function $type(r)$ maps into category of service).

Now, we can define a utility function according to (11), (12), (14):

$$utility : Event \rightarrow \mathrm{R},$$

$$utility(\{t, u, r, vo\_id, QoS, v\}) = \begin{cases} h(u,r)s(r), & \text{if } v \geq SLA(u,r,vo\_id) \\ \dfrac{v}{SLA(u,r,vo\_id)} h(u,r)s(r), & \text{if } v < SLA(u,r,vo\_id) \end{cases} \quad (16)$$

.

Let us denote a set of traces that are used to estimate the reputation of resource r in VO with identifier vo_id up to the current time t with

$$Trace|_{(vo\_id,\ r,\ t)} = \left\{ \begin{array}{l} \{t', u', r', vo\_id', QoS', v'\} \in Trace : \\ r = r', vo\_id = vo\_id', t' \leq t \end{array} \right\}. \quad (17)$$

Let us denote a set of *utility()* function values derived from traces $Trace|_{(vo\_id,\ r,\ t)}$ (15), (17) with

$$O_{(vo\_id,\ r,\ t)} = \{\ z(t, t_c) \cdot utility(\{t, u, r, vo\_id, QoS, v\}) \mid \{t, u, r, vo\_id, QoS, v\} \in Trace|_{(vo\_id,\ r,\ t)}\} \quad (18)$$

Definition 5. Reputation is expectation of *utility*() function:

$$rep(vo\_id, r, t) = \mathrm{E}[\ utility(O_{(vo\_id,\ r,\ t)})\ ] = \int utility(O_{(vo\_id,r,t)})\ p_{utility}(O_{(vo\_id,r,t)})\ dO_{(vo\_id,r,t)} \quad (19)$$

.

If we do not want to discriminate values from *utility*() function by time then we might use $z(t, t_c) = 1$.

In order to approximate expectation we can use a sample mean (18)

$$rep(vo\_id, r, t) = \frac{1}{\left| O_{(vo\_id,r,t)} \right|} \sum_{x \in O_{(vo\_id,r,t)}} x \quad (20)$$

where $|\cdot|$ denotes the cardinality of a set.

The reputation of an organisation in VO is the aggregation of the reputation of all resources it provides to VO according to (5), (15), (19):

$$rep(vo\_id, t) = \frac{1}{\left| f^{-1}_{vo\_id}(o\_id) \right|} \sum_{r \in f^{-1}_{vo\_id}(o\_id)} rep(vo\_id, r, t) . \quad (21)$$

The reputation of a resource in all VOs according to (19) can be estimated as follows

$$rep(r, t) = \frac{1}{\left|\mathrm{VO}\right|_r} \sum_{vo\_id \in \mathrm{VO}|_r} rep(vo\_id, r, t). \quad (22)$$

Reputation model for users

In the reputation model proposed in [8] the corresponding model for user is built using a penalty function. If a user performs an action that does not conform to a VO or resource policy, the user is charged with a penalty. This penalty is used to estimate utility function for the user, and subsequently a reputation of the user. But audit of user actions and checking these actions against the VO or resource policy is a difficult task, especially from implementation perspective. There should be clearly defined criteria to assess user actions and corresponding program components to do that.

We propose to include into the reputation model a statistical model of user behaviour (SMUB) that was developed for computer networks and further modified for distributed systems, in particular Grid systems [10, 11, 12, 13]. This model is based on the analysis of statistical data that is gathered after a user executes actions in a Grid system. The model was trained and verified on real data that was collected from GILDA infrastructure (https://gilda.ct.infn.it) of the EGEE project. The model was able to discriminate behaviour of different types of users and to detect synthetically generated intrusions with a rate of more than 90%.

In particular, the model accounts for different statistical parameters given by the following attributes

{*S*, *ET*, *CPU*, *WT*, *CW*, *ES*, *CT*, *STD*, *RAM*, *VM*, *VO*, *RB*},

where *S* is a site where a user task was executed, *ET* is a execution target, *CPU* is a task CPU time, *WT* is a task wall time, *CW* is defined as CPUWall = CPU/W, ES is a task exit status (success or with errors), *CT* is a task creation time in Grid system, *STD* is a start time difference, i.e. difference between time when a task started to execute on computational resource and time when task was sent to Grid system by the user, *RAM* is a volume of RAM used by user task, *VM* is a volume of virtual memory used by user task, *VO* is a virtual organization a user belongs to, *RB* is a resource broker hostname that was used to schedule the user task.

