Journal of Computer Science 5 (9): 635-645, 2009
ISSN 1549-3636

Utility-Based Policy Management System for Virtual Organization

Anas J.A. Husain and Chan Huah Yong
School of Computer Sciences, University Sains Malaysia,
11800 USM, Pulau Penang, Malaysia

Abstract: Problem statement: Policy issues have become a challenge within Virtual Organizations (VOs) that integrate participants and resources, spanning multiple physical institutions. Each of the VO classes has different goals that hope to be achieved by providing policies. The main question in this regard is that "how such policies can help the entire VO participants achieve their goals". **Approach:** As a first step to address this question, we developed and evaluated a policy management framework within a specialized context, namely Utility based Policy Management System (U-PMS). We proposed an approach for policy federation, in which a group of VO participants agreed to adopt common standards to provide a common infrastructure that unifies the way the policies were applied in VO. **Results:** The evaluation results demonstrated that the proposed approach was able to manage VO policies and achieve higher utility for VO participant through its management functions. It outperforms the related system by 9.99% for acceptance function and 5.77% for conflict resolution and 4.65% for policy merging. **Conclusion:** We observed that the utility of VO participants' can be improved through managing the applied polices in VO.

Key words: Multi-attribute utility theory, virtual organization management, policy utility

INTRODUCTION

A grid computing infrastructure provides a mechanism to facilitate the integration of high performance resources across dynamic and geographically dispersed organizations into a transparent virtual high-performance computing environment. This virtual aggregation is called a Virtual Organization (VO). VO[1] is a virtual environment that integrates highly distributed resources with necessary communication to enable its members to combine and use these resources. Many of today's virtual organizations[2-6] are formed to tackle large-scale scientific problems. Since VO attracts more individuals and/or institutions over time, managing and controlling the behaviors and operations in this environment may become quite challenging. The uncontrolled resource access and usage creates an unfair sharing environment with the potential selfish behaviors of some VO users which violates fair resource usage.

The use of policies which govern the access to the resources and control activities and behaviors of the users was proposed through management systems because of these challenges[7]. The term policy is very common in management and has many meanings[8]. The policy in our environment can be defined as the rules and conditions that govern the behavior of the system[1], more specifically rules and conditions which are applied to the usage and in accessing the resources.

Participants collaborate in a VO in order to achieve their requirements. These requirements represent the desired goals of those participants from joining the VO. The main goal of VO is to provide service to its participants and this is represented by supporting them to achieve their requirements. The level of achievement of these requirements is called here as the utility for participants. Thus, achieving maximum utility for VO participants is the most important objective for VO.

The ultimate control of policy providers over their policies may result in achieving higher utility for individual participants rather than the entire VO participants. Policy providers are resource providers, resource consumers or VO representatives. Although resource providers may in principle agree to allow the VO access to their resources, each provider still retains ultimate control over the policies that govern access to its resources. Thereby, utilizing VO resources will be subjected to the preferences and desires of its provider. This may lead to inefficient resources sharing that deprive many participants from using these resources. Resource consumers may also place constraints on properties of the resources they are prepared to work

Corresponding Author: Anas J.A. Husain, School of Computer Sciences, University Sains Malaysia, 11800 USM, Pulau Penang, Malaysia

with. They may even have their own policies, specifying their preferences for actions that the system will take on their behalf. Thereby, the potential consumer's greediness may results in both increased and unexpected load on the resources of others in the VO and may also increase the time needed to complete the work. VO representatives may wish to apply some policies about how their users access the resources assigned to the VO. Since VO resources are located and controlled within multiple participant organizations, this may result in arbitrary policies that violate the autonomy of providers and the rights of their local users.

These conflicts between goals and requirements of different VO classes will become increasingly common and make it difficult to support overall VO participants to achieve their requirements. Therefore, we see that the utility for VO participants can be improved by controlling the way the policies are applied in VO environment through appropriate management functions that allow for policy utility evaluation. Policy utility represents its ability to support VO participants to achieve their requirements. The problem can therefore be summarized in the following questions:

- How the VO participants can achieve their goals through the applied policies?
- How the utility levels of VO participant's can be improved through policy management functions?
- How to evaluate the utility of policies?
- How the requirements and goals of VO participants can be used in the policy exploration and adopting decisions?

In this study, the major focus is on how to manage VO policies -including the way that policies are applied and the consistency among them-in a way that achieves higher utility for VO participants. Utility of policy represents its ability to support VO participants to achieve their requirements.

We proposed a new policy management system that supports VO participants to achieve their requirements through different management functions. The proposed system is considered an important VO service that can contribute to any applied policy management system. It introduces a new management task namely policy acceptance and it enhances the policy consistency management task. Since policy utility evaluation concerns with the different and multiple participants' requirements as evaluation criteria, we adopt the use of Multi Attributes Utility Theory (MAUT)[14] to solve the problem. In sum, this study aims to accomplish its main objective by achieving maximum utility for the VO participants.

