Managing QoS in a NGN using a PBM approach

Pedro Gonçalves, José Luis Oliveira, Rui Aguiar
ESTGA, DETI, IT
Universidade de Aveiro
Aveiro, Portugal
{pasg, jlo, ruilaa}@ua.pt

Ricardo Azevedo
Applied Research and Knowledge Dissemination
Portugal Telecom Inovação
Aveiro, Portugal
ricardo-a-pereira@ptinovacao.pt

Abstract—**Next Generation Network (NGN) management represents an enormous challenge due to the large number of managed elements, the variety of roles the managed entities play in the network and the difficulty of orchestrating management actions. Several NGN standardization bodies point toward Policy Based Management as the best approach for NGN network management.**

This paper describes a management solution for a NGN IP Multimedia Subsystem (IMS) QoS management scenario based on WBEM technology. The proposal is for a WBEM-based policy server, a graphical policy editor application and instrumentation logic for the NGN QoS Management. The graphical policy editor reduces the learning curve imposed by the policy specification language, allowing the specification of policies through a rich and user-friendly visual interface, hiding the CIM syntax complexity but keeping CIM's potential.

Index Terms— PBM, Policy edition, IMS, NGN, CIM

I. Introduction

The move toward all IP observed in the last two decades made it very easy to create new companies offering new and revolutionary services that started to compete with traditional communication services like the old telephone service. The competition created by new services, typically belonging to companies that offer Internet services as well, seriously reduced the revenues of the telecom operators. These companies felt they were giving their golden revenues by offering IP connectivity that allowed the development of new services by new companies.

The operators' response to the deployment, in a controlled environment, of those new services in a heterogeneous IP network was the creation of a new network architecture named Next Generation Network (NGN). Several standardization bodies – ITU-T [1], ETSI TISPAN [2], 3GPP [3] or Broadband Forum [4] – have been involved in normalization of the NGN architecture. Generally NGN network architectures consist of a layered architecture that standardizes interaction between the network client, the network platform and the service providers. They define open standards that allow service providers to develop new services, integrating management of the network platform equipment.

The management of such a platform is a very complex challenge: it integrates management issues from several management areas (*e.g.* resource management, user management and service management). Those have to be configured in a coordinated fashion in order to allow seamless operation. Moreover, network elements typically have different models, different manufacturers and even different configuration models. A seamless network operation requires equipment configuration to be performed in an integrated way, in a short period of time and with conflict detection / resolution support.

Policy Based Management (PBM) is being identified as the most appropriate approach for such a complex scenario, also by the NGN standardization entities [2, 4]. In fact, 3GPP in release 7 defined a *Policy and Charging Control* architecture [5], which presents many similarities to the basics of a PBM.

This paper describes a management solution for a NGN QoS management scenario. Section 2 describes the NGN network as well as its management issues. Section 3 describes the policy-based management technologies used in the management platform and section 4 presents the proposed management architecture. Section 5 identifies and specifies a set of policies to illustrate the policy specification process proposed in the paper. Finally section 6 draws some conclusions.

II. Next Generation Networks

Next Generation Networking (NGN) is a broad term to describe some key architectural evolutions in core and access telecommunication networks. The common idea behind NGN is that one network transports all information and services (voice, data, and all sorts of media such as video). NGNs are commonly built around the Internet Protocol, and therefore the term *all-IP* is also sometimes used to describe the transformation towards NGN. ITU-T defines it as a packet-based network able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent of underlying transport-related technologies. It offers unrestricted access by users to different service providers and it supports generalized mobility which will allow consistent and ubiquitous provision of services to users [1].

A. IP Multimedia Subsystem

IP Multimedia Subsystem (IMS) was proposed by the Third Generation Partnership Project (3GPP) as an overlay framework to deliver multimedia services in mobile IP networks [3]. Other standardization bodies, such as ITU and ETSI, have adopted this framework for their NGN proposals.

IMS is a layered architecture that separates the service, the control and the transport planes, offering significant benefits in terms of service creation and maintenance savings. Its framework is agnostic in terms of access network technology and it has been receiving great attention from the ESTI TISPAN in order to achieve fixed mobile convergence.

