Journal of Information Assurance and Security 1 (2008) 27-37

Managing Behaviour Trust in Grid Computing Environments

Elvis Papalilo and Bernd Freisleben

Department of Mathematics and Computer Science, University of Marburg,
Hans-Meerwein-Str. 3, D-35032, Marburg, Germany
{elvis, freisleb} @informatik.uni-marburg.de

Abstract: *In this paper, an approach for managing behaviour trust of participants in Grid computing environments is presented. The basic idea of the approach is to view the interaction process between Grid participants similar to an industrial production process, and use statistical methods of quality assurance to discover deviations in the behaviour of Grid participants in order to assess their behaviour trust. Simulation results are presented to demonstrate the feasibility of the proposed approach.*

Keywords: Grid computing, trust establishment, trust management, behaviour trust, quality of service, statistical methods of quality assurance.

1. Introduction

Grid computing environments are open distributed systems composed of autonomously operating participants that interact with each other using specific mechanisms and protocols to offer and/or use services (e.g. computation, storage, and bandwidth). In this respect, the challenges faced by today's Grids are:

- each of the participants represents different aims and objectives,
- participants can join and leave the environment at any time,
- a Grid participant can change its identity and re-enter the environment, thus avoiding punishment for any past wrongdoing, and
- the participants' capabilities are heterogeneous (i.e. there are different qualities for the same offered services).

Typically, participants do not have sufficient knowledge about their interaction partners in the environment. As a result, it is quite difficult to rely on the outcome of the interaction process.

Trust management mechanisms are a promising solution for strengthening the confidence in the quality of the interaction process between Grid participants. We define trust in Grid environments as *the extent to which every participant in a Grid environment, in a specific moment of time, with an evidence of relative security regarding the identity and the behaviour of their counterparts, is willing to interact with them, even though unexpected negative outcomes could result from the entire interaction process.*

In previous work [1], we have presented a probabilistic trust model for both the identity and the behaviour of the interaction parties. In this model, trust values are accumulated and calculated based on past direct reciprocal interactions and/or indirect interactions. Each of the participants continuously monitors the behaviour of their partners during an active interaction, and the monitoring process is configured based also on the properties of the running application. However, it is still difficult to discover the "real" behaviour of a collaboration partner from the "observed" behaviour. There could have been deviations that have skipped the monitoring process, making the partner somehow "not trustworthy". Furthermore, there is no framework within which the deviating behaviour of a partner is going to be tolerated.

In this paper, an approach for managing the behaviour trust of Grid participants is presented. Statistical methods of quality assurance for identifying the "real" behaviour of a participant during an interaction and for "keeping" the behaviour of the participants "in control" are used. If the behaviour of a participant is "out of control", then this participant:

- cannot be used as an interaction partner for certain applications, because the expected behaviour and the trust requirements were not met, but the participant could be still considered for other applications with "moderate" trust requirements, or
- can not be considered anymore for further interactions, independent of the expected behaviour and the trust requirements of applications.

Simulation results are presented to demonstrate the feasibility of the proposed approach.

The rest of the article is organized as follows. In section 2, it is explained what the behaviour of Grid participants is and how behaviour trust is measured. Furthermore, a list of possible threats to trust that Grid participants establish between them is presented. In section 3, our view on how behaviour trust is established and managed among Grid participants is given. Section 4 presents our model for managing (behaviour) trust in Grid environments. A system architecture that supports the behaviour monitoring model is presented in section 5. Section 6 evaluates the performance of the behaviour verification strategies used. Using the GridSim simulator [2], some collaboration scenarios, with different verification and error frequencies are implemented. Section 7 discusses related work on the behaviour of Grid participants and behaviour trust. Section 8 concludes the paper and outlines areas of future research.

Received January 7, 2008.

2. Behaviour Trust of Grid Participants

2.1 The Problem of Behaviour Definition in Grids

In the literature, the behaviour of collaborative parties in Grid environments remains an abstract notion. Participants behave either “good” or “bad”. In most cases, “good” behaviour reflects the expectations of a participant to simply receive a response from another participant involved in an interaction or sometimes to get accurate results. If an interaction party behaves differently from these “normal” expectations, its behaviour is labelled as “bad”. Participants with “good” behaviour are considered as trusted ones and are thus eligible for future interactions. Participants with “bad” behaviour have only minor or no possibility to be considered for further interactions.

To support a flexible behaviour management and classification system, additional mechanisms are necessary, e.g. splitting behaviour in detailed elements, observing them continuously and offering the possibility for behaviour classification.

2.2 Behaviour Trust and Quality of Service

Depending on the field of application, users recognize usability in terms of different aspects of Quality of Service (QoS). In Grid environments, usability is an important factor [3]. Hence, it is meaningful to investigate the relationship between QoS and the behaviour of participants in Grid environments.

QoS refers to the ability of a Grid system/participant to provide network and computation services such that each user’s expectations for timeliness, quality and performance are met. There are several dimensions of QoS described in the literature [4], e.g. parameters like accuracy, precision and performance. To support a QoS dimension, users request or specify a level of service for one or more attributes of these dimensions, and the underlying control mechanisms should be capable of delivering these services at the requested QoS levels.

QoS deals with a range of expected behaviours of individual participants which only as a whole define the completion of the service a user (or an application) demands. In this context, it is important to map a user’s expectations and preferences to the system parameters and capabilities.

Trust is the most important social element in Grid systems that can be defined as having the confidence that an interaction party will offer the desired QoS, behaving as expected. Trust management is the process of deciding what entities are to be trusted to complete particular actions, and if the interested participant can be allowed to use the services offered or not. A trust system for Grid environments should offer flexible and easy to use components that can be configured to the specific needs of a user according to the application requirements.

