International Journal of Production Technology and Management (IJPTM)
Volume 10, Issue 2, July- December 2019, pp.101-109, Article ID: IJPTM_10_02_015
Available online at http://iaeme.com/Home/issue/IJPTM?Volume=10&Issue=2
ISSN Print: 0976- 6383 and ISSN Online: 0976 – 6391
DOI: https://doi.org/10.17605/OSF.IO/FVJPM
© **IAEME** Publication

INSPECTING THE POLICY CONFLICTS IN DISTRIBUTED SYSTEM MANAGEMENT

Dr. K. Venkatasalam
Associate Professor, CSE Department, Mahendra Engineering College,
Namakkal, Tamil Nadu, India

M. Pandiyan
Assistant Professor, CSE Department, Mahendra Engineerin
g College,
Namakkal, Tamil Nadu, India

ABSTRACT

The activities required to ensure that broad distributed networks will operate in compliance with their users' goals are referred to as distributed system management. This goals are usually stated in the form of policies, which are then interpreted by system administrators. There are advantages of offering automatic assistance to human administrators or automating repetitive management functions. It is desirable to provide a model of policies as artefacts that can be represented by the framework itself in order to accomplish this. This is a summary of the model. There is no doubt that policy conflicts will arise. Human administrators may be able to handle these disputes informally, so in order for an automated framework to recognise and resolve them properly, it must first analyse the forms of dispute that may arise. We examine the different forms of policy overlap that may exist to explain how this study relates to the different types of policy dispute. This study is placed in the light of other work on policy, authority and similar fields, including deontic reasoning, and several alternative approaches to dispute prevention and resolution are proposed.

Key words: Management policy, policy conflicts, authority, conflict resolution, distributed system management.

Cite this Article: K. Venkatasalam and M. Pandiyan, Inspecting the Policy Conflicts in Distributed System Management, *International Journal of Production Technology and Management (IJPTM)*, 10(2), 2019, pp. 101–109.
http://iaeme.com/Home/issue/IJPTM?Volume=10&Issue=2

1. INTRODUCTION

This paper introduces a policy model for distributed system management and examines several potential forms of policy disagreement in terms of the model's components. For organisations to coordinate their own operations and communicate with others, large distributed computing networks are becoming increasingly necessary. They are usually made up of several integrated networks that cover the operating infrastructure of many separate companies. For a variety of reasons, active management rather than reactive management is needed for programmes of this kind. For starters, multiple organisations already rely on distributed networks to operate, and they've progressed from supporting to organisational positions, necessitating a proactive approach to ensuring that they work properly. Second, in many situations, the machine is not managed from a single stage, but often requires the collaboration of many different administrators to keep it running. Third, distributed networks are extremely dynamic in many ways: they may comprise hundreds of thousands of resources and be used by thousands of users; they may be distributed across large geographical areas, across international borders, across different regulatory authorities, and across time zones; they may contain various materials manufactured by diverse manufacturers; they may be distributed across large geographical areas, across international boundaries, across different regulatory authorities, and across time zones; they may contain hundreds of thousands of resources and be used by thousands of users.[1]

1.1 Distributed System Management

The role of managing the operation demanded of a framework is referred to as distributed system management. A series of Open Systems Interconnection (OSI) management specifications has been created to make this feasible for heterogeneous systems and to deal with uncertainty. They divide overall administration into functional areas of accountability. The main functional areas identified by OSI are: Configuration Management, which is concerned with controlling the installation of both hardware and software components within a distributed system or application; Performance Management, which is concerned with optimisation of performance to improve the service provided to users in terms of better throughput, response times, or reliability, or to reduce operating costs; and Fault Management, which is concerned with fault management to improve the service provided to users in terms of better throughput, response times, or reliability, or to reduce operating costs. In addition, while it is not a typical OSI Systems Management feature, monitoring of state, failures, output, and utilisation information is required to help any of the above management functions. The solution to distributed systems management, like any other management job, is to behave as far as possible based on general policies rather than specific situations. This entails the development of policies that refer to classes of device components and consumers rather than specific units in abstractly specified circumstances. The same regulation, for example, may extend to anyone in a department or to the whole collection of files related to an application. The classification of machine artefacts into management domains, which policies can correspond[2].

