Chapter 6

Conflict Resolution

There are several well-known phrases describing policy conflicts, for example: Conflict of interests describes a situation where a single person has tasks relating to two different enterprises, and carrying out both together conscientiously may be impossible. Conflict of duties is the dual situation; it describes a failure of the control principle of separation of duties requiring at least two different people to be involved in carrying out important transactions. Conflict of priorities occurs when the resources available are not sufficient to meet the demands upon them. Other forms of conflict are typically avoided by human managers, for example: an action is simultaneously authorised and forbidden; or, someone has a duty to carry out an action, which is forbidden.[Moffet94]

Conflicts among policies produce mistakes in the running of the system or an unpredictable behaviour can appear too. Human administrators can solve such conflicts, however, policy-based management tries to solve all conflicts among policies in an automatic way [Lupu99] [Chomicki00].

The process of conflicts detection starts with a suitable validation process before the storage of policies in the repository. It is essential to revise that the syntax of the policies is correct. The syntactic validation depends on the used language; for example, in the case of XML they follow the syntax of the data type definition (DTD) that a set of rules establishes to specify the allowed marks and the circumstances under which they are allowed. Therefore, the initial stage to validate the policies specified in XML is to check that they accomplish all DTD specifications. The second stage consists of making quite sure that policies satisfy all system restrictions, both topology restrictions and other restrictions created by the network administrator. All attributes values forming a policy must be verified as valid. The attribute values can depend on the values that other attribute in another policy owns. From edition point of view, the last stage is to validate that the defined policy is not responsible for any conflict with any other previously defined policy.

Conflicts can also happen at the execution time. Due to this, the existence of a conflict must be noticed and once it has been considered, it is necessary to know how to treat it. It is very important to know and classify all classes of conflicts that can happen potentially in our system. Some conflict examples are given next:

1. If a policy establishes that traffic from marketing department has high priority and other policy establishes that the web traffic has low priority, then a problem can emerge when someone from the marketing department wants to access to a web page; the router will doubt whether to give it high or low priority.
2. Si se consideran las siguientes políticas:

   Policy 1: *Any access to the WebServer has Class of Service 1.*

   Policy 2*: Users with high priority in the SLS profile have Class of Service 2.*

   Both policies work properly on their own, but a problem exists when they two

   Have to be applied in the network. For example, in case any user from the High Priority Users group wants to have access to a service of the WebServer, it

will doubt whether to give Class of Service 1 or 2.

3. Another example is if a policy says that a specific user has access to the system and another policy says that the same user does not have access. There is no means of deciding whether the action is to be permitted, and unless there is some means of resolution this could lead to a deadlocked situation if the conflict were encountered in an automated system. These kinds of conflicts come from policies of the admission control module, there is some literature on policies for Admission Control that specify different languages to clearly assign access authorizations and authorization rules in different contexts (relational models, object oriented models, multimedia systems, XML, etc. Some of these topics are discussed in [Jajodia 00].
4. When a packet arrives at a router, there are one or more policies that can be applied. If all applicable policy actions can be executed, then there is no conflict. On the other hand, if the actions of two applicable policies cannot be executed (for example, a policy establishes a DiffServ field to 0 and other policy establishes the DiffServ field to 1), then there is a conflict.

There are different ways to detect a conflict among policies, in [Damianou01] a tool is proposed to analyse conflicts among policies. This tool belongs to a policy roles-based management framework. Methods that detect runtime conflicts between policies act at the moment in which a policy has to be applied. This engine would determine the logic conflicts, however, high speed is required to analyse at real time if a conflict can appear among a big set of policies.

Because of our policy representation is based on the PCIM scheme, we take advantage of the disjunctive normal representation (the set of policies is a disjunction of rules where every rule consists of AND predicates) and also we take advantage of the conjunctive normal representation (the set of policies is a conjunction of rules where every rule consists of OR predicates) [Moore01]. If in this set there were two rules that are contradictory, as for example, a rule and its logic negation, the set of policies would have a logic contradiction. Expert systems have efficient techniques to find these kind of logic

contradictions. However, we also have to analyse when a set of policies is in conflict without having logic contradictions.

Conflicts happen in different circumstances, for example when policies are applied in the network and there are other policies taking contradictory values in relation to other policy values. A method to solve this problem is the fact of considering all possible characteristics from a situation (all combinations of users and applications) and tries to establish all the necessary policies to eliminate ambiguity and conflicts. The disadvantage is that, as a result, an exponential execution time algorithm can appear.