Abstracting the common attributes from the variety of demands that the user, aiming at an optimal level of QoS, places on the participants in the environment, the components of behaviour trust could be derived from the parameters of QoS like: reliability (correct functioning of a service over a period of time), availability (readiness for use), accessibility (capability of responding to a request), cost (charges for services offered), security (security level offered), performance (high throughput and lower latency), etc.

Each of these parameters can be directly measured or broken up in measurable elements, in order to offer the possibility to create a history with data from past interactions among collaborating parties in a Grid environment:

- *Availability*: measured as the ratio of the number of successful contacts to the provider (service) and the total number of requests. A non-available service is implied to be not accessible.
- *Accessibility*: measured as the ratio of number of successful service “ready-state” responses and the total number of requests.
- *Accuracy*: measured as the ratio of total number of correct responses received from a provider (service) and the total number of responses from that provider (service).
- *Response time*: time between sending a request to a service and receiving a response from it.
- *Latency*: intended to measure the speed with which a service can process a given request. Possible measurements can be conducted using the time when the request reached the service and the time when the service finished processing the request.
- *Throughput*: number of concurrent requests handled by the provider of the services.

Fig. 1 presents a view of the behaviour trust elements considering different roles (consumer or provider) that the participants play at certain moments of times.

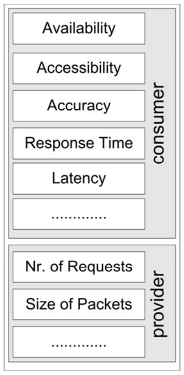


Figure 1. Behaviour trust elements

By analyzing the history of the collected data regarding the above behaviour trust elements using statistical methods of process control, and by also considering the personal experience of each of the participants together with their preferences and their personalized notion of normal or anomalous behaviour, it is possible to automate the classification of the behaviour of participants in Grid environments.

2.3 Behaviour Trust

Every time a *trustor* (the party that trusts) collaborates with a *trustee* (the party that is trusted), a direct experience is established between them. The output of the collaboration determines the type of experience the trustor had with the trustee and thus the *trust on the behaviour* of the trustee. This is known as *direct behaviour trust*. It is calculated based on the entirety of the behaviour elements under observation. Considering the participants *X* as trustor and *Y* as trustee, direct behaviour trust is expressed according to formula (1):

$$T_X^B(Y) = \prod_i B(Y)_i \qquad (1)$$

where $B(Y)_i$ represents the behaviour trust elements under observation. Each of the behaviour trust elements is calculated as the number of "positive" observations divided by the total number of observations during the collaboration, as generalized in formula (2):

$$B(Y) = \frac{"positive"\_observations}{total\_observations} \qquad (2)$$

2.4 Behaviour Trust Threat Profiles

Trust constitutes the only considerable currency in Grid environments where its participants have to rely on. Reliability on trust information is the confidentiality that one should have regarding the offered experience and the current behaviour of others in the environment. The two primary types of adversaries in Grid environments, able to put reliability of trust at risk, are selfish and malicious participants. They are primarily distinguished by their goals in the environment. Selfish participants wish to have a considerable profit for their mediocre contribution or achieve a better "social" position in the environment to the detriment of the other participants. The goal of malicious participants, on the other hand, is to cause harm to either specific targeted participants in the environment or to the environment as a whole.

To accomplish their goal, both types of participants are willing to exploit any vulnerability [5] and any type of coalition with other participants [6]. The following trust threat profiles help us to identify the specific threats that put the reliability of trust information at risk in Grid environments.

2.4.1 Abusive/Malicious "Gossiping"

In a Grid environment, participants can exchange their personal direct experiences. This should not be seen as an obligation for the participants, but merely as a possibility to exchange information and thus helps to reduce the level of the uncertainty in the environment. Each of the participants should independently decide whether to consider this kind of information and at what degree.

The possible threat to the trust information offered to participants is abusive or even malicious "gossiping" from selfish or malicious participants with the sole aim of:

- discrediting participants in the environment - For certain targeted participants or for everybody else in the environment, low trust values are offered to interested parties, or
- supporting certain target participants for an undeserved profit - Higher trust values, i.e. greater competences are offered regarding certain target participants in the environment.

2.4.2 Deceiving Trust

As previously stated, a participant can gather information and learn the behaviour of its partners over a number of direct interactions. In this case, the participant reasons about the outcome of the future interactions with these participants. This trust information can be abused by selfish or malicious participants. The following threats to this type of information can be identified:

- From "High" to "Low" – A higher level of trust gives some assurances on the competence and the reliability of the target participant. The existence of this general principle in Grid environments has some disadvantages. Selfish participants can use it in order to deceive their interaction parties. At the beginning, they could fulfil the expectations of their interaction parties and offer services of high quality, reasons for which they were chosen among the others in the environment, but as soon as they have reached a "high social position" in the environment, they start act differently by lowering the quality of their offered services.
- Trust Manipulation – In current Grid environments, for each of the participants it is easy to change or manipulate their identity information. Current technology offers the possibility to the participants to identify their interaction partners, but no assurances on their real identity. As a result, suspicions on the behaviour and the intentions of single participants exist. This problem becomes serious especially in cases when malicious participants impersonate the identity of highly trusted participants and try to collaborate with others in the environment.
- Stealing Trust Information - In general, Grid systems are vulnerable to all typical network and computer security threats and attacks. Furthermore, the implementation of Web Service technology into the Grid [7] will bring a new wave of threats, in particular, those inherited from XML Web Services.

The nature of the trust information saved (direct experiences) can be considered as valuable and thus as an attractive target for malicious participants (e.g. a list of most trusted participants in the environment could be extracted in order to attack them or hinder their normal activity). Encryption mechanisms could help further for securing the communication between Grid participants [8].

3. Establishing and Managing Trust Among Grid Participants

A high degree of trust in a participant means that it is likely to be chosen as an interaction partner. Conversely, a low degree of trust suggests that the participant cannot be selected anymore, especially in the case when other, more trusted interaction partners are available. In this way, the proposed trust model aims to guide a participant's decision making process regarding how, when, and who to interact with.