2. POLICIES

2.1 Management Policies

All structured organisations have rules, which are described by the dictionary as an organization's plans to achieve its objectives. They are the engine that drives management. They have two purposes: defining the organization's priorities and allocating capital to accomplish those goals. In a bureaucratic manner, the policies are used as a management tool. A high-level policy directs a boss, who may accomplish his or her objectives by enacting lower-level policies that affect other managers in the hierarchy. Most organisations provide Policy Statements that provide guidance to its representatives in specific situations. Policies can offer constructive feedback on the organization's objectives and how they should be accomplished, or they can impose limits about how the goals should be achieved. Such policy statements delegate (authorise access to) the tools required to achieve the objectives. Budgets are what they're called as they spend resources. The need for autonomous managers to be able to negotiate, create, question, and execute policies that relate to a given general collection of circumstances is a popular theme in distributed system management[3].

The interconnection between two network administration areas, such as a Public Network (PN) and a local Imperial College (IC) network, is an example of cooperation between autonomous managers. In order to share management details and determine access authority, the PN and IC network managers must communicate. Assume that there are two pertinent policies in effect: PN policy grants the PN Manager complete control of all network management processes, while IC policy allows the IC Network Manager to update his users on the state of the academic subset of PN nodes on a daily basis. These administrators are referred to as policy topics. In the absence of all such policies, the PN Manager has the right but not the responsibility to supply routine status reports, while the IC Network Manager has the obligation but not the authority to collect the information. To satisfy IC's criteria, the PN Manager must establish (create) an additional regulation. As seen in figure 1b, one solution is to develop an imperatival strategy that requires the PN Manager to produce status details and provide it to the IC Network Manager on a regular basis. As seen in figure 1c, an alternate solution is to establish a policy that grants the IC Network Manager the authority to conduct the operations required to receive the normal status details. This example highlights one of the model's key points. Policies that initiate practises and policies that grant authority to carry out activities will also occur independently of one another. However, if only either of the two types of policies applies to a certain procedure, it would not be carried out. For management operations to be carried out, a boss must be the focus of two types of policies: one that gives him permission to carry out the activity and another that forces him to do so[4].

2.2 The Need to Model Policies

We define system management as the process of putting the policies of the organization(s) in charge of the system into action. Independent managers need a system that allows them to question, discuss, set up, and alter policies. Of course, the tried-and-true way of making phone calls and exchanging document may be used, but there are possible advantages of utilising the distributed mechanism itself to connect and store policy, particularly in terms of automated management. As a result, regulations must be able to be represented and modified inside a computing framework. It's critical that the way policies are represented and the procedures used to discuss them are consistent across management applications. The possible advantage of computer-assisted support for distributed system management, or even full automation in suitable situations, is a significant element in distributed system management. With the automation of many areas of management of distributed systems and information networks, it's become necessary to reflect management strategy within the computer system so that automated administrators can understand it and control their behaviour. Since 'regulation' is such a broad concept, there is no chance of capturing both types of policy in a model. We discern between two types of policy here, though acknowledging that there might be several others that are none[5].

2.3 Policy Conflicts

We'll use the dictionary meaning of dispute as a starting point: "opposition, disparity, disagreement." Policy disputes are defined by a number of well-known words. Conflict of interests refers to a case in which a single individual is responsible for two separate businesses, and it might be difficult to carry out these activities ethically. A breach of the control theory of division of duties, which requires at least two separate parties to be interested in carrying out critical transactions, is referred to as a conflict of duties. When the available resources are insufficient to satisfy the demands placed on them, a conflict of interests arises. Other types of confrontation are more primitive, and are usually (but not always!) avoided by human administrators, such as where an activity is both authorised and prohibited; or where someone is obligated to carry out an action that is prohibited. Human administrators use a mix of systematic and intuitive principles, as well as informal negotiation, to recognise, avert, and settle disputes. They don't really do a good job. Automated processes are expected to take a far more formal path, because if issues are not handled properly, the device will crash completely. This paper investigates how much our policy model should be used to analyse disputes with the aim of preventing, identifying, and resolving them[6].

3. A MANAGEMENT POLICY MODEL

The key type of regulation that is of concern to distributed machine management is management intervention policies; in a nutshell, they define a permanent, positive or negative, imperative or authority for a group of policy subjects to accomplish objectives or activities on a collection of target items. Inside the computer code, software objects may be used to depict human subjects or goals.