Related work: Based on our knowledge, policy management solutions for VO which are related to a participant's utility do not exist yet. Most of the studies are instead focused on policy creation, specifications and enforcement issues. Some of the researchers focus on the policy conflict handling issues but their researches has weaknesses that will be discussed later and they did not consider the utility concept in their resolution techniques. Since we handled this problem in our proposed technique, we also review some of the researches on resolving conflicts among policies.

Firstly, conflict resolution approaches were aimed at assigning priorities to the policy rules, where the conflicts will be resolved by choosing the rules whose priorities satisfy some criterion. PCIMe (Policy Core Information Model extension)[9] relies solely on numeric priorities to resolve conflicts. In PCIMe a priority can be set for both rules and rule groups. A priority is about the relationship between the parent policy set and the subordinate policy group or rules (in PCIMe, rules and rule groups are objects that have hierarchy structures among themselves). The assignment of a unique numeric priority value becomes easier since the value is used only in relationship to other priorities in the same policy set. Another way to specify priority is by pair-wise comparison of rules. Precedence relationships are a natural way of expressing user-priorities because they increase rule autonomy and do not force the rule designer to know about all the rules in the system[10]. OPS5 and knowledge works[11] use a combination of some static properties (e.g., specificity) of the rules and some dynamic properties (e.g., recency of the instantiations of rules) to determine relative priority of rules. Specificity means that more special-case rules should override more general-case rules. Recency means that more recent rules are allowed to override less recent rules. Authority is a commonly used criterion, which makes rules from a more authoritative source to override a less authoritative one. Agrawal *et al*.[10] proposed a priority system for rules in database to achieve automatic deterministic behaviors. The mentioned priority system is incrementally maintained by combining user-defined priorities with default priorities. Default priority is based on some static properties of rules and is used to form a total order of the rules. User-defined priorities are specified by defining precedence relationships between rules. Conflicting rules are considered in the default order unless user-defined precedence constraints force an inversion.

However, there is a limitation in using only priority to resolve conflicts. There are cases that precedence cannot be established correctly, e.g., two rules in which a subject in one rule is more specific but an object is less specific or vice versa and as a result their precedence graph becomes cyclic. This kind of conflict is called “irresolvable”.

Secondly, Meta-policy is another approach that has been used widely in conflict resolution. Meta-policy is the policy about policies and is used to manage the interaction of rules in policy. Meta-policy can be used in conflict handling to specify how to reason about the priorities and application specific constraints such as constraints of actions and constraints of roles. Meta-policy can be specified either in the policy specification language or independently. In ASL (authorization specification language)[13], a logical authorization language, one can specify resolution rules to resolve conflicts and specify different constraints on authorization policy. However, by integrating conflict handling meta-policy in the policy specification, we increase complexity and decrease flexibility of the policy system. Jajodia *et al*.[12] proposed a language of meta-policy in access control. Strategies are used to define order of preference between rules and strategies themselves and can be used to reason about priorities. By associating strategy name with prioritization fact, the order of preference can be set on strategies to obtain more complex strategies and strategies can be recursively defined.

Expressive power always carries complexity with meta-policies which needs to be controlled in case the meta-policy itself requires some meta-control when it becomes too complex. Another problem with this approach is that it is very difficult to determine the proper relationship between a new rule and existing rules and set the appropriate meta-rules when the rule-base becomes large.

The common issue in all existing conflict resolution approaches is that they neglect the utility of VO participants. This may result in low levels of utility for participants which have conflicts with the main goal of the VO. In our study, we propose a way to resolve the conflicts based on participants themselves rather than some properties of polices rules. In our method, participants are able to define their requirements and the system will resolve the conflict based on the policy that best satisfies these requirements. This ultimately achieves higher utility and satisfaction for VO participants.

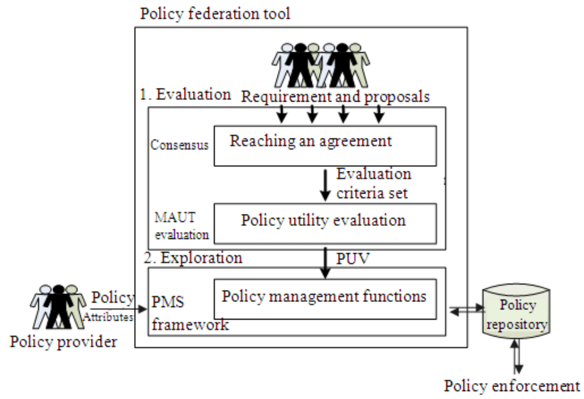

Fig. 1: Utility-PMS overview

System design and requirements: The proposed system is able to achieve higher utility for participants through managing VO polices. It responsible for two management functions; policy acceptance function and policy conflict handling function. According to our policy acceptance strategy, before adopting any policy in the VO environment, its utility for the VO participants should be examined through an evaluation process called policy utility evaluation. Policy utility evaluation results in a numerical value called Policy Utility Value (PUV). PUV presents a reference for all policy management selection decisions. Policy management selection decisions include policy adopting decisions for new policies and selecting the best policy among a set of conflicting policies. According to our conflict resolution strategy, the utility of conflicting policies will be evaluated and the policy with higher PUV will be selected. However, when multiple participants collaborate in VO with different requirements, an agreement needs to be reached between those participants upon set of common requirements with common evaluation standards to be a reference for policy utility evaluation. This supports the entire VO participants for their requirements to be considered in policy management selection decisions. In Fig. 1, an overall system is provided to achieve our research objectives.