3.1 Trust Management

There are a number of ways for a Grid participant to establish trust with its counterparts. First, it can interact with the target participant(s) and learn its/their behaviour over a number of interactions. In this case, the participant reasons about the outcome of the direct interactions with the others. When an interaction with a new participant is started, i.e. when no information on previous behaviour exists, it can use its beliefs about different characteristics of these interaction partners and reason about these beliefs in order to decide how much trust should be put in each of them. Furthermore, the participant could ask others in the environment about their experiences with the target participant(s). If sufficient information is obtained and if this information can be trusted, the participant can reliably choose its interaction partners.

3.2 Trust Relationships

Trust relationships are modelled as directed graphs where trust is a unidirectional directed edge from the trustor to the trustee.

A distinct feature of the trust relationships is their dynamicity. According to the observed behaviour during a collaboration, in case of undesired or unexpected behaviour from the other party, participants can decide on the future of the current collaboration (or future collaborations) with that partner.

The following trust relationships are considered:

- **consumer – provider (provider - consumer):** participants trust that their counterparts will behave properly during the collaboration. This *belief* is constructed based on the observations of past collaborations and/or experiences of others. At the same time, the belief also expresses the *expectations* of the parties that their partners will show at least the same behaviour as in previous direct or indirect collaborations. Consumers expect providers to supply services on the desired level of quality, and providers expect their consumers to behave accordingly. This relationship is *bilateral, i.e.* both parties have to trust each other (not necessarily at the same level) for the interaction to take place.
- **consumer (provider) – recommenders:** this kind of trust relationship differs from the trust relationship established between consumers and providers. The experience the participants have with their counterparts during single collaborations is considered to be "personal". It can be freely *offered* as recommendation to others in the environment that *ask* for it, without establishing any trust relationship in this direction. The trust relationship exists only from the side of the party that needs these recommendations. Every participant can ask the others if recommendations are needed, but not necessarily fully consider them. The reasons are:
    - since the experience the participants make in the environment is personal, "good" or "bad" experience made from one partner does not necessarily have the same meaning for the others, and
    - malicious participants could intentionally offer low trust values for good behaving participants and high trust values for others with mediocre or no contribution at all.

In our model [1], recommendations depend:

- on the user/application specification of the trust requirements (if any recommendation is going to be considered at all, or what trust value to assign them) and
- on a history of the past recommendations and the resulting behaviour of the participants recommended by them.

Let us consider the relationship of a Grid participant to the others in the environment as presented in Fig. 2.

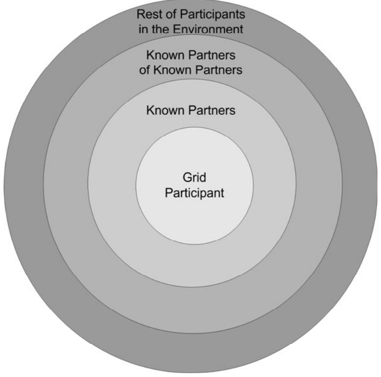

Rest of Participants in the Environment
Known Partners of Known Partners
Known Partners
Grid Participant

Figure 2. Relationship between Grid participants

The recommendations could be received from the participant itself (direct experience) $T^{B}_{X,D_{collaboration}=0}$, recommendations from known partners (known sources of information) $R_{X,D_{collaboration}=1}$ and recommendations from unknown participants in the environment $R_{X,D_{collaboration}>1}$ ($D_{collaboration}$ represents the distance in hops between the participant and its recommenders). Furthermore, there exists a separation of behaviour trust and recommendation trust. The "ability/inability" of a participant to offer valuable recommendations should in no way interfere with decisions regarding its "capability" to properly behave during the collaboration (consumer-provider; provider-consumer) with another participant.

3.3 Trust Requirements

The overall decision whether to trust an interaction partner or not may be affected by other non-functional aspects that cannot be generally determined for every possible situation, but are rather considered to be under the control of the user(s) when requesting such a decision. In addition, while the basic functionalities of two Grid applications could be similar, differences in application behaviour could be caused by different domain specific trust requirements. Therefore, flexible and easy to use components that can be tailored to the specific trust requirements of a user for each application must be offered.

The user's trust requirements include:

- initialization values that the user is willing to assign to each of the new partners.
- behaviour trust element(s) of interest.
- selection of sources for getting trust information from (e.g. recommendations).

- verification strategies - the user may choose to *"trust-no-one"* (verify the accuracy of every single response), or apply *"optimistic-trust"* (a minimal verification frequency with values in [0, 1] is selected).
- clearance number "*i*" - establish the number *i* of responses to be verified sequentially (100% verification at the beginning of the collaboration).
- stopping rule "$nc_{BhvT\_element}$" - the tolerance that the user has against nonconformities of the behaviour of the other party.

4. Behaviour Trust and Statistical Methods of Quality Assurance

Considering different sources for gathering trust information from (self experience, indirect experience, user/application trust requirements), each participant sorts out the collaboration partners and starts interacting only with the "most trusted" of them. During the collaboration, the behaviour trust elements are verified either with 100% or with a certain verification frequency [1]. According to the verification results, the trust values are updated and influence the decision making process whether the collaboration with a certain participant will continue or will be interrupted. Clearly, if the expected behaviour was met, the collaboration will most probably continue. The problems start once any deviation from the expected behaviour of a collaboration partner is recognized.

The behaviour deviations are:

- deviation within the current collaboration and
- deviation over time.