Other policies, on the other hand, are difficult to incorporate within this system. For example, the policy that "the same party cannot be allowed to enter a payment and sign the payment check" is difficult to model as a management action policy. Since two management action policies are in question: one authorising X to input accounts for payment and the other authorising X to authorise accounts for payment, it's best defined as a policy regarding management action policies (PAMAP policy). According to PAMAP, the two management intervention plans cannot coexist[7].

We don't know how to model PAMAP policies in a useful way right now, so we have to go "outside the model" to explain them. This isn't relevant right now because there's already a lot of research to be conducted on ordinary management intervention policies and their interrelationships. However, this is a major disadvantage that would need to be solved in the long run. We specify certain characteristics of the proposals we'll be addressing in order to have a more specific working description than merely "plans." We begin with the assumption that policies are meant to affect behaviour. Policies, on the other hand, are unconcerned by split-second actions to take steps that are then immediately carried out. When a boss orders things to be performed just once and immediately, such as 'Shut the door!', he is not creating a policy; however, he is merely causing the operation to be carried out. Since it describes a particular potential activity or recurring acts, or because it refers to the ongoing management of a situation, our concept of strategy necessitates persistence[8].

3.1 Policy Modalities - Imperatival and Authority

We differentiate between policies that are meant as imperatives to trigger acts and policies that grant or withhold authority for actions to take place, as seen in the example above. Actions are operations carried out by agents on goal subjects if two prerequisites are met: imperative and authority:

- **Imperatival** policies are those which cause actions to be initiated (or deterred). A common form of imperative is an obligation which is undertaken by the agent, and many of our examples refer to obligations as a form of imperative;
- **Authority** Policies are those that enable acts to be carried out with authority. Figure 1 depicts our perspective of the universe. Agents are objects that view and implement imperatival policies that they are the targets of. When these policies' requirements are met, the agent performs an activity aimed at a target entity. A reference monitor intercepts all conducted activities and then permits them to continue (authorises them) if the relevant authority policies warrant them. The goal item is affected by the approved

operation.

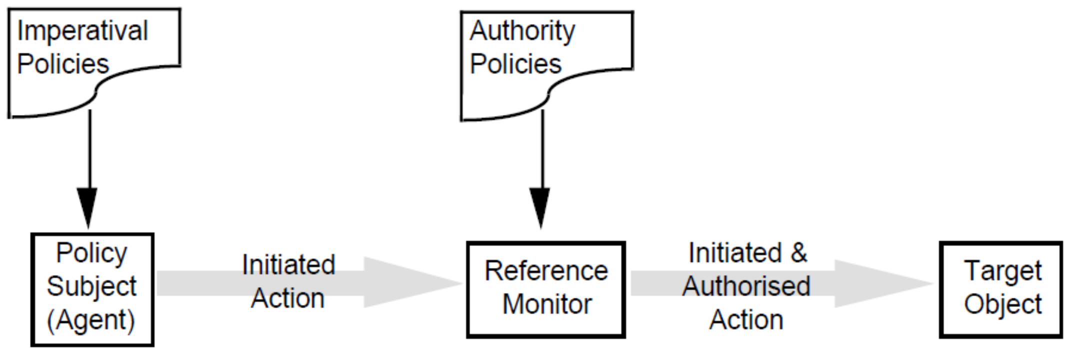

Figure 1 The Roles of Imperatival and Authority Policies

3.2 Policy Attributes

Modality, policy topics, policy aim items, policy priorities, and policy limits are all characteristics of policies, whether they are concerned with imperatives or power. Figure 2 shows how we illustrate policies using a typical graphical convention (without constraints). The topics and aim objects are displayed in traditional Venn diagram convention for graphical simplicity, while the collection of priorities is shown as a list attached to the policy modality. Subjects are described by circles, and goal objects are represented by triangles.