METHOD

To achieve the research objectives, we have designed a policy management system responsible for achieving higher utility for VO participants through managing VO policies according to their utility. We designed and developed a PMS with policy acceptance and conflict handling functions to measure its ability to achieve higher utility for VO participants. The evaluation was performed through an experimental test which is an execution of the simulation system several times with different simulation data inputs according to selected system parameters. The result was compared with the related existing systems that have the same function to investigate the efficiency of the proposed system.

System detailed design: The proposed system is composed of two parts; policy evaluation and policy exploration. Policy evaluation is responsible for aggregating participant's requirements through an agreement process to allow creating a consistent set of evaluation criteria that can be used as references for policy utility evaluation. Policy exploration is responsible for managing policies by performing the system management functions and policy selection decisions. These decisions will be considered based on the obtained policy utility value from policy utility evaluation.

Policy evaluation: The purpose of a policy evaluation process is to provide a mechanism that determines the policy ability to support VO participants in achieving their requirements, which called here as policy utility. Policy evaluation requirements in a VO environment are a consistent set of evaluation criteria and an evaluation function.

Evaluation criteria: The policy evaluation criteria consist of the attributes that reflect the areas of importance to the participants in its policy selection decisions. Each evaluation criterion consists of one or more attributes that are common in one goal. For example, the attributes can be the amount of access time that is assigned to a set of users on a specific resource and the number of those allowed in accessing that resource. These attributes affect the ideal level of resource workload which is considered as one of the important requirements for VO participants. Through the evaluation criteria, the system is able to assess the strengths and weaknesses of the policy utility. It can use that assessment in making a related policy selection decisions.

The use of participants' requirements is the key of our proposed solution. It is important that the policy evaluation criteria clearly reflect the participants' requirements in order to assess its ability to achieve these requirements. But due to the complexity of participants and their conflicting requirements, the process of policy evaluation becomes challenging and requires a consistent set of evaluation criteria that would combine the participant's opinions to form a reference for policy evaluation.

We propose to use evaluation standards to uniform the application of evaluation criteria. Standards establish the ideal level of acceptability for a requirement or criterion and provide the basis on which the ratings above and below the minimum levels are set. It is a measurement baseline that will be used by the system to determine whether a policy meets, exceeds, or fails to meet a participant's requirements. Each attribute in the evaluation criteria must have an evaluation standard. Therefore, evaluation standard values are the values that the VO participants need to agree upon in order to form a consistent set of evaluation criteria, which we refer to it as a common criteria set.

Through evaluation standards, all VO participants will be able to agree on a set of evaluation criteria without depriving anyone of the responsibility of making decisions. At the same time it is guaranteed to produce a coordinated and consistent set of criteria. These criteria accurately represent the participants' requirements and leads to an effective policy selection decisions.

Evaluation function: According to the multi-attribute utility theory, the overall evaluation of an object is defined as a weighted addition of its evaluation with respect to its relevant value dimensions. The common denominator of all these dimensions is the utility for the evaluator. Therefore, the policy evaluation function v(x) is defined as a weighted addition of its evaluation with respect to a common criteria set. The overall Policy Utility Value (PUV) is defined by the following function:

$$v(x) = \sum_{i=1}^{n} w_i v_i(x)$$

Here, v_i (x) is the evaluation of the policy on the i-th common criterion, where it represents the ability of the policy to accomplish this criterion. The w_i the weight determining the impact of the i-th criterion on the overall evaluation (also called the relative importance of a dimension), n is the number of different common criteria and:

$$\sum\nolimits_{i=1}^{n} w_i = 1$$

For each common criterion, the evaluation $v_i(x)$ is defined as the evaluation of the relevant attributes:

$$v_i(x) = \sum_{a \in A_i}^{n} w_{ai} v_{ai}(l(a))$$

Here:

A_i = The set of all attributes relevant to criterion $_i$,
v_{ai} (l(a)) = The evaluation of the actual level l(a) of attribute a on criterion $_i$
w_{ai} = The weight determining the impact of the evaluation of attribute a on common criterion $_i$. w_{ai} Also called the relative importance of attribute a for criterion $_i$

For all i-th criteria (i=1,...,n) holds $\sum_{i=1}^{n} w_{ai} = 1$.

Finally, for each criterion attribute a, the evaluation l(a) can be defined according to the criterion of this attribute. The common parameters for this evaluation are: the current achieved value of that attribute at evaluation moment and the evaluation standard value for that attribute. Different criteria require different evaluation function e.g.:

$$l(a) = \frac{a}{\text{Evaluation standard}}$$

In the following discussion we define some criteria and we provide their evaluation functions. The policy evaluation process with respect to each criterion attribute represent the policy ability to support participants to achieve this attribute and can be accomplished through two steps: The first step is to evaluate the current system (level of achievement of that attribute) with respect to the underlying attribute without considering the policy values; the second step is to evaluate the system with the new policy values. The difference between these two evaluations will be the impact of the policy on the participants and can represent its ability to achieve that attribute. If the assessment after applying the new policy is increased, then the policy achieves higher utility for the participant.