This set of parameters is used to detect anomalous patterns in user actions to raising alarms in the system. Such patterns could include, for example, situations when a task is running for a significant amount of time, or when the CPU utilization is up to 100% [20]. In order to detect such patterns from data that was logged during user activities we use intelligent techniques, namely neural networks [21]—a multilayer perceptron (MLP) trained by means of extended-delta-bar-delta (EDBD) algorithm [22] which represents a fast modification of standard error backpropagation algorithm [23]. For each user a neural network is trained to form an opinion to discriminate between the normal and abnormal user behaviour. When neural network is trained a target output is set to 1 for input data that corresponds to the normal user behaviour, and to 0 for data that corresponds to the abnormal user behaviour. In order to represent both situations (normal and abnormal) in training data sets we use records from Grid monitoring system: records about past user actions represent the normal behaviour while records from other users and synthetically generated data represent the abnormal user behaviour. The synthetic data can be generated using generative models [24] and incorporated into the training data sets in order to represent abnormal patterns that were not present in data sets from monitoring system. Therefore, neural network acts as a classifier. If we put an independent sample (not present in the training data set) to neural network input the output value will be between 1 and 0. This value can be treated as a posterior probability of normal/abnormal user behaviour: higher values correspond to normal actions while lower values correspond to potential anomalous patterns.

In such a way, we propose to use the output of the SMUB model in order to estimate the reputation of user in VO. Such a user model will be specific to a VO depending on its goals and types of tasks being executed. For example, VO can be oriented on applications that require execution of large number of tasks with relatively small amount of data to be processed by a single task. In such a VO, tasks that consume almost full amount of RAM and virtual memory of resources would be considered as an anomalous pattern. In turn, other VOs can be oriented on applications where a single task consists of a number of elementary tasks each processing large amount of data (an example includes Earth science domain and satellite data processing [25, 26]).

Let us provide a formal description of the reputation model for user based on the SMUB model as it was done for the reputation model for resource provider in previous subsection.

Definition 6. *Event* for the user is a set:

$$Event = \{t, u, r, vo\_id, \mathbf{x}\}. \quad (23)$$

where $\mathbf{x} = (S, ET, CPU, WT, CW, ES, CT, STD, RAM, VM, VO, RB)$.

Definition 7. Trace corresponds to the sequence of events:

$$Trace = \bigcup_p Event_p = \bigcup_p \{t, u, r, vo\_id, \mathbf{x}\}_p . \quad (24)$$

An analogue of *utility*() function (as it was defined for resource providers) is defined in the following way

$$utility : Event \rightarrow \mathrm{R},$$
$$utility(\{t, u, r, vo\_id, \mathbf{x}\}) = SMUB_{(u,\ vo\_id)}(\mathbf{x}), \quad (25)$$

where $SMUB_{(u,\ vo\_id)}(\mathbf{x})$ is an output of the SMUB model.

It should be noted that, in general case, under *utility*() function for users we can use other user behaviour models (e.g. [27, 28]) or a combination of different models to capture different aspects and patterns of user behaviour.

In our case, according to (24), the SMUB transformation is performed by a neural network, and specific to user and VO. Let us denote a set of traces that are used to estimate the reputation of user u in VO with identifier vo_id up to the current time t with

$$Trace|_{(vo\_id,\ u,\ t)} = \left\{ \{t', u', r', vo\_id', \mathbf{x}'\} \in Trace : \begin{array}{l} u = u', vo\_id = vo\_id', \\ t' \le t \end{array} \right\}. \quad (26)$$

Let us denote a set of *utility()* function values derived from traces $Trace|_{(vo\_id,\ u,\ t)}$, according to (15), (25), (26) with

$$O_{(vo\_id,\ u,\ t)} = \{\ z(t, t_c) \cdot utility(\{t, u, r, vo\_id, \mathbf{x}\}) \mid \{t, u, r, vo\_id, \mathbf{x}\} \in Trace|_{(vo\_id,\ u,\ t)}\} \quad (27)$$

Definition 8. Reputation is the expectation of *utility*() function (25),(27)

$$rep(vo\_id, u, t) = \mathrm{E}[\ utility(O_{(vo\_id,\ u,\ t)})\ ] = \int utility(O_{(vo\_id,u,t)})\ p_{utility}(O_{(vo\_id,u,t)})\ dO_{(vo\_id,u,t)} \quad (28)$$

.