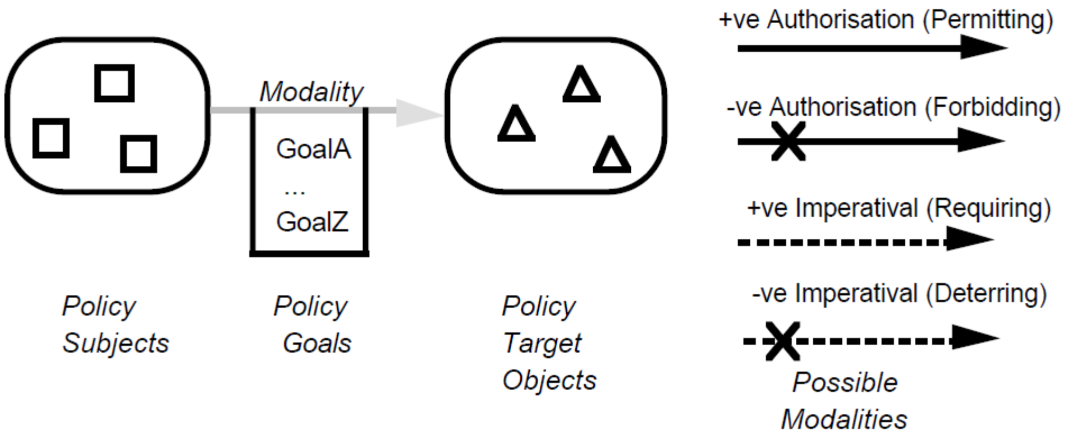
+ve Authorisation (Permitting)
-ve Authorisation (Forbidding)
+ve Imperatival (Requiring)
-ve Imperatival (Deterring)
Modality
GoalA
...
GoalZ
Policy Subjects
Policy Goals
Policy Target Objects
Possible Modalities

Figure 2 A Management Action Policy

3.3 Modality

Positive authority (permitting), negative authority (forbidding), positive imperatival (requiring or obliging), and negative imperatival (forbidding) are the four modalities of a regulation (deterring). We don't rule out the likelihood of other useful policy modalities being proposed, but these are sufficient for our purposes. As previously said, we consider a duty to be a particular kind of imperative.

3.4 Policy Subjects and Target Objects

Policies are for organisational objectives that must be met by others. The policy subjects attribute identifies a group of users with all policies in this model. The policy subjects are the individuals who are affected by the policy, i.e., those who have the responsibility or power to carry out the policy objective under the policy restrictions. When a protocol is automated as a computer machine order, the policy subject is the consumer who will enter the system command. Many rules, by extension, are directed to all users, perhaps bound by some predicate, and the regulation subjects attribute's meaning is the collection of all users[9].

The regulation goal objects attribute specifies which artefacts are targeted by the policy. Any of the policy subjects and goal items may be defined either by enumeration or by a predicate that must be fulfilled. Specific policy topics and goal objects are seldom defined because policies are usually expressed in terms of organisational roles and object realms, rather than entities. Using management domains is one way to define organisational positions and enumerate classes of items. It's worth noting that, although both imperatival and authority policies are specified as sets, the traditional set membership is likely to be different. An authority policy's set of subjects clearly specifies who has the authority to do acts, and there are no issues whether the set includes a significant number of participants. However, when most actions are performed by one person, and where the same goal is assigned to more than one, conflict will arise. It may be appropriate for the subject set to be defined as a position, such as Security Administrator, with more than one member, but then it may be necessary for members.

3.5 Policy Goals

The policy targets attributes can be expressed as a high-level agenda that determines what the planner can do in general terms that do not specify how to accomplish the objectives. Alternatively, the objectives may be reduced to a series of more specific activities that detail how to accomplish the desired outcome. Actions are defined using an alphabet of operations that can be conducted on device properties, making them amenable to automatic interpretation. A high-level aim can be broken down into a variety of different action steps. Refining a target to a series of behaviour is equivalent to refining a set of parameters into a computer program's comprehensive specification.

4. REPRESENTING MANAGEMENT ACTION POLICIES AS OBJECTS

It is useful to view management action policies as objects on which operations can be performed. For simplicity we assume the following minimal set of operations:

- Create a policy;
- Destroy a policy;
- Query a policy.

To execute operations on policy properties, authority might be needed. No restrictions may be required if the computer system is merely a documentation assist. In the other side, limits on activities are needed if the rules are actually used to manipulate device behaviour, as in the case of access control policies.

It is simpler to decide what policies occur and modify them when policies are represented as explicit objects that are viewed by managers. To discourage modification, policies may be rendered read-only if appropriate. Many programmes, on the other hand, specify policies indirectly by coding it into the system's execution or the manager components. Even if encoding a protocol into an implementation is the most realistic way to enforce such rules, there should also be a (high level) policy object to directly define the policy such that it does not get updated for a new device update without the user realising it.