Agreement: In order to consider the entire VO participants requirements in policy utility evaluation without depriving any one, they need to agree upon evaluation criteria set that are optimized to satisfy their requirements to be supported through VO polices. The problem is for VO participants to agree upon an evaluation standard value for each common criterion attribute after each participant has proposed what the value should be. To reach a consensus, VO participants are able to communicate through a centralized message passing system with a predetermined coordinator, every participant proposes a single value of evaluation standard related to each criterion attributes in the agreement problem and communicates with the predetermined coordinator passing the values. The coordinator waits until it has collected all N values (including its own). It then evaluates the function majority (v_1, v_2,…,v_N), which returns the value that occurs most often among its arguments. The final value will be used as the evaluation standard for its related common criterion. The agreement process is not in the scope of this study, so we won’t discuss it in details.

Policy exploration: Policy exploration is responsible for managing VO policies according their PUV’s and achieving higher utility for participants. Policy exploration management functions are: (1) Policy acceptance function, (2) Policy conflict resolution function, (3) Policy merging functions. These functions require different decisions and will be based on the predetermined PUV.

Figure 2 shows the policy flow process in the policy exploration. After the PUV is calculated, the policy acceptance decision will be taken based on its utility value. For the adopted policies, the next step is exploration from any overlapping between the new policies and existing policies. If there is an overlapping, policy exploration determines the case of that overlapping. If the overlapping policies have similar actions, the utility value will be calculated for each possible merging combination and the one with the higher utility value is selected. If the overlapping policies have contradicting actions, the utility value will be calculated for each overlapping policy and the policy with the higher utility value will be selected. The selected policy will be saved into a system repository.

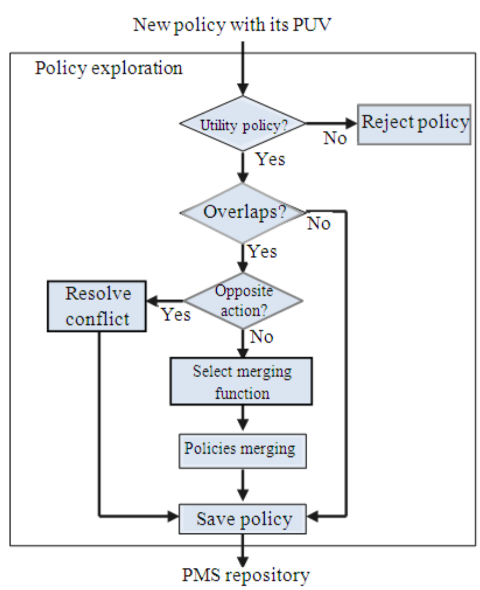

Fig. 2: Flow of policy exploration process

Policy acceptance: According to our proposed framework, we introduce a new management function for managing VO policies which can be used for any policy management purpose in the collaborative environments. Policy acceptance strategy provides an efficient technique to achieve a higher utility for VO participants through the adopted policies in VO. Policy acceptance is responsible for examining the utility of any policy and its ability to support VO participants to achieve their requirements. It accepts only the beneficial policies based on their PUV. This procedure guarantees that the utility levels of participants will be improved or at least maintained. Since the common criteria clearly reflect opinions of all participants, the accepted policies (which are based on the common criteria) are also implicitly accepted by all participants.

Policy conflict handling: Another way to achieve better utility for VO participants is through handling conflicts of VO policies. Since policy overlapping and conflicts cannot be prevented in VO environment due to the diversity of the policy providers and policy enforcement issues of these conflicts, conflict handling becomes a very important requirement in this environment. The policy conflict handling function provides an efficient resolution technique based on the policy utility concept. Policy conflict handling includes conflict resolution function and policy merging function.

Conflicts among VO policies occur when the new accepted policy overlaps any existing policies and their actions are contradictory. In this case the system needs to decide which policy to choose and which policy must be ignored. Based on our solution, the policy that achieves better utility for VO participants must be selected. The other policies will be deleted.

Policy merging function required when there is an overlapping between the new accepted policy and one or more of the existing policies where their subjects are both authorized or forbidden to do the same action for the same target under overlapping constraints. These overlapping objects must be merged to form one new policy. Because there are several possibilities for policy merging, corresponding to various combinations of overlapping between objects sets, we need to select the appropriate function for this merging. For example if we have two policies overlapping in time constraints with the same subject and action and target objects, we need to decide whether to apply a union or intersect function to these overlapping sets. We call the selection decision of appropriate policy merging function as a policy merging decision and it will be based on PUV. For each policy formed from each possible overlapping combination, in addition to the original overlapping policies, the system will select the policy that has a higher PUV to be saved in the system repository. The rest of the policies will be deleted.

Evaluation and performance metrics: In order to evaluate our proposed system and measure its ability to achieve the research objectives, we have developed a system that simulates the VO environment and its PMS requirements. In this discussion our simulation system requirements and design details are presented. Finally, we present the performance metrics which enable us to measure the system performance.