In order to approximate expectation we can use a sample mean

$$rep(vo\_id, u, t) = \frac{1}{\left|O_{(vo\_id,u,t)}\right|} \sum_{x \in O_{(vo\_id,u,t)}} x . \quad (29)$$

The reputation of an organisation in VO (from users' perspective) is the aggregation of the reputation of all users that participate in VO. According to (6), (29)

$$rep(vo\_id, t) = \frac{1}{\left|g^{-1}_{vo\_id}(o\_id)\right|} \sum_{r \in g^{-1}_{vo\_id}(o\_id)} rep(vo\_id, u, t) . \quad (30)$$

According to (2), (10), (29) the reputation of a user in all VOs can be estimated as follows

$$rep(r, t) = \frac{1}{\left|\mathrm{VO}\right|_u\right|} \sum_{vo\_id \in \mathrm{VO}|_u} rep(vo\_id, u, t) \quad (31)$$

Analysis of Attacks

In this section we will study different security threats scenarios in the area of trust and reputation management that were proposed in [15], and analyse how the proposed model responds to these attacks. It should be noted that some of these attacks can be handled by the existing mechanisms already implemented in Grid systems.

Individual malicious peers. Malicious peers always provide bad services. From Grid perspective, there can be either a resource that always provides unreliable services, or malicious user that always tries to harm a system. Such an unreliable resource will provide poor services to the users that will result in $v << SLA(u, r, vo\_id)$, and thus the reputation of this resource will be low. Moreover, such resource will get no or few number of tasks when reputation is incorporated into resource broker to schedule tasks among resources of Grid system. As to the malicious users, the output of the user behaviour model $SMUB_{(u,\ vo\_id)}(\mathbf{x})$ will be always low, close to 0. Thus, tasks submitted by the user could be cancelled.

Malicious collectives. This is a threat when malicious peers form a malicious collective. In Grids, there could be a user that tries illegally to improve the reputation of a particular resource. If the user and resource belong to the same organization that kind of behaviour will be captured by the alliance function $h(u, r)$. In order to improve the reputation value considerably the user will need to submit a lot of simple tasks. In such a case the reputation value of the resource will be bounded with the $s(type(r))$ function. If the user wants to keep illegally the reputation of

the resource at a high level over a long period of time, he or she will need to constantly submit tasks since the reputation is accumulated only over specific period of time. Such behavioural patterns will be detected by the SMUB model, and the user may be punished with a reputation for such actions.

Malicious collectives with camouflage. This is a threat which is not always easy to tackle, since its resilience will mostly depend on the behavioural pattern followed by malicious peers [15]. These correspond to the malicious collectives with the variable behaviour. In our model, such variability could be detected using SMUB model. Moreover, reputation value for such users will vary considerably over the time as well. Therefore, with such an approach it is possible to punish such wit the reputation.

But it is to be noted that patterns of user behaviour could be various, and it is very difficult to build a model that will capture all the possible situations. Thus, the success of detecting malicious collectives with camouflage greatly depends on the model of user behaviour.

Malicious spies. This a threat when malicious peers (spies) always provide good services when selected as service providers, but they also give the maximum rating values to those malicious peers who always provide bad services. In Grids this corresponds to the situation when a user with high reputation provides the maximum rating to unreliable resources. This type of threat was already described in *malicious collectives* subsection.

Sybil attack. An adversary initiates a large number of malicious peers in the system. Each time one of the peers is the system is scheduled as a resource provider it provides malicious service and after that it is disconnected and replaced by another peer [15, 20, 29]. In Grids, such an attack is hardly implemented since appropriate certificate should be obtained from certificate authority in order to integrate a resource into the Grid system. Another way to tackle this problem is to use the methodology of active experiment to monitor the availability of resources.

On the other hand, this attack may become a threat when Grids are integrated with the Sensor Web or sensor networks. Sybil attack implemented through the sensor web could cause harm to Grid entities. But security issues relating to the integration of Grids and Sensor Web are out of scope of this paper and should be investigated in the future works.