The first type of deviation has a more immediate effect on the current collaboration and the validity of the data being processed. If a 100% verification strategy is applied, it is easy to tell that until that specific moment, no other deviation has happened. On the contrary, if a verification frequency is applied, it is not possible to tell that no more deviations except those verified existed. The question is *whether there is any difference between the calculated behaviour trust and the real behaviour of a participant.*

The second type of deviation affects more than the current collaboration and deals with the trustworthiness of a participant in general. For example, consider a participant offering a high processing speed for the tasks assigned, and also applying high charges for the offered services. At the beginning, this participant behaves according to the expectations of its partners. As time passes, it starts trying to maximize its profit by offering the service to a larger number of customers, affecting the processing speed negatively, but applying the same charges. The question is *how to detect these "long term" deviations of the behaviour of a participant.*

Another question that arises either in the case when deviations are observed within a single collaboration or in the case when deviations are observed over time, is: *how long should a collaboration continue with a participant despite its anomalous/malicious behaviour?*

To answer these questions, new functionalities need to be added to the trust model. For this purpose, the use of statistical methods of quality assurance is proposed. The basic idea is to view collaboration(s) among participants as a "production process" where the behaviour trust elements under observation establish the "quality" of the collaboration process.

4.1 Sampling and Sampling Distribution

Statistical methods for monitoring and improving the quality of manufactured goods have been around since the early 1920s when W. A. Shewhart introduced the graphical control chart method for detecting possible problems in manufacturing processes [9].

The term "quality" is broadly used by service industries and embraces all the characteristics of an entity (goods or services) that determine the capacity to satisfy the expressed and implicit requirements of who uses it.

Current applications of statistical methods of quality assurance have widened to include many service industries as well as traditional manufacturing applications.

General aspects of quality are:

- The quality of output process. Goods and services are produced with various degrees of quality.
- The conformity to already set process regulations. This aspect refers to the adherence of the product to specifications and tolerances assigned to it in the planning phase.

Every output possesses a number of somehow measurable elements which contribute jointly to the formation of the quality of the product. These elements can be indicated as quality characteristics or quality parameters. Quality parameters are evaluated in comparison to the specifications or the established values for any of the quality parameters of the product/service. The desired value of the quality parameters is defined as nominal value or target value.

The primary goal of statistical quality assurance is to draw conclusions about the fulfilment of a quality standard in a population based on information about individual units. From the statistical point of view, the quality standard of an individual unit is related to the specific realization of its quality parameter, and the quality standard of the population (the entire output) is related to a function parameter, or a functional parameter of the distribution of the quality parameter. The population consists of a finite or infinite collection of elements where the sample(s) to be verified are taken from. A random sampling procedure is the procedure of selecting a finite number of units from a population through a random mechanism.

4.2 Continuous Sampling Plans

The aim of a continuous sampling plan (CSP) is to control the verification process depending on the verification results in such a way that the maximum of the average outgoing quality (AOQ) does not exceed a specified limit. AOQ can be defined as the fraction of "defective/non-conforming" entities which are not detected through the verification process with respect to the total number of processed entities.

In terms of the plan parameters:

$$AOQ(P \mid i;k) = \frac{(k-1)P(1-P)^i}{1+(k-1)(1-P)^i} \qquad (3)$$

where for $P = 0$ or $P = 1$: $AOQ(0 \mid i;k) = AOQ(1 \mid i;k) = 0$.

P is the fraction of defective entities discovered during the verification process with respect to the total number of entities verified and k is the sampling interval where one entity can be picked up from for verification. Considering the frequency of the verification presented in [1], it can be calculated according to:

$$k = \frac{1}{-((1 - V_{\min}) \cdot T^{B}_{last}) + 1} \quad (4)$$

where $V_{\min}$ is the minimal verification rate, set by the trustor and T^{B}_{last} represents the trust value of a trustee at a certain moment of time.

Execution of a CSP for any of the behaviour trust elements follows the following algorithm:

1. initialize the variables for the number of entities (responses, results etc.) verified through 100% inspection v_i = 0, number of the entities to be verified through the frequency of verification in (4) v_k= 0 and number of defective entities found d_f= 0;
2. $v_i = v_i + 1$, verify the v_i-*th* entity;
3. if defective, then $d_f = d_f + 1$;
4. if $d_f <= nc_{BhvT\_element}$, repeat from step 2 until $v_i = i$, otherwise interrupt collaboration and do not consider the received responses from that participant;
5. $v_k = v_k + k$, verify the v_k-*th* entity;
6. if defective, then $d_f = d_f + 1$;
7. if $d_f <= nc_{BhvT\_element}$, repeat from step 5 until no more tasks are left, otherwise interrupt collaboration and do not consider what is received from that participant.

Figure 3. Continuous sampling plan

4.3 "In-Control" Behaviour of Grid Participants

The underlying concept of statistical process control is based on the comparison of current process' output with the previous outputs. These data are used to calculate the control limits for the expected measurements of the output of the process. Data from the running process is collected and is compared to the control limits. The majority of measurements are supposed to lie within the control limits. Data that fall outside the control limits are examined and perhaps will later be discarded. If this is the case, the limits are recomputed, and the process is repeated.

There are several ways how to implement process control. From the key monitoring and investigating tools we make use of control charts.

Control charts consist of:

- Center line, at the average of the statistic by default:

$$CL = \bar{p} \quad (5)$$

- Control Limits (Upper Control Limit (UCL) and Lower Control Limit (LCL)):

$$Control\_Limits = \bar{p} \pm 3\sqrt{\frac{\bar{p}(1-\bar{p})}{\bar{n}}} \quad (6)$$

where $\bar{n}$ is the mean value of all sample sizes and $\bar{p}$ is the mean value of defective entities found in all sample sizes.

In general, a process is considered as statistically stable over time (with respect to the parameter under observation) if the distribution of this parameter does not change over time. Stability makes it possible to predict the range of variability to expect in the parameter in the future. A parameter is one of the behaviour trust elements mentioned above. Each of the Grid participants keeps this element for each of their interaction partners under continuous "observation". An example is the accuracy of the responses coming from interaction parties during the interaction. Our goal is to discover the fraction of defective responses coming from a participant over a period of interest (e.g. as long as the interaction takes place). Samples of measurements are periodically taken at one or more stages during the interaction.