Simulation system design: We have developed a simulation system which provides an appropriate VO environment and allows participants to specify their different polices. Through these policies we measure the system's ability to achieve better utility for VO participants through its different policy management functions. Using Pondor terminology[15], the policy structure that we used in our simulation system design is similar to the IETF model with some differences; the policy structure is as follows: participants are able to specify (1) A policy subject which can access and use a policy resource e.g., a participant's name, (2) Policy resources target that policy subjects can access and use, (3) Time constraints that define when the policy subject can access and use a policy resource. Policy time constraints can be designed in different ways according to the nature of the target environment. We designed policy time constraints to be defined by the start time and end time in terms of hours. The time period we consider is the day hours (24 h). Thus, the provided policies represent constraints for every day during its validity date. However, we use a database to simulate the PMS repository. The simulation system is efficient enough to provide support for a varied number of policies, participants, users and evaluation criteria. Therefore, it enables us to do different experiments in order to evaluate the system performance efficiently and ultimately assess the utilization levels preserved by participants.

Accordingly, we developed our simulation system using java programming language (JDK) with Net Beans IDE within windows XP operating system environment. We developed the policies database using MYSQL which connected to the main program. We ran the simulation system many times based on the predetermined scenarios. For each scenario we ran the simulation one time on one machine and each time the simulation configured with different number of participants and users for each participants, different numbers of polices policies were specified.

Simulation requirements: A set of participants' requirements are needed to be used in our simulation system and the ability of the system to achieve these requirements should be measured. We proposed a set of requirements that we thought are the most important requirements for most of the VO participants. We assume that these requirements are common for all participants and they need to agree upon an evaluation standard for each requirement. The relative importance of these requirements is almost the same with a little deference. Later we will present relative importance values for each requirement. Our method is able to adopt any requirement that can be proposed by any participant. The requirements are described below:

Level of resource sharing (L.Sharing): the amount of time by which the providers make their resource available for execution to the VO participant users upon request.

Resource workload: Maintaining the workload of VO resources around their ability and avoiding exceeding the execution time that may be caused by the excessive quantity of users. Workload here refers to the access and usage time by others.

Level of access (L.Access): the amount of time that a specific VO participant can access and use different VO resources upon demand.

Access quality (Access Q): Percentage of the access quality according to quality of target resources.

Fairness: Fairness among participants means that each participant gains equity of time in which it can access and use VO resources in relation to the other participants in VO.

Balance of Resources workload: means that each resource will be assigned an equal of participant access time (workload) in relation to the other resources in VO. Thus the VO resources are allocated to VO participants evenly.

These requirements are set to be common criteria through participants' agreement. As discussed earlier, policy evaluation results in PUV which can be obtained by calculating the achieved levels of common criteria before considering the policy and after considering the policy. The difference will be the PUV which may be positive or negative. The achieved levels of the above common criteria can be calculated through the following equations respectively:

$$\mathrm{L.Sharing}(R_i) = \frac{\mathrm{SharingTime}(R_i)}{\mathrm{Ideal\,SharingTime}(R_i)} * 100 \quad (1)$$

$$\mathrm{Workload}(R_i) = \frac{\mathrm{SharingT}(r_i)}{\mathrm{Mr}} * 100 \quad (2)$$

If:

$$\mathrm{Sharing\,T}(R_i) \geq \mathrm{idealsharingT}(R_i)\mathrm{else\,Workload}(R_i) = 100$$

$$\mathrm{L.Access}(pa_i) = \frac{\mathrm{AccessT}(pa_i)}{\mathrm{Ideal\,AccessT}(pa_i)} * 100 \quad (3)$$

$$\mathrm{AccessQ}(pa_i) = \frac{\mathrm{QAccessT}(pa_i)}{\mathrm{Ideal\,QAccessT}(pa_i)} * 100 \quad (4)$$

SharingTime(R_i) is the time period assigned to resource R_i and the specified subjects can access and execute their work at that time period. IdealSharing (R_i) is the evaluation standard for the current criterion which represents the amount of time that the resource can serve different users concurrently with reasonable efficiency and performance (level of acceptability) from participants' viewpoints. The Time used in all equations is constant, since we used day time unit in our policy design, the time will be equal to 24 h. For example, a participant's access time being 74 h means this participant's users (many users) have 74 h in total in which they can access and use the specified resources per day. M_r represents the maximum period of time that VO participants can be assigned to a specific resource (or this resource can be shared among them) at a point of time, where M_r = Users number of VO participants *Time. The AccessTime(pa_i) is the total amount of time that the participant can use and access the specified VO resources. The Ideal Access Time (pa_i) is the evaluation standard for the current criterion which represents the amount of time that a participant (pa_i) must be allowed (level of acceptability) to access and use resources from the participants' viewpoints. The QAccess Time(pa_i) is the total amount of time that the participant is allowed to access and use high quality resources. The idealQAccess Time(pa_i) is the evaluation standard for the current criterion which represents the amount of time that a participant (pa_i) must be supported (level of acceptability) to access and use high quality resources from the participants' viewpoints.