The call/session control layer is composed of three entities collectively called *Call Session Control Function* (CSCF) that process SIP signaling packets in the IMS world. The *Proxy*-CSCF is the initial interface between the terminal and the IMS core functions. Among other features, the P-CSCF is responsible for forwarding QoS requests to the policy control layer. *Interrogating*-CSCF is the function within the home network that is able to determine the *Serving*-CSCF with which a user should register. *Serving*-CSCF is the function that registers the user and provides him with the service. It performs routing and translation, provides billing information, maintains session timers, and interrogates the HSS (*Home Subscriber Server*) to retrieve authorization, service triggering information and user profile.

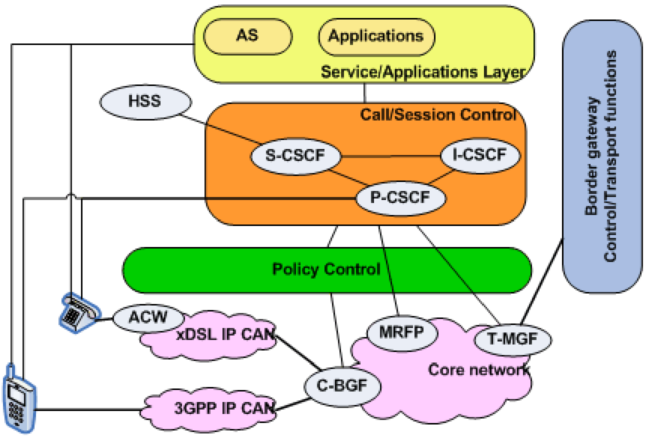


Figure 1 - IMS/TISPAN architecture

Since service and access network levels are separated, different network and control architectures could exist within the same service enabler architecture (IMS). TISPAN NGN control layer [2] can be introduced to perform the interaction between the two levels during session establishment, modification, tear down, or user network attachment. In such a control layer, two main elements exist to guarantee that transport elements are well configured and ready for the user's traffic. One of the control elements is the Resource and Admission Control Sub-system (RACS) [6]. This ensures that the resources reserved in the transport elements correspond to the session negotiated at the service level and allowed by the operator policies and user subscription profile.

Basically, RACS provides policy-based transport control of the services to the applications. This enables applications to request and reserve transport resources from the transport networks within the scope of RACS. Its scope extends to the access and core networks, as well as to points of interconnection between them in order to support end-to-end QoS. By offering a level of hidden interaction between applications and the transport resources themselves, RACS also ensures that applications do not need to be aware of the underlying transport networks [6], since it hides network topology and technology details from applications and IMS entities.

Another control entity is the Network Attachment Sub-System (NASS) [7]. This entity is responsible for the user's authentication and authorization. It acts as an authentication server, and provides IP connectivity information for the user, i.e., IP address, default gateway, in the attachment process. User's location (port Id and circuit Id for the ADSL scenario) is also stored in the NASS database. This information could be given to applications upon request and based on operators' policies.

III. Policy-Based Management

Policy-Based Management (PBM) is a management paradigm that uses policies for system management. Policies are rules that govern a system behavior, usually implemented in a form of *if(condition) then (action)* sentences.

PBM paradigm allows an abstraction of the vendor's specific configuration details through use of a policy manager element. The automation character of the management approach reduces the equipment management effort. This is of special importance in large management scenarios like a communication operator network. Moreover, during the configuration process, it avoids possible human error due to the repetitive actions performed by human operators. Furthermore, central implementation of the configuration actions eases implementation of conflict detection in the configuration elements as well as implementation of transaction support for the configuration actions.

During the last two decades several technologies related to PBM management approach [8-10] and policy specification languages [11, 12] have been promoted. To satisfy the necessities of such definitions several data models, for policy information representation, were also specified [13].

The following subsections describe the PBM technology used in our management platform.

A. WBEM

Web Based Enterprise Management (WBEM) [8] is a management technology initially proposed by several companies like Microsoft, Compaq, BMC Software, Cisco Systems, and Intel to the Distributed Management Task Force (DMTF). The initial objectives were to propose a management platform where the management systems could share the systems' information in the agnostic fashion of a vendor. Reusing the technologies was also a requirement.

The information model chosen to represent the management information was the Common Information Model (CIM) data model [14], a previous DMTF standard. The information encoding and transport technology choices were influenced by some Web technologies/protocols which were very popular at that time: i) XML was chosen for information encoding [15] and ii) HTTP for management information transport [16].