In this proposal, all requirements from service managers, performance managers or mistakes ones, follow a validity process to keep consistence and integrity in the network. Despite this validation, it can happen that different managers have performing requirements causing any kind of conflict and make the network fall in a state of inconsistency. These conflicts can be avoided by means of suitable modelling and design techniques based on the policy roles that we define, the generic policy and a method of conflict resolution that use graphical policy representations in a hyper dimensional space.

Each role has a default policy that is applied when any other policy can be selected. In this way, to avoid the inconsistencies of the system, we have to do an analysis of the system coverage reached by the policies. The application of the default policy avoids specific situations in the network for which we cannot apply any policy. This kind of conflicts is avoided by means of a correct design of the policies that control the different functional areas of the management system.

In this sense, a module was designed to detect conflicts among policies. It operates just before the storage process of policies in the LDAP directory. This module, before allowing a policy to be stored, searches in the directory if there is a policy that has been previously stored with the same conditions and a different action, that is to say, contradictory policies that originate certain circumstances in the network and avoid making a clear decision about what policy to apply in the network.

There can also emerge conflicts in the network when, under certain network circumstances, more than one policy of the same role have to be applied. We detect this kind of potential conflicts when there is an overlapping between the cubes that represents the policies from the same role that can be applied.

In order to avoid this kind of conflicts, a policy called generic policy was designed. This policy indicates the item to which priority must be given in the policy selection process. Some items that can be given this priority are the network resources, the QoS offered to the user, the business order, a specific service priority, expense reduction, etc. Once we know the generic policy indication, then the minimum distance between that item and the policies in conflict is computed. For example, if the generic policy indicates to give priority to the network resources, then the distance from the average point of every policy to the three-dimensional space origin is calculated, choosing the biggest distance to accomplish the condition indicated by the generic policy, that is to say, the furthest policy from the origin is chosen due to the fact that at that point is where the delay and losses own minimum values and where the BW owns the maximum values. In the origin is where the user takes advantage of the situation and far from the origin is the network that takes it.

6.1 Geometric representations for the resolution of conflicts

A graphic representation of the policies defined according to the parameters and metrics on which the conditions are formulated is very useful in order to analyse conflicts. With this representation it is possible to detect overlaps among policies both graphically and analytically. In this way, without some kind of overlap between the policies there can be no conflict between them.

In case an overlap occurs and different policies could be applied before a user's request and the network status, some criteria can be applied, as for example, the way to find the distance from a geometric point required by the user to the maximum optimum point of the

policy geometric distribution. Therefore, it is necessary to formulate the policy geometric function mathematically.

The overlap relationship between sets of objects exists when their intersection is nonempty, as shown in figure 1. We analyse conflicts using the kind of overlap as our last level of classification, first we tried to apply generic and default policies. Overlapping can occur when the value of some metric correspond between two or more policies from different roles.

There is also necessary to do an analysis in order to detect if an overlapping involves or not a conflict. Next there is an example in which attributes of policies overlap but there is no conflict: When a high level imperatival policy is refined into more concrete lower level policies or sets of actions, there is an overlap between the high and lower level policies. This does not lead to conflict, as managers will translate only the more concrete lower level policies into action. For this reason, it is important to recognise if the overlap of policies is a conflict. If there are no objects at all in common between two policies, there is no possibility of conflict.

In reference to the policy definition related to routing according to the Quality of Service, three metrics are used (delay, losses and bit-rate). In order to show graphic simplicity, policies are defined as cubic in a three-dimension space. See figure:

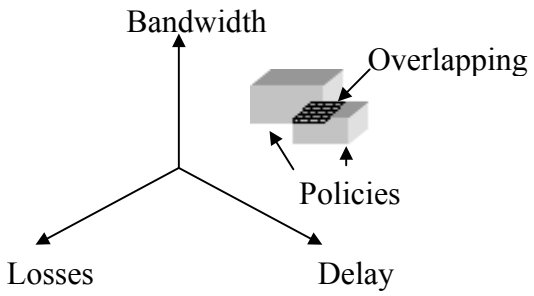

Figure1 . Representation of policies according to three dependent parameters

In case we want to implicate more dependent parameters in the space (e.g. four dimensions), we would need other expressions like the hypercube or the hyper-sphere. In this sense, specific representation tools have been implemented to make the study about potentiality for conflicts among policies easier. See next figures and summary in appendix II.5.