Fairness is calculated based on the participants' access time in relation to each other. In order to deal with different amounts of participant's access time that may highly vary, we normalize their access time to adjust its range to a value between 0 and 100 as a percentage by the equation:

$$\text{Access}(pa_i) = \frac{\text{AccessTime}(pa_i)}{\text{Mpa}_i} * 100$$

$\text{Access}(pa_i)$ = The normalized participant access time

M_{pai} = The maximum period of time that can be assigned to a participant i to access and use VO resources at a point of time, where M_{pai} = Number of i-th participant users*Number of VO resources *Time

With the normalized access time for each participant, the fairness is represented by the total deviation of each participant's access time from the average of all participants' access time periods. Again, the total deviation may highly vary based on the number of participants and the varying access time periods. Thus, we normalize the fairness of participants and calculate its percentage by the following equations:

$$\text{Fairness} = \left[1 - \frac{\sum_{i=1}^{m} \left|\text{Access}(pa_i) - \text{AvgaccessTime}\right|}{MD_{pa}}\right] * 100 \quad (5)$$

Where:

m = Represents the number of VO participants

AvgAccessTime = Represents the average of access time periods for all participants and calculated by:

$$\text{AvgAccessTime} = \frac{\sum_{i=1}^{m} \text{Access}(pa_i)}{m}$$

MD_{pa} represent the maximum deviation degree that can exist among participants' access time periods at a point of time, we use MD_{pa} to normalize the value of deviation since it may vary, it adjusts the fairness to a value between 0 and 100%. To calculate MD_{pa}, we found that the maximum deviation among participant's access time periods occur only when the average is 50. Therefore, the maximum deviation between these periods can be calculated through average equation:

$$50 = \sum_{i=1}^{m} \text{Access}(pa_i) / m$$

Since:

$$\sum_{i=1}^{m} \text{Access}(pa_i)$$

Represents the total deviation of n participants, the max deviation of n participants becomes MD_{pa} = n*50.

Balance of resources workload is calculated based on resources sharing time in relation to each other. In order to deal with the different amounts of resources sharing time that may highly vary, we normalize it to adjust its range to a value between 0 and 100%:

$$\text{Sharing}(R_i) = \frac{\text{SharingTime}(pa_i)}{\text{Mr}} * 100$$

$\text{Sharing}(R_i)$ = The normalized resource sharing time

M_i = The maximum period of time that the VO participants can be assigned to a specific resource i(or this resource can be shared among them) at a point of time, where M_i = number of all VO participant Users* Time

With the normalized sharing time for each resource, the balance is represented by the total deviation of each resources sharing time from the average of resources sharing time periods. Again, the total deviation may highly vary based on the number of resources and the varying of sharing time periods. Thus, we normalized the balance of resources workload and calculate its percentage by the following equations:

$$\text{Balance} = \left[1 - \frac{\sum_{i=1}^{n} \left|\text{Sharing}(R_i) - \text{AvgSharingTime}\right|}{MD_r}\right] * 100 \quad (6)$$

Where:

n = Represents the number of VO resources

AvgSharingTime = Represents the average of sharing time periods for all VO resources and is calculated by:

$$\text{AvgSharingTime} = \frac{\sum_{i=1}^{n} \text{Sharing}(R_i)}{n}$$

MD_r represents the maximum deviation degree that can exist among resources sharing time periods at a point of time, we use MD_r to normalize the value of the deviation since it may vary, it adjusts the balance to a value between 0 and 100%. To calculate MD_r, we find that the maximum deviation among resources sharing time periods occurs only when the average is 50.

Therefore, the maximum deviation between these periods can be calculated through average equation:

$$50 = \sum_{i=1}^{n} \text{Sharing}(R_i) / n$$

Since:

$$\sum_{i=1}^{n} \text{Sharing}(R_i)$$

Represents the total deviation of n resources, the max deviation of n participants become $MD_r = n*50$.

Utility level metric: We proposed the level of utility that is preserved by VO participants as a metric to be used for system performance evaluation. The system outputs need to be measured, analyzed and compared with other models. This metric represents the system ability to support the VO participants achieving their requirements which was discussed earlier.

A utility level of participants has a value ranging from zero to 100%. Achieving higher value means the system is better. The overall utilization level can be calculated through the following steps:

Step 1: With respect to each criterion, we calculate the achieved levels for each participant through the Eq. 1-6.

Step 2: With respect to each criterion, we use the above calculation and calculate the achieved levels for all participants through the following equations:

$$\text{WORKLOAD} = \left[1 - \frac{\sum_{i=1}^{n} \text{overload}(R_i)}{n}\right] * 100 \qquad (7)$$

$$\text{SHARING} = \left[1 - \frac{\sum_{i=1}^{n} \text{overload}(R_i)}{n}\right] * 100 \qquad (8)$$

$$\text{ACCESS} = \frac{\sum_{i=1}^{m} \text{levelofAccess}(pa_i)}{m} \qquad (9)$$

$$\text{QUALITY} = \frac{\sum_{i=1}^{m} \text{AccessQuality}(pa_i)}{m} \qquad (10)$$

The fairness and balance can be calculated through the Eq. 5 and 6.