Man in the middle attack. A malicious peer can intercept the messages between other peers, rewrite the message and change reputation values. Our model relies on the existing Grid security mechanisms to tackle this attack [20].

Driving down the reputation of a reliable peer. Malicious peers give the worst rating to those benevolent peers, who indeed provide good services. Projecting on to the Grids, a malicious user will provide poor ratings to the resources, though user's tasks were completed successfully with appropriate QoS value. One possible way to tackle this problem is that tasks of users with low reputation value are never sent to resources with high reputation by the resource broker. Moreover, QoS metrics in Grids are measured by the monitoring system, and to change these values, and consequently reputation, the malicious user should crack it.

Partially malicious collectives. Malicious peers provide malicious actions for some kind of services, and, for others, they provide good services. In Grids, this threat corresponds to users with variable behaviour (covered in *malicious collectives with camouflage* subsection), and to resources that for some types of services provide poor. By just considering a different score for every service offered by a resource, this threat is mitigated most of the times [15]. That is why we included into our model a score function to provide different reputation values for different types of tasks executed by resources.

Malicious pre-trusted peers. Some models are based on the strategy that there is a set of peers that can be trusted before any transaction is carried out in the system (known as pre-trusted peers) [30]. Our model does not include the notion of pre-trusted peers.

Reputation Management System

Developed model could be used by VOs with reputation management system. Paper [8] presents the architecture of such reputation management system in order to facilitate the rating of both VO resources and VO users. The architecture is shown in Figure 2.

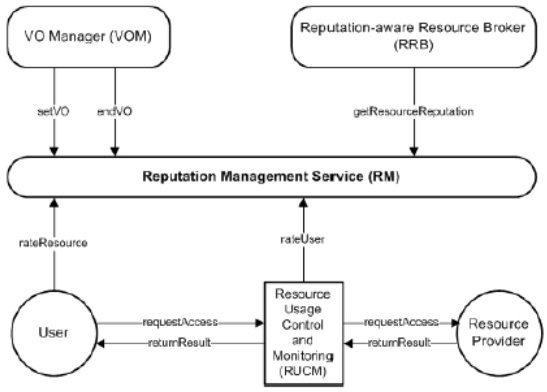

Fig. 2. Reputation management in VOs

Conclusions

In this paper we extended the existing utility-based reputation model [8] by incorporating a statistical model of user behaviour (SMUB) that was previously developed for computer networks and distributed systems, and different functions to address threats scenarios in the area of trust and reputation management that were proposed in [15]. These modifications include: assigning initial reputation to a new entity in VO, capturing alliance between consumer and resource, time decay function, and score function. We analysed the described reputation model against the existing attacks and how the model overcomes them following security threat scenarios presented in [15].

References

1. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // Int J of Supercomputing Applications. – 2000. – 15(3). – P. 200-222.
2. Grandison T., Sloman M. A Survey of Trust in Internet Applications // IEEE Communications Survey and Tutorials. –2000. – 3.
3. McKnight D.H., Chervany N.L. The Meaning of Trust. Technical Report MISRC Working Paper Series 96-04, University of Minnesota. Management Information Systems Research Center, 1996.
4. Gambetta D. Can We Trust Trust? In: D. Gambetta (ed). Trust: Making and Breaking Cooperative Relations. Department of Sociology, University of Oxford, 1988.
5. Josang A., Ismail R., Boyd C. A Survey of Trust and Reputation Systems for Online Service Provision // Decision Support Systems. – 2007. – 43(2). – P. 618-644.
6. Abdul-Rahman A., Hailes S. Supporting trust in virtual communities // In: Proceedings of the IEEE 33rd Hawaii Int Conf on System Sciences (HICSS '00). – 2000. – vol 6. – P. 6007.
7. Kerschbaum F., et al. A trust-based reputation service for virtual organization formation // In: Proceedings of the 4th Int Conf on Trust Management. Lecture Notes in Computer Science Springer. – 2006. – vol. 3986. – P. 193–205.
8. Arenas A., Aziz B., Silaghi G.C. Reputation management in grid-based virtual organisations // In: Fernandez Medina E., Malek M., Hernando J. (ed.) Proc. International Conference on Security and Cryptography (SECRYPT 2008), Porto, Portugal. – 2008. – P. 538-545.
9. Luke T.W.T., Jennings N.R., Rogers A., Luck M. A Hierarchical Bayesian Trust Model based on Reputation and Group Behaviour // In: 6th European Workshop on Multi-Agent Systems, Bath, UK. – 2008.
10. Shelestov A., Skakun S., Kussul O. Intelligent Model of User Behavior in Distributed Systems // Int J on Inf Theory and Applications. – 2008. – 15(1). – P. 70–76.
11. Shelestov A., Skakun S., Kussul O. Complex Neural Network Model of User Behavior in Distributed Systems // In: Proc. of XIII-th International Conference Knowledge-Dialogue-Solutions. Varna, Bulgaria. – 2007. – P. 42–49.
12. Skakun S., Kussul N., Lobunets A. Implementation of the Neural Network Model of Users of Computer Systems on the Basis of Agent Technology // J. of Automation and Inf. Sci. – 2005. – 37(4). –P. 11–18.
13. Kussul N., Skakun S. Neural Network Approach for User Activity Monitoring in Computer Networks // In: Proc. of the Int. Joint Conf. on Neural Networks, Budapest, Hungary. – 2004. – 2. – P. 1557–1562.
14. Azzedin F., Maheswaran M. Integrating Trust into Grid Resource Management Systems // In: First IEEE International Workshop on Security and Grid Computing. – 2002.