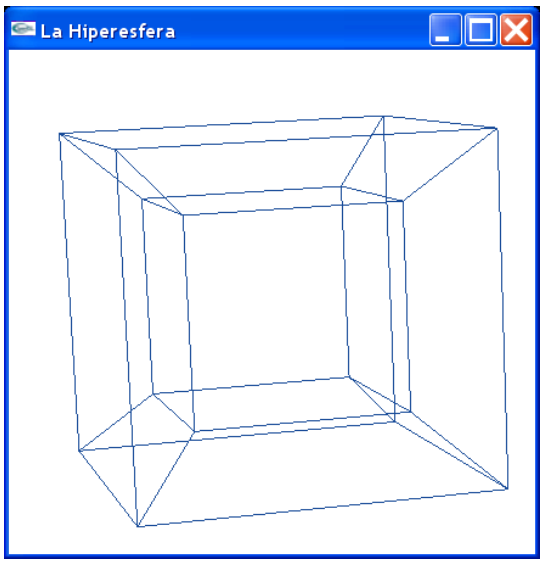

Figure 2. Representation in hypercube of the policy space according to four metrics

When the radium is small, we can work with the hypercube; when the radium tends to infinite, we can substitute the hypercube by a hyper-sphere.

A hyper-sphere, in a four-dimension space, is the geometric place of x, y, z and w points that satisfy the relation of the fact that the addition of the squares of the coordinates is equal to the square of the hyper-sphere radium. The mathematic analysis of these structures can be arranged systematically by means of the finite element method.

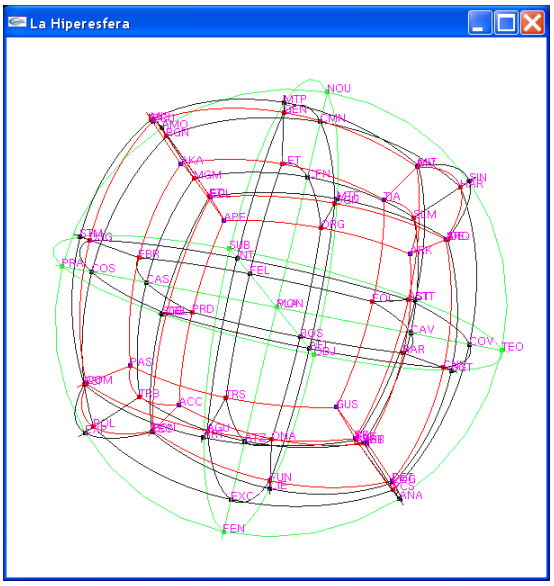

Figure 3. Representation in hyper-sphere of policy space according to tour metrics

6.1.1 FEM (Finite Element Method)

The new FEM techniques can be used to solve mathematically policy spaces with continuous structures (surfaces and volumes) and with complex geometrics. FEM is a numeric method about differential equations resolution. It works discreeting the structure into divers-shaped elements (they can be surfaces, volumes and bars) that are connected by means of nodes. Solution is just approximate by now depending on the nodes results. FEM starts from the matrix computation in the nodes balance planning by means of an equation system coming from the contribution of the elements.

Nowadays, FEM shares leadership with the matrix method. There are a lot of programs that mix the analysis carried out by both methods. The application of FEM has been left for the analysis of continuous elements (type flat or screen), while porticos are still being discreeted into bars and using the matrix method.

The steps to follow in the analysis of structures by means of the method about displacements through FEM are:

1. The whole is divided by means of lines or imaginary surfaces in a number of finite elements. This process stage is usually developed by means of algorithms incorporated to wire-netting computer programs during the pre-process stage.

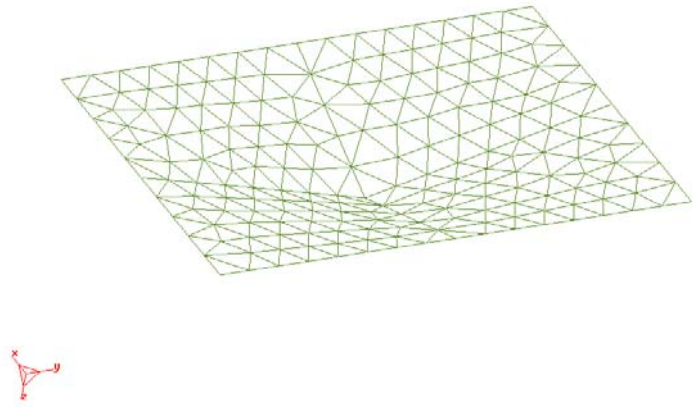

Figure 4.Techniques FEM

2. All elements are supposed to be connected among them by means of a discreet number of points or nodes placed on their perimeter. The displacement of these nodes will be the key question of the problem as it happens in the simple analysis of structures by means of the matrix method.

3. We can take a set of functions defining exclusively the displacements field within every "finite element" according to the function of the nodal displacements of the mentioned element. For example, the displacements field within a lineal element of two nodes could be defined by: $u= N_1\ u_1 + N_2\ u_2$, being N_1 y N_2 the mentioned functions (shape functions) and u_1 and u_2 the displacements in the node 1 and in the node 2.