5. OVERLAPPING POLICIES

Both topics and goal items are represented as collections of objects in our model of management intervention policies. When the intersection of two sets of points is not zero, the overlap relationship occurs, as seen in figure 3.

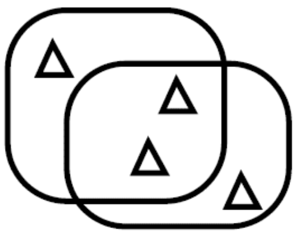

Figure 3 Overlapping Sets

We believe that without any sort of correlation between the subjects in two laws, there will be no disagreement between them, so overlap is critical to our discussion of policy conflicts. We use the kind of overlap as our first stage of classification to analyse conflicts further down. There are many combinations of overlap between subjects in the topic and goal object sets of the policies, leading to different combinations of overlap between policies[10]:

- **Double overlap** - both the subjects and the target objects of the two policies overlap;
- **Subjects & Target objects overlap**;
- **Subjects - Targets overlap** - One policy's topics and another policy's goal items differ. Overlap of any sort is, of necessity, a requirement for certain types of non-conflictual relationships between police forces. Below are a few instances of policy aim entity attributes that overlap but do not conflict.
- **Authority hierarchies**. A well-defined authority structure exists in many organisations. When a senior manager delegated power to subordinate managers, the aim items were normally divided into subsets and allocated to separate managers. The goal object set in the policy that assigned authority to the higher level manager obviously overlaps with the target object sets delegated to the hierarchical managers. An imperatival strategy prevents the higher-level boss from having authority over the assigned priorities, even in the unlikely event of a subordinate manager's breakdown.
- **Imperatival policy hierarchies**. There is apparent overlap between the intended objects of higher and lower level policies as a high level imperative regulation is distilled into more precise lower level policies or collections of activities. This would not cause friction since administrators can just put the more concrete lower-level strategies into effect.
- **Responsibility**. When considering an aim to be accomplished, there is a difference between responsibility for and duty to. We believe that breaking down the principle of duty into two distinct imperatival policies that each refer to the same group of goal subjects would be beneficial. The objects of the two proposals will be the manager who is accountable for meeting the target and the manager who is responsible for the first manager. Current research is being conducted on this subject.

6. MANAGEMENT ACTION POLICY CONFLICTS

Conflicts between management behaviour and strategy are not well known, and we do not claim in this paper to include all potential types of dispute. The policy paradigm, on the other hand, has made systemic study of policy disputes possible. This research is only in its early stages, however it has taken a move forward by recognising that the convergence of artefacts between policies – one or all policy subjects and goal objects – is a critical prerequisite for confrontation.

There is no chance of disagreement between two strategies since they have no objects in common. Figure 5 depicts how we identify disputes. Conflict of modalities and conflict of goals are the two main distinctions. Conflicts of modalities may be identified without regard to the policy goal's context, while conflicts of objectives are either semantically based or application-dependent.

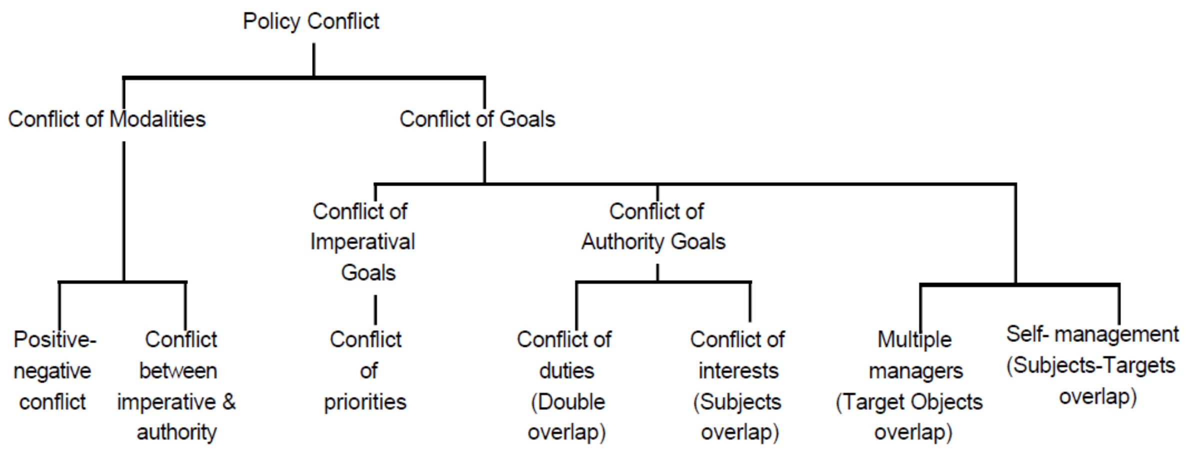
Policy Conflict
Conflict of Modalities
Conflict of Goals
Conflict of Imperatival Goals
Conflict of Authority Goals
Positive-negative conflict
Conflict between imperative & authority
Conflict of priorities
Conflict of duties (Double overlap)
Conflict of interests (Subjects overlap)
Multiple managers (Target Objects overlap)
Self- management (Subjects-Targets overlap)