If the observed behaviour trust element is "in control", the fluctuations are expected to lie around the common mean (centre line). If it is "out of control", the mean changes and flips outside the control limits. We consider as "out of control" only fluctuations outside the upper control limit (UCL), because this indicates an increase of the non-conforming behaviour in comparison with the behaviour the participant exhibited until that particular moment of time.

The steps to follow for constructing the control charts for behaviour trust elements are:

1. Gather the data for the period of interest.
2. Calculate the centre line.
3. Calculate the control limits.
4. Verify if data lie within the control limits.
5. Classify the behaviour of the participant.

Figure 4. Control charts for behaviour trust elements

We make a finer separation regarding the types of behaviour that a Grid participant exhibits. In a social environment where interaction among participants is established based on interpersonal relations, there are differences in the individual expectations that each of the participants has for the behaviour of its interaction partners. As a consequence:

- if the observed behaviour lies on CL or between CL and LCL, the participant behaved as expected. Some anomalies were observed, but the trustor's fault-tolerance was not exceeded;
- if the observed behaviour lies between UCL and CL, the number of the observed anomalies has most

probably exceeded the trustor's fault-tolerance and the current collaboration was interrupted. However, for moderated trust requirements (i.e. a greater fault-tolerance), the participant can still be considered for future interactions since the anomalous behaviour lies within the expected limits;

- if the observed behaviour lies outside the UCL, then the participant is banned and not considered anymore for future interactions.

5. Managing Collaboration Among Grid Participants

The notions discussed are summarized in the following model. The underlying trust system is in charge of monitoring the interaction and the behaviour of the interaction partners according to the strategies suggested by the user.

In our approach, behaviour trust data are collected using the system architecture presented in [1]. An extension to this system architecture is shown in Fig. 5.

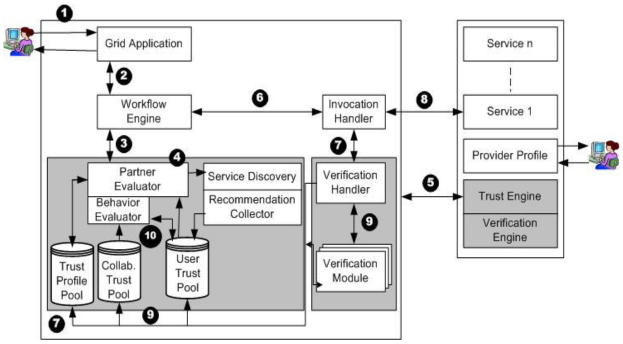


Figure 5. Architecture of a Grid system supporting our trust model

The system consists of two main components, the *trust engine* and the *verification engine*. The trust engine manages trust values and offers partner discovery and rating functionality to higher level applications, such as workflow engines or job scheduling systems. The verification engine handles the verification of behaviour trust elements and generates the necessary feedback for the trust engine regarding the specific interaction partner.

The user specifies his or her trust requirements along with the input data (step 1) using the workflow engine of the remote Grid system (step 2). A list of potential partners is obtained from the trust engine (step 3) after discovering the most suitable partners (step 4) with the help of the recommendations obtained from the others (step 5).

Invocation of the partners is then delegated to an invocation handler (step 6), which consults the verification engine (step 7) for synchronizing the distribution and verification strategies. The selected partners carry out the assigned services, and results are then collected by the invocation handler (step 8) and verified through the verification engine, using a strategy and verification module consistent with the user supplied trust profile (step 9). Verification values are stored in the trust pool (TP) and collaboration trust pool (CTP). TP stores the history of all (past) observations regarding each of the interaction partners, and CTP stores values regarding the current interaction. TP forms the so-called user "personal experience", regarding past interactions with other participants in the Grid environment.

The verification values are passed to the behaviour evaluator module which classifies the behaviour according to the algorithm in Fig. 4, using the values stored in TP and CTP (step 10).

The overall result of this process is then passed to the workflow engine that collects results for the application to present them to the end user.

6. Evaluation of the Performance of the Verifications and of the Sampling Plans

The aim of our experimental work is to evaluate the performance of the trust model:

- First, the mean absolute error regarding the real behaviour shown by a participant to the observed behaviour (either through applying the verification model only or making a double check through applying the statistical model) will be measured. For this purpose, the following formula is used:

$$MAE = \frac{\sum_{i=1}^{n} | Bhv_{observed}(i) - Bhv_{real}(i) |}{n} \qquad (7)$$

  where n is the number of sequential experiments.

- Second, the behaviour of a participant through different sequential experiments will be monitored to show possible fluctuations in the current behaviour of a participant compared to the behaviour previously shown.

To evaluate our approach, we have modelled a set of resources and users using GridSim [2]. Although GridSim already offers a simulation infrastructure where one has the possibility to specify users and resources as separated entities, we adapted and extended GridSim to reflect our view of a Grid environment:

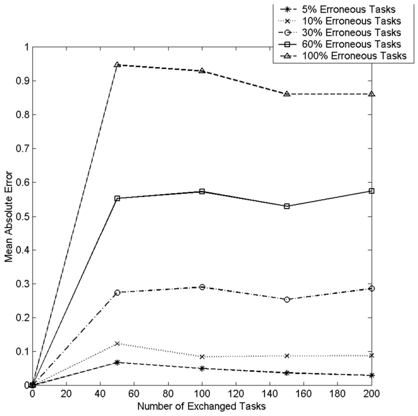
- Each user creates/owns a resource. The idea is to show that in real Grid environments, participants could play both roles; consumers and providers of services.
- Obviously, each user sends his/her tasks to every resource in the environment (considered to be suitable), except for its own resource.
- The behaviour trust elements considered are the accuracy of the responses coming from the different providers, their availability, accessibility, and speed of processing.