Step 3: The utility level for all participants can be calculated based on the Eq. 5-10. In this study, the proposed criteria are fairly competitive but there is little difference in their importance. We think that access quality requirement is part of level of access requirement. Therefore, we proposed to give it less importance than the other requirements where it can be seen as an advantage rather than a requirement. However, to maintain the balance among the importance we assume the weights of our criteria (Level of sharing, level of access, resource workload, fairness, balance of workload and quality of access) as (0.2, 0.2, 0.2, 0.15, 0.15, 0.1) respectively. The summation of the weight equals to 1. Thus, according to equation 11, the utility can be calculated by:

$$\begin{aligned}\text{UTILTY} = {} & 0.2(\text{SHARING+ACCESS+WORKLOAD})+ \\ & 0.15(\text{FAIRNESS+BALANCE})+0.1\text{QUALITY}\end{aligned} \qquad (11)$$

RESULTS AND DISCUSTION

For evaluation purposes, we ran our simulation system in different scenarios with respect to the proposed PMS functions. The results are compared with the related systems results. In an evaluation study, ability of the system in achieving higher utility for VO participants through its different management functions is measured. According to the system management functions, the tests are divided into three test cases, namely policy acceptance, conflict resolution, policy merging. In order to expose the throughput and system performance, we run each test case in different scenarios and compare with related systems that have the same functions. The main parameter that substantially influences the participant's utilization level is the number of policies that are applied in VO.

However, for testing purposes we select the number of rules based on the potential sizes of VO as shown in Table 1. In our tests, we considered only large VOs. This is because in small VOs the problem of a participant's utility is not significant due to the small number of policies.

We have got three scenarios for each system’s function, a combination of the three mentioned number of policies parameters as shown in Table 2. The achievement levels of participants requirements mentioned previously are computed for all participants based on Eq. 5-10. In order to calculate the overall utility preserved by VO participants, Eq. 11 is used.

Table1: summary of potential VO’s sizes

No.	VO Size	Potential users in VO	No. of potential rules
1	V. small	25 user or less	Few Hundreds of rules
2	Small	26 to 100 user	Hundreds of rules
3	medium	100’s of users	Thousands of rules (small)
4	Large	1000’s of users	Tens of thousands of rules (medium)
5	V. large	10,000’s of users	Hundreds of thousands of rules (large)

Table2: simulation scenarios for system evaluation

System function		Scenarios	No. of policies
a	Policy acceptance	a1	Small
		a2	Medium
		a3	Large
b	Conflict resolution	r1	Small
		r2	Medium
		r3	Large
c	Policy merging	m1	Small
		m2	Medium
		m3	Large

In the policy acceptance function test case, our system is compared with other similar existing systems. Since no previous study has used this function, we compared our policy acceptance function with the standard system. The standard system accepts all policies without examining the utility of the new policies. The simulation results of our system and the standard system for all scenarios shows that our system performs the best in comparison to the combinations of the six criteria that represented the utility for VO participants. It is also shown that our policy acceptance function achieves 78.16% utility for VO participants while the standard system achieves 68.17% utility only.

In the policy conflict resolution test case, our system is compared with other similar existing systems that handle conflict resolution function, namely, policy conflict resolution based on recency system and conflict resolution based authority system. Recency means that more recent rules are allowed to override less recent rules. Authority makes rules from a more authoritative source, which overrides a less authoritative one. The simulation results of our system and the related systems for all scenarios shows that our system performs the best in comparison to the combinations of the six criteria that represent the utility for VO participants. The result shows that our system achieves 78.16% utility for VO participants while the recency of rules achieves 70.86% utility and authority of rules achieves 73.89% utility only.

In the policy merging test case, our system is compared with other similar existing systems that had been commonly used for policy merging functions, namely a system without policy merging, policy merging based on intersection of overlapping objects system and policy merging based on the union of overlapping objects system. The simulation results of our system and the related system for all scenarios are summarized and represented in Table 3. As a result, our system performs the best in comparison to the combinations of the six criteria that represent the utility for VO participants. The result shows that our system achieves 76.75% utility for VO participants while the no merging system achieves 75.2% utility, intersection system achieves 70.34% utility and union system achieves 70.73% only. It should be mentioned that a system which is based on no-merging, apart from inefficiency in utilizing participants, creates several problems in relation to policy enforcement and selection process when handling a user's requests.

Table 3: Simulation result for policy merging test case

Sr. No.	Metrics	Utility	No merging	Union	Intersect
C1	L.SHARING	75.70	78.14	81.40	60.41
	WORKLOAD	100.0	97.02	93.05	100.00
	L.ACCESS	71.02	79.42	85.82	59.82
	QUALITY	22.29	19.38	25.28	23.29
	BALANCE	82.26	79.57	75.99	85.84
	FAIRNESS	93.95	83.87	70.43	92.61
	Utility	78.00	77.37	76.54	73.14
C2	L.SHARING	84.00	84.00	84.00	84.00
	WORKLOAD	63.13	70.97	47.45	84.69
	L.ACCESS	96.40	84.47	100.00	52.99
	QUALITY	25.93	23.73	30.29	17.28
	BALANCE	71.27	76.29	61.24	85.07
	FAIRNESS	98.62	87.58	71.02	63.43
	Utility	76.78	74.84	69.16	68.34
C3	L.SHARING	84.00	84.00	84.00	84.00
	WORKLOAD	54.05	51.25	40.05	65.25
	L.ACCESS	98.65	99.45	99.45	76.25
	QUALITY	30.30	27.93	38.29	23.29
	BALANCE	69.31	67.52	60.35	76.48
	FAIRNESS	98.07	90.34	59.43	71.02
	Utility	75.48	73.41	66.50	69.55
Overall utility		76.75	75.21	70.73	70.35