Figure 5 Category's Policy Conflict

7. CONCLUSIONS

This paper outlined our methodology to formalising management practises, and using the formalisation as a basis for analysing policy disputes. The various forms in which policies may intersect have been discovered to strongly correlate to the informal intuitive grouping of policies. In the long term, automation of policy dispute identification and resolution would be critical for successful automated distributed system management; if any future or current conflict requires human interaction, most of the value of automation would be lost. Before this concept will become a fact, though, improvements must be made on at least three fronts.

To begin, a thorough understanding of the distributed system management application functions is needed. In this article, we have provided our examples based on what is obvious and common. And through understanding the real conflicts that exist will we be certain of the fields that need the most effort. Second, improvement on the generic formalisation of policy is needed. It must be systematic and only then would it be possible to deal with it rationally, whether by modelling models or formal logic reasoning. It must also be generic, and if policy conflicts need to be addressed, they must all fall within a compatible context. Third, the theoretical model's functional applications must be generated and produced. Work on this has begun in ventures like Domino, but there is still a long way to go until it pays off

REFERENCES

[1] D. Skeen and M. Stonebraker, "A Formal Model of Crash Recovery in a Distributed System," IEEE Trans. Softw. Eng., vol. SE-9, no. 3, pp. 219–228, 1983, doi: 10.1109/TSE.1983.236608.

[2] G. Surekha and P. L. Priya, "Categorizing and Analyzing the Data Stream Requirements in Distributed System," vol. 6, no. 2, pp. 622–623, 2018.

[3] R. V. A. N. Renesse, K. P. Birman, and W. Vogels, "P164-Van_Renesse," ACM Trans. Comput. Syst., vol. 21, no. 2, pp. 164–206, 2003.

[4] C. Fidge, “Fundamentals of distributed system observation,” IEEE Softw., vol. 13, no. 6, 1996, doi: 10.1109/52.542297.

[5] F. Cristian and C. Fetzer, “The timed asynchronous distributed system model,” Dig. Pap. - 28th Annu. Int. Symp. Fault-Tolerant Comput. FTCS 1998, vol. 1998-January, pp. 1–10, 1998, doi: 10.1109/71.774912.

[6] J. D. Moffett and M. S. Sloman, “Policy conflict analysis in distributed system management,” J. Organ. Comput., vol. 4, no. 1, pp. 1–22, 1994, doi: 10.1080/10919399409540214.

[7] K. M. Chandy and L. Lamport, “Distributed Snapshots: Determining Global States of Distributed Systems,” ACM Trans. Comput. Syst., vol. 3, no. 1, pp. 63–75, 1985, doi: 10.1145/214451.214456.

[8] K. M. Chandy et al., “World-wide distributed system using Java and the Internet,” IEEE Int. Symp. High Perform. Distrib. Comput. Proc., pp. 11–18, 1996, doi: 10.1109/hpdc.1996.546168.

[9] A. Vermeulen, “for the Long Term Hormonal,” Endocrinol. Metab., vol. 74, no. 4, 1992.

[10] D. Gunter and B. Tierney, “NetLogger: A toolkit for distributed system performance tuning and debugging,” IFIP Adv. Inf. Commun. Technol., vol. 118, pp. 97–100, 2003, doi: 10.1007/978-0-387-35674-7.

[11] Neela Madheswari A, "The availability of Workloads for Grid Computing Environments", International Journal of Engineering Research and Technology, p.no. 211-213, 2015