WBEM specification leads to several open-source implementations [17-19]. All the projects were maintained by some *big* vendor, which led to several commercial solutions in this area (*e.g.* IBM Tivoli, CiscoWorks2000 from Cisco, and Solaris WBEM service).

Typically, WBEM-based solutions contain four elements: i) a CIM Object Manager (CIMOM) acting as a central management server, ii) a management information repository that stores management information and is handled by the CIMOM, iii) a management console that interfaces with the human operator and, finally, iv) a management provider that interfaces with the managed elements. Management providers implement the management logic adaptation for the technology-specific details of the managed elements belonging to the management platform.

B. CIM data model

CIM [14] is a standard developed inside DMTF for representation of management information. CIM is an object oriented model that represents all the IT components. It includes expressions for common elements that must be presented to management applications (*e.g.* object classes, properties, methods and associations). Due to its expressive nature, it is possible to translate between CIM and other information models. The elements of the model are Schemas, Classes, Properties and Methods. The model also supports Indications and Associations as types of Classes and References as types of Properties.

CIM includes several sub-models: i) the Core Model comprised of several associations and abstract classes that represent the generic characteristics common to all management areas; ii) the Common Model, represented as the ellipses in Figure 2, comprised a set of management information models related to each management area, but independent of the implementation details; and iii) the Extensions Schema, not represented in the Figure 2, which extends the Common Model classes in order to represent the technology-specific details.

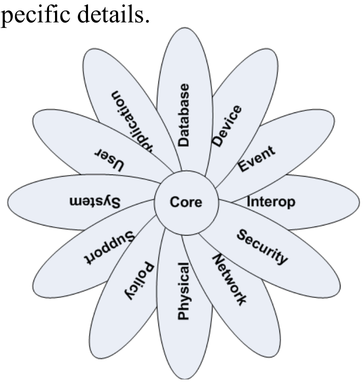


Figure 2 - CIM data model

Several management technologies reused the CIM information model [8, 20, 21], or were at least based on the CIM information model [22].

C. CIM Policy data model

CIM Policy model [13], illustrated in Figure 3, is an information model jointly developed by DMTF and IETF Policy Framework Work Group as part of the CIM schema. Policies are represented as instances of the CIM Policy model classes in a *if(condition) then (action)* form.

CIM_PolicyRule represents the CIM policies organized in groups in a form of *CIM_PolicyGroup*, implementing aggregation as an instance of *PolicyRuleInPolicyGroup*. Policy components can be represented as instances of *CIM_PolicyCondition* and *CIM_PolicyAction*. Policies, conditions and actions can be aggregated and associated with a policy through the usage of association classes (*CIMConditionInPolicyRule* and *CIMActionInPolicyRule*). A separated policy condition named *CIM_PolicyTimePeriodCondition* represents a time / period condition in which the policy is valid.

Implementation of a policy rule in CIM typically instantiates objects of the classes *CIM_PolicyRule* and *CIM_TimePeriodCondition* and instantiates extensions of the *CIM_PolicyAction* and the *CIM_PolicyCondition* classes.

The CIM Policy data model allows definition of policy scopes represented through an aggregation named *PolicySetAppliesToElement* between the *CIM_ManagedElement* and *CIM_PolicySet* classes. The aggregation represents the managed elements in which the policy is valid.

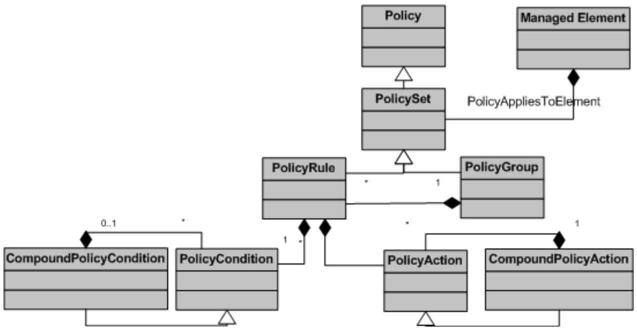


Figure 3 - CIM Policy data model

IV. The Management Solution

This section describes the implemented management architecture. The overall architecture and its principal elements is presented, as well as the CIM extensions developed for policy implementation.