In total, 25 users/resources were created. Each of the users has different values for *deadline* and *budget*. Resources offer different processing *speeds* and charge different amounts of *Grid$* for their services offered.

6.1 Evaluating the Performance of the Verification Model

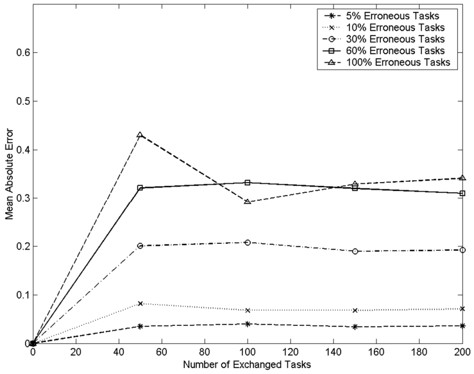
In the first group of experiments, the behaviour trust element under observation is the *accuracy* of the responses coming from a provider. As minimal verification frequencies, the values V_{min}= 5% and V_{min}= 30% are considered. The initial trust applied for the "unknown" participants is 1.0. The number of tasks varies between 50 and 200 tasks. The trustee randomly introduces errors in the responses it sends back to the trustor. The error frequencies vary between 5%, 10%, 30%, 60% and 100% of the tasks. The trustor also sets the "clearance number" (number of tasks to be verified sequentially). For the experiments presented here, the "clearance number" *i* (100% verification at the beginning, see Section 3.3) is 0 and 75. The verification takes place as the collaboration between parties continues.

Clearance number 0. The performance of the verification process for this verification strategy is shown in Fig. 6 and in Fig. 7.



V_{min}= 5%

Figure 6. Mean absolute errors for initial verification frequency 5%

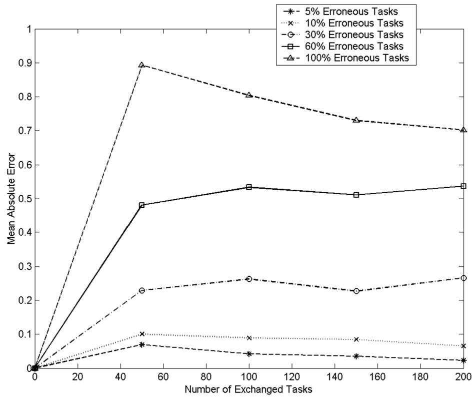


V_{min}= 30%

Figure 7. Mean absolute errors for initial verification frequency 30%

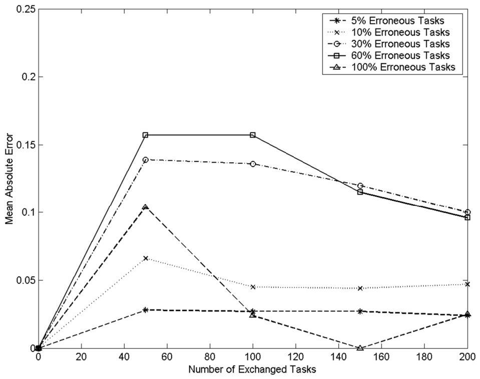
In the figures it can be seen that the mean absolute error diminishes as the minimal verification frequency increases. The average quality level (AOQ) was calculated according to (3), after the verification took place. Since the sampling interval *k* (4) varied according to the last updated trust value, a *mean value* for the sampling interval is used. The "errors" found with this method were added to the previously found "errors" during the verification process. The fraction of the "total" number of erroneous tasks with respect to the total number of tasks coming from the single trustees was calculated to determine the behaviour shown by them.

The performance of the verification process for this verification strategy is shown in Fig. 8 and in Fig. 9. The decrease of the mean absolute error, as a result of the doubled verification, is easily identifiable (comparing it also to the mean absolute error presented in Fig. 6 and in Fig. 7, respectively). For a minimal verification frequency of 5%, the improvement of the observations is about 6% (as the result of a small verification frequency) and for a minimal verification frequency of 30%, the observed improvement is over 50%.



V_{min}= 5%

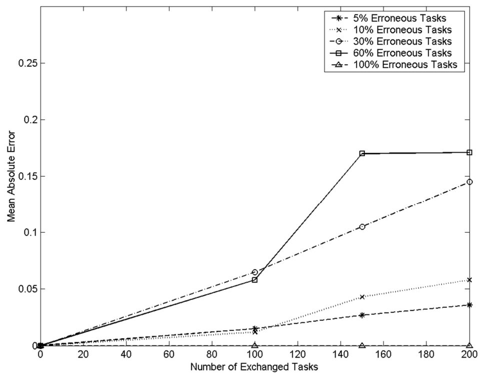
Figure 8. Mean absolute errors for the statistical verification method and initial verification frequency 5%



V_{min}= 30%

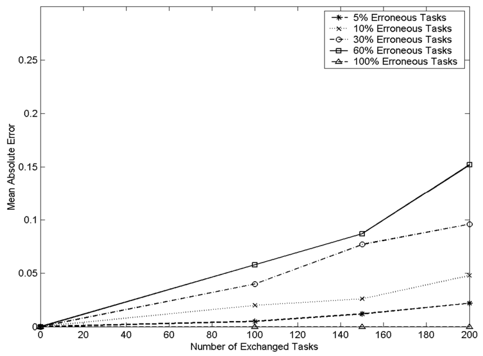
Figure 9. Mean absolute errors for the statistical verification method and initial verification frequency 30%

Clearance number 75. The performance of the verification process for this verification strategy is shown in Fig. 10 and in Fig. 11.