15. Gomez Marmol F, Martınez Perez G (2009) Security threats scenarios in trust and reputation models for distributed systems. Computers & Security 28: 545–556
16. Kussul N, Shelestov A, Skakun S (2009) Grid and sensor web technologies for environmental monitoring. Earth Science Informatics 2(1-2): 37-51
17. Menasce DA, Casalicchio E (2004) Quality of service aspects and metrics in Grid computing. In: Proc. 2004 Computer Measurement Group Conference, Las Vegas, USA.
18. Hong-Linh T, Samborski R, Fahringer T (2006) Towards a Framework for Monitoring and Analyzing QoS Metrics of Grid Services. In: Proc. Second IEEE Int Conf on e-Science and Grid Computing (e-Science'06).
19. Al-Ali R, von Laszewski G, Amin K, Hategan M, Rana O, Walker D, Zaluzec N (2004) QoS Support for High-Performance Scientific Grid Applications. In: Proc. IEEE International Symposium on Cluster Computing and the Grid 2004. (CCGrid 2004). pp 134–143.
20. Chakrabarti A (2007) Grid Computing Security. Springer-Verlag Berlin Heidelberg
21. Haykin S (1999) Neural Networks: A Comprehensive Foundation. Upper Saddle River, New Jersey, Prentice Hall
22. Minai AA, Williams RJ (1990) Back-propagation heuristics: A study of the extended delta-bar-delta algorithm. In: IEEE Int Joint Conf on Neural Networks vol I pp. 595-600
23. Werbos PJ (1994) The roots of backpropagation: from ordered derivatives to neural networks and political forecasting. John Wiley & Sons, Inc., New York
24. Bishop CM (2006) Pattern Recognition and Machine Learning. Springer
25. Shelestov A, Kussul N, Skakun S (2006) Grid Technologies in Monitoring Systems Based on Satellite Data. J of Automation and Inf Sci 38(3): 69–80.
26. Fusco L, Cossu R, Retscher C (2007) Open Grid Services for Envisat and Earth Observation Applications. In: Plaza AJ, Chang C-I (ed) High performance computing in remote sensing, 1st edn. Taylor & Francis Group, New York, pp 237-280.
27. Oh SH., Lee WS (2003) An anomaly intrusion detection method by clustering normal user behavior. Computers & Security 22(7): 596–612.
28. Wang W, Guan X, Zhang X, Yang L (2006) Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data. Computers & Security 25(7) 539–550.
29. Douceur JR, Donath JS (2002) The Sybil attack. In: Proceedings for the 1st international workshop on peer-to-peer systems (IPTPS ’02). pp 251–60.
30. Kamvar S, Schlosser M, Garcia-Molina H (2003) The EigenTrust algorithm for reputation management in P2P networks, Budapest, Hungary.

Поступила в редакцию 16.12.2009