A. Overall architecture

The developed architecture, illustrated in Figure 4, is organized in three layers. A central Management Server, which manages control layer entities, is the main entity of the management layer. The Management Server can be configured by the network operator through a graphical interface application. After definition of the policies, the graphical application inserts the policy information in the Management Server. The CIM Object Manager (CIMOM) inserts the policy information in an internal repository, as well as the management information collected by the lower layer entities. Our Management Server was built on an open-source implementation of a WBEM based management server [17].

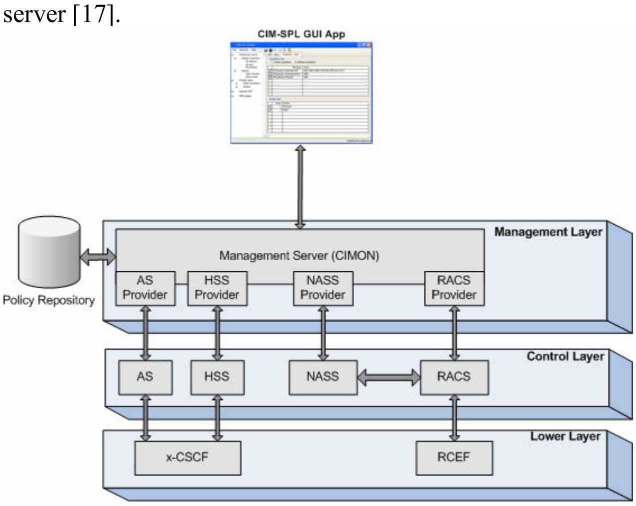


Figure 4 - The management three layers architecture.

The Control Layer is composed of specialized management elements: RACS for the resource management, Network attachment Subsystem (NASS) for the authorization management, Home Subscriber Server (HSS) for the user profile management and the Application Server (AS) for service and application management. Each of the entities belonging to the control layer of the management architecture is in charge of the management of some entity type, or of some particular aspect related to the lower layer of the management platform.

The Lower Layer presents the operational entities: the RCEF implements the admission control decisions as well as the network resource management taken from RACS, and the x-CSCF perform service admission control according to the configuration information defined by HSS and AS.

Although a broader architecture is proposed, as illustrated in Figure 4, our implementation strategy was restricted to QoS assurance.

B. *CIM Policy extensions*

In our implementation we developed several extensions to the CIM Policy data model in order to represent our needs in terms of policy information, following an *action-condition* approach. Figure 5 illustrates the developed model.

Conditions are represented as a conjunction or a disjunction of a set of small conditions. Conditions were created as an extension of *RACSCondition* in order to inherit the properties common to all conditions. The actions are defined as a list of actions that should be executed when the policy condition is evaluated to be true. The class *RACSAction* encapsulates the common properties of the actions implemented by our management solution.

Figure 5 also presents, as examples, some instantiations of action policies that the RACS entity may implement in a real operator scenario. Three of them are briefly described next:

i) **Firewalling** is the act by which the RACS instructs the transport element to open a specific pin-hole for the traffic description requested by the IMS layer. Without the enforcement action provided by this action the traffic is simply dropped at the transport element.

ii) **Shape** is one of the possible techniques. If the request is made for a specific traffic that belongs to a specific contract or traffic profile, traffic shaping can be used to impose additional delay on packets.

iii) **Mark** – There are some scenarios where firewall and shape are not sufficient. If different network segments have different QoS architectures (e.g. IntServ and DiffServ, or Diffserv with different DSCPs), the transport elements in the border of such segments should be able to re-mark traffic packets. RACS, which has knowledge of network architectures in its own domain, should enforce the re-marking schema in the edge routers.