V_{min}= 5%

Figure 10. Mean absolute errors for initial verification frequency 5%



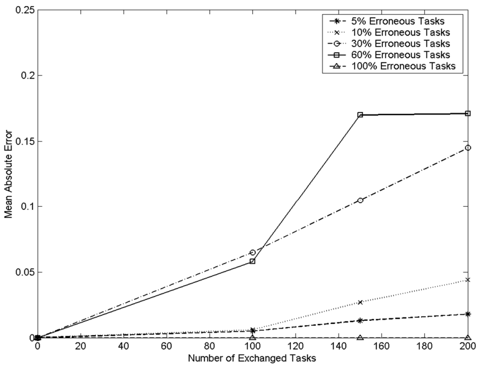
V_{min}= 30%

Figure 11. Mean absolute errors for initial verification frequency 30%

The experiments indicate that the behaviour a trustee showed while sequentially verifying its responses (clearance number equal or greater than 75) corresponds to the behaviour it showed during the entire collaboration. Thus, no oscillations were observed after the sequential verification took place.

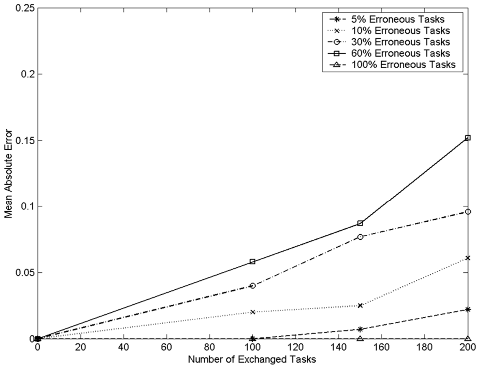
The performance of the verification process for the statistical verification strategy is shown in Fig. 12 and in Fig. 13.

It is evident that the higher the clearance number (as in this case 75 or higher), the more similar is the performance of the statistical model with the performance of the verification model only.



V_{min}= 5%

Figure 12. Mean absolute errors for the statistical verification method and initial verification frequency 5%



V_{min}= 30%

Figure 13. Mean absolute errors for the statistical verification method and initial verification frequency 30%

To summarize, the simulation results showed that:

- although errors were found when applying the verification strategy, a gap between the observed behaviour and the real behaviour of a trustee still exists. Applying the statistical model helps for having a better view on the errors that could have skipped the verification process. The re-evaluated behaviour for a provider is near to the real behaviour that the provider exhibited, and
- the higher the clearance number, the more similar the performance shown by the statistical model with the performance of the verification model only will be (as a result of the sequential verification taking place at the beginning of the collaboration).

6.2 Evaluating the Performance of the Sampling Plans

Additional simulations were conducted to observe the behaviour of the providers (resources) between experiments. This time, not only the *accuracy* of the responses coming from a provider but also its *availability*, *accessibility* and *speed of processing* were considered.

Behaviour trust is calculated as the product of all these single behaviour trust elements. It is used to observe the conformity of the behaviour a provider showed during a collaboration, with the expectations a user built based on the results gathered from previous simulations.

The CL, UCL and LCL were measured for different providers (resources) according to (5) and (6). The graphical representation is shown in Fig. 14, Fig. 15 and Fig. 16.

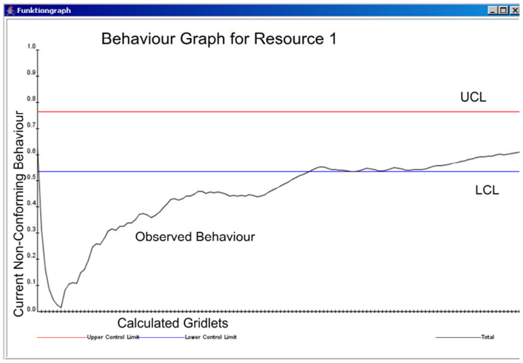


Figure 14. Monitoring the behaviour of “Resource 1”

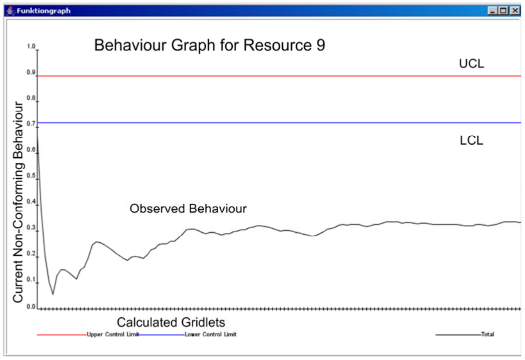


Figure 15. Monitoring the behaviour of “Resource 9”

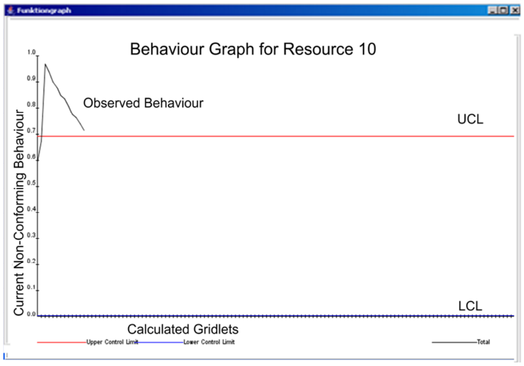


Figure 16. Monitoring the behaviour of “Resource 10”

It can be observed that:

- “Resource 1” fulfilled the expectations of its trustor. The number of the introduced (“real”) and discovered errors remains within the user’s *fault tolerance* (allowed errors).
- “Resource 9” behaved better than expected by its trustor.
- “Resource 10”, introduced many errors during the current collaboration. The number of discovered errors exceeded both the user’s *fault tolerance* and its expectations regarding the behaviour of this partner. The collaboration with this partner was interrupted and other partners were contacted. Furthermore, “Resource 10” was added to a “black list” and not considered during the subsequent simulations.

7. Related Work

In Grid environments, the general notion of behaviour of collaborating parties is considered by making use of behaviour trust management systems. Azzedin et al. [10] present a formal definition of behaviour trust where for trust establishment among entities their “experiences” together with a decay function that reflects the possible decays with time are considered. In their model, behaviour trust is limited to a general abusive or abnormal notion of behaviour of the participants during the interaction.