4. These displacement functions will exclusively define the deformation of the element according to the nodal displacements. These deformations, together with the constitutive characteristics of the material, will also define the tension status in the entire element and consequently in its perimeter.

5. A strength system concentrated on the nodes is then determined in order to balance all the tensions in the perimeter and any other tendencies or delivered previsions, which results in a relation between strength and displacements in the shape of F = k . u. It is very similar to the matrix computation.

6. Last system resolution allows to obtain the displacements in the nodes and to define with them the displacements field in the finite element approximately.

7. In the post-process stage, all results are presented, generally in a graphic way in order to analyse them.

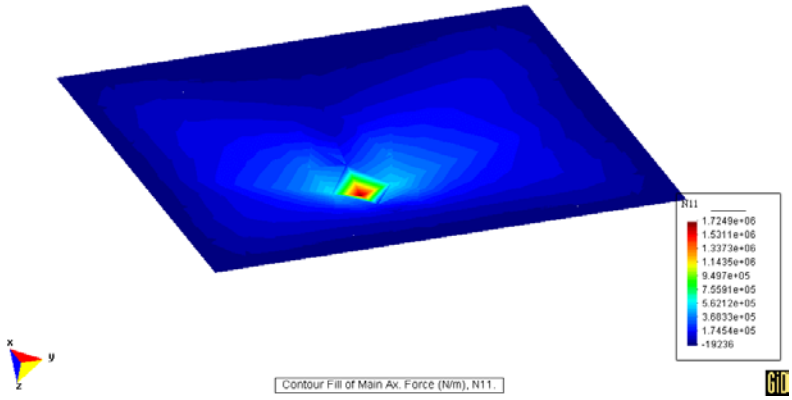

Figure 5. Representation

6.2 Contributions to the chapter

In this chapter we present an innovative way to solve conflicts among policies and letting the PBMS systems to support scalability for large networks.

There are several different levels at which conflicts can be prevented or resolved. At the highest level, the language used for system description may be designed to prevent and to resolve any conflict. Our system also detects potential conflicts before the storage of the policy using a first conflict detector process that analyse new policies do not have conflict with other previously stored policy.

Later, there is a logical revision via the two PCIM representations (disjunctive and conjunctive). We also use elements such as: the generic policy and default policies in each role.

Potential conflicts of interest may be detected by application-specific tools, which are aware of conflicting goals. By other hand, runtime conflicts may be detected in real time or in advance using prediction schemes in order to avoid them.

The most interesting part is the conflict resolution via mathematic analysis corresponding to the geometric representation of policies, which can be solved by means of evaluation techniques about finite methods (FEM).

References

[Barba02] Antonio Barba, Joel García, Lluis M. Xirinacs..*Manual para representaciones geométricas de políticas*. Internal Report ENTEL, UPC. 2002.

[Chomicki00] Chomicki J., Lobo J., Naqvi S., *A Logic Programming Approach to Conflict Resolution in Policy Management.* Principles of Knowledge Representation and Reasoning. Publisher Morgan Kaufmann, San Francisco, USA. pp. 121-132 2000

[Damianou01] N. Damianou, N. Dulay, et.al. *The Ponder Policy Specification Language*. Policy 2001: Workshop on Policies for Distributed Systems and Networks, Bristol, UK, Springer-Verlag 2001

[Jajodia00] S. Jajodia, P. Samarati, et al. *Flexible Support for Multiple Access Control Policies*. ACM transactions on database systems 26(2) pp. 214-260. 2000.

[Lupu97] Lupu Emil, Sloman Morris. *Conflict Analysis for Management Policies.* Fifth IFIP/IEEE International Symposium on Integrated Network Management IM'97. San-Diego, May 1997. Publisher Chapman & Hall.

[Lupu99] Lupu E., Sloman M. *Conflicts in Policy-based Distributed Systems Management*. IEEE Transaction on Software Engineering, Vol 25. No.6 pp. 852-869 November/December 1999

[Moffett94] Moffett, J.D. and Sloman M.S., *Policy Conflict Analysis in Distributed System Management*, Journal of Organizational Computing 4(1), pp 1-22, 1994

[Moore01] Moore, E. Ellesson, J. Strassner, A. Westerinen. *Policy Core Information Model -- Version 1 Specification.* IETF Request for Comment (RFC) 3060. February 2001.

[Sloman99] Sloman, M., Lupu E., *Conflict Analysis for Management Policies.* Integrated NetworkManagement VI IEEE U.S.A. May 1999.

[Zienckiewicz] O. Zienckiewicz - R. L. Taylor. *El Método de los Elementos Finitos*. Ed. McGraw Hill / CIMNE