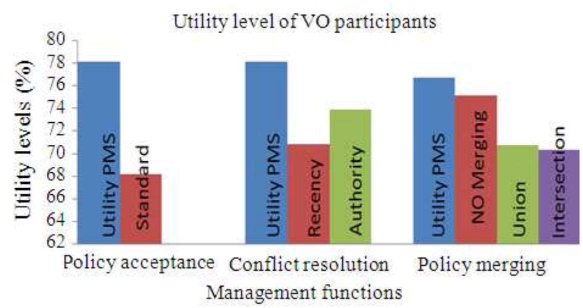

Fig. 3: Achieved utility levels through management functions

In summary, the overall utility for VO participants is verified in Fig. 3. The best system is the one that achieves more utility for VO participants. Therefore, our system performs the best among the other existing systems in regards to all of the functions for all the scenarios.

CONCLUSION

We have discussed the policy problem in the VO environment and how it affects the participant's utilization level. We have provided an efficient framework that is able to achieve better utility level for VO participants through performing two management functions namely, policy acceptance and conflict handling. These functions are based on the utility values for VO participants. We have developed a simulation system and evaluated our approach with respect to the participant's utilization level. The results showed that our system performs better than the related systems for all the scenarios and all the functions. There are still problems not fully explored in this study. Firstly, we have not studied the agreement function that the participants can use in order to agree upon a set of evaluation criteria standards. Secondly, we did not clearly address how the participants can communicate and exchange proposed values related to the evaluation standards.

REFERENCES

1. Foster, I., C. Kesselman and S. Tuecke, 2001. The anatomy of the grid: Enabling scalable virtual organizations. Int. J. High Perform. Comput. Appli., 15: 200-222. http://portal.acm.org/citation.cfm?id=1080644.1080667
2. National Computational Science Alliance (NCSA) project. http://www.ncsa.uiuc.edu
3. The USATLAS Grid project, 2002. http://www.usatlas.bnl.gov
4. GriPhyN, 2002. The grid physics network. http://www.einet.net/review/14892-862541/GriPhyN_Grid_Physics_Network.htm
5. The information power grid project. NASA, 2002. http://www.cfdreview.com/research/02/06/24/230243.shtml
6. NPACI, 2004. National partnership for advanced computational infrastructure. http://npacigrid.npaci.edu/
7. Strassner, J., 2003. Policy-Based Network Management: Solutions for the Next Generation. 1st Edn., The Morgan Kaufmann Series in Networking, ISBN: 10: 1558608591, pp: 516.
8. D. Verma, 2001. Policy Based Networking: Architecture and Algorithms. New Riders Publishing, California, ISBN: 1578702267 pp: 187.
9. Moore, B., L. Rafalow, Y. Ramberg, Y. Snir, A. Westerinen, R. Chadha, M. Brunner R. Cohen and J. Strassner, 2003. Policy core information model extensions. http://www.priorartdatabase.com/IPCOM/000011355/
10. Agrawal, R., R.J. Cochrane and B. Lindsay, 1991. On maintaining priorities in a production rule system. Proceeding of 17th International Conference on Very Large Data Bases, Sept. 3-6, Morgan Kaufmann Publishers Inc., San Francisco, CA., USA., pp: 479-487. http://portal.acm.org/citation.cfm?id=672166
11. XANALYS, 2001. http://www.xanalys.com/software_tools/products/knowledgeworks.html
12. Jajodia, S., P. Samarati and V.S. Subrahamanian, 1997. A logical language for expressing authorizations. Proceeding of the IEEE Symposium on Research in Security and Privacy, May 4-7, IEEE Xplore Press, Oakland, CA., USA., pp: 31-42. DOI: 10.1109/SECPRI.1997.601312
13. Jajodia, S., P. Samarati, V.S. Subrahamanian and E. Bertino, 1997. A unified framework for enforcing multiple access control policies. Proceeding of the ACM SIGMOD International Conference on Management of Data, (ICMD' 97), ACM Press, pp: 474-485. http://www.citeulike.org/user/weqa/article/1453236
14. Winterfeld D. Von and W. Edwards, 1986. Decision Analysis and Behavioral Research. Cambridge University Press, Cambridge, England, ISBN: 052125308X, pp: 604.
15. Damianou, N., N. Dulay, E. Lupu and M. Sloman, 2001. The ponder policy specification language. Proceedings of the International Workshop on Policies for Distributed Systems and Networks, July 29-31, Springer-Verlag, London, pp: 18-38. http://portal.acm.org/citation.cfm?id=646962.712108