Lin et al. [11] use the belief, disbelief and uncertainty to weight the trustworthiness of the collaborating parties. The authors deal with a general notion of behaviour trust that is established before interaction takes place among participants.

Wu et al. [12] propose a system for detecting, classifying and controlling malicious Grid-abuse attacks. They deal with this problem using a source-based approach, system calls are analyzed through statistical methods to distinguish between attack programs and normal ones.

Finally, Teacy et al. [13] develop a probabilistic approach for managing behaviour trust in agent-like Grid systems. They concentrate on the accuracy of the trust values coming from third parties (third parties’ experience). A participant is considered as trustworthy only if it has a high probability of fulfilling its obligations during the interaction. In their work, they assume that the agents do not change their behaviour; this is a disadvantage of this model.

To the best of our knowledge, none of the approaches considers the questions posed at the beginning of section 4. There is no specification on what the behaviour of Grid participants really is, and few metrics regarding the measurement of behaviour trust are offered. Only a general notion for the behaviour of the participants is specified, leaving out different contexts that determine the behaviour of Grid participants.

8. Conclusions

In this article, we presented a methodological approach for monitoring and managing the behaviour of participants in Grid environments through the use of statistical methods of quality assurance.

Through a “proof of concept” implementation of the model and different simulations, we showed that when applying our statistical model, the re-evaluated behaviour of a participant after the verifications is very near to the “real” behaviour that the participants exhibited during the interaction.

There are several issues for future work. For example, other behaviour trust elements should be considered during future work, together with more complex scenarios. The aim is to evaluate the effects that trust has in Grid environments and the performance of every single participant together with the efficiency of our trust model in the face of more elevated and intensive threats.

Acknowledgments

This article is an extended version of the paper "Managing Behaviour Trust in Grids Using Statistical Methods of Quality Assurance" published in the Proceedings of the Third International Symposium on Information Assurance and Security (IAS 2007) held at Manchester, UK. We want to thank the anonymous reviewers and the organizers of the conference, especially Dr. Ning Zhang and Dr. Ajith Abraham. Finally, this work is financially supported by BMBF (D-Grid Initiative), by DFG (SFB/FK 615, Teilprojekt MT).

References

[1] E. Papalilo, T. Friese, M. Smith, B. Freisleben: "Trust Shaping: Adapting Trust Establishment and Management to Application Requirements in a Service-Oriented Grid Environment". In *Proceedings of the 4th International Conference on Grid and Cooperative Computing (GCC), Beijing, China*, pp. 47-58, 2005.

[2] GridSim. Available: http://www.gridbus.org/gridsim.

[3] I. Foster, C. Kesselman: *The Grid2: Blueprint for a New Computing Infrastructure,* Morgan Kaufmann, 2004.

[4] A.S. Ali, O. Rana, D.W. Walker: "WS-QoC: Measuring Quality of Service Compliance". In *Proceeding of the Second International Conference on Service-Oriented Computing Short Papers (ICSOC), New York, USA*, pp. 16-25, 2004.

[5] P. Lindstrom: "Attacking and Defending Web Services". In *http://www.forumsystems.com/papers/Attacking_and_Defending_WS.pdf*. 2004.

[6] D. De Roure, N. Jennings, N. Shadbolt: "Research Agenda for the Semantic Grid: A Future E-Science Infrastructure". In *www.semanticgrid.org/v1.9/semgrid.pdf*, 2001.

[7] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich: "The Open Grid Services Architecture". In *http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf*, 2005.

[8] E. Papalilo, B. Freisleben: "Combining Incomparable Public Session Keys and Certificateless Public Key Cryptography for Securing the Communication Between Grid Participants". In *Proceedings of International Conference on Grid Computing, High-Performance and Distributed Applications (GADA'07), Vilamoura, Algarve, Portugal. R. Meersman and Z. Tari et al. (Eds.): OTM 2007, Part II, Springer Verlag, LNCS 4804*, pp. 1264–1279, 2007.

[9] H:J. Mittag, H. Rinne: *Statistical Methods of Quality Assurance*. Chapman & Hall/CRC, 1993.

[10] F. Azzedin, M. Maheswaran: "Evolving and Managing Trust in Grid Computing Systems". In *Conference on Electrical and Computer Engineering, Canada, IEEE*, pp. 1424-1429, 2002.

[11] C. Lin, V. Varadharajan, Y. Wang, and V. Pruthi: "Enhancing Grid Security with Trust Management". In *Proceedings of the IEEE International Conference on Services Computing (SCC), Shanghai, China, IEEE*, pp. 303-310, 2004.

[12] J. Wu, D. Cheng, W. Zhao: "Detecting Grid-Abuse Attacks by Source-based Monitoring". In *Proceedings of the First International Workshop on Security in Parallel and Distributed Systems (IWSPDS), San Francisco, California, USA*, pp. 565-571, 2004.

[13] W.T.L. Teacy, J. Patel, N.R. Jennings, M. Luck: "TRAVOS Trust and Reputation in the Context of Inaccurate Information Sources". In *Autonomous Agents and Multi-Agent Systems 12 (2), Kluwer Academic Publishers* pp. 183-198, 2006.

Author Biographies

Elvis Papalilo received his B.Sc. degree in computer engineering from the Polytechnic University of Tirana, Albania, in 1998, and his Ph.D. degree in computer science from the Department of Mathematics and Computer Science of the University of Marburg, Germany, in 2008. His research interests include trust and security in distributed environments, Grid computing, policy languages and statistical methods of quality assurance.

Bernd Freisleben is a full professor of computer science in the Department of Mathematics and Computer Science at the University of Marburg, Germany. He received his M.Sc. degree in computer science from the Pennsylvania State University, USA, in 1981, and his Ph.D. degree in computer science from the Darmstadt University of Technology, Germany, in 1985. His research interests include distributed/parallel systems, cluster/network/Grid computing, and middleware for internet application development.