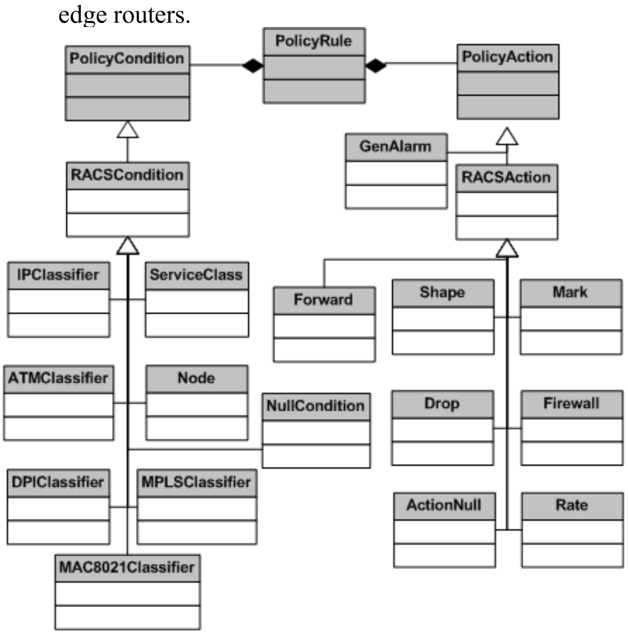


Figure 5 - The management information model

The RACS implementation instantiates the model presented in Figure 5. It implements the abstract behavior of a policy server – each time an event is triggered by any external entity a set of conditions are verified and a set of actions are performed. Based on such implementation it is effortless, from a development point of view, to support different network and service requirements. Since the behavior of the component is completely configured by the definition of policies stored in the database, support for new services or network elements is just a matter of database manipulation. This functionality provides, in addition, a huge amount of freedom to the network administrator – it is possible, for example, to change the events, the conditions and the actions during run-time, giving the RACS the possibility to be configured on the fly and able to work in almost every scenario.

V. Policy Specification

This section describes the policy specification process: it exemplifies the policies specified by the proposed management infrastructure and describes the policy specification process.

A. *Implemented policies*

QoS management in a NGN operator environment requires some traffic conditioning and actions to be performed. Several QoS rules were defined based on the actions that could be implemented by the lower layer management logic.

Blocking of a service or a specific server address (e.g. forbid Skype servers access), the definition of QoS guarantees of some traffic flow (e.g. degrade P2P traffic), the reformatting of traffic (e.g. mark packet labels before the traffic comes in a MPLS network) and firewall control (e.g. opening the transport element firewall) are examples of such policies.

Table 1 presents four examples of policies, applied to the QoS domain in a NGN scenario. Conditions and actions are also referred to.

Nevertheless it is important to state, even if it is not dealt with in this paper, that not only QoS management could be defined as policies. Another example is emergency calls; information about how an emergency call should be routed within operator entities can also be considered as a policy, with its conditions and actions, defined by the network administrator. When an emergency call signaling enters the network, the elements must verify location information, network capabilities and state, and finally route the signaling to the appropriate elements in order to be quickly answered.

Table 1 – Policy description

Policy Name	**Policy Description**		
	Purpose	***Conditions***	***Actions***
Forbide Skype	Prohibit access to Skype traffic	Traffic pattern	Drop
Degrade P2P	Dregradate access for P2P traffic	Port Number Protocol Traffic pattern	Mark
MPLS labels	Mark packets with correct labels	Dest. Address DSCP	Mark

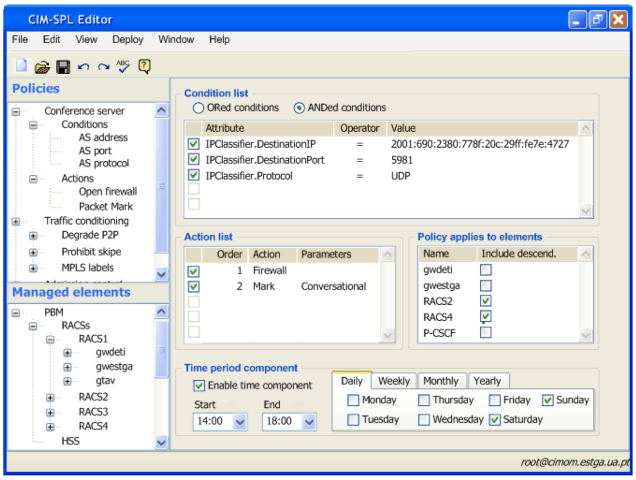
Figure 6 illustrates the policy definition of a rule that allows access to an application server, for instance a conferencing server. Policy conditions include the packet destination address, the packet destination port and the packet transport protocol. Once the conditions are verified, they are sent to the RACS in order to enforce them into the transport element – open the correct pin-hold and re-mark the packets for the purpose of receiving different QoS conditions.

B. *Policy edition*

Although very flexible and appropriate for NGN scenarios, CIM policy specification requires high technical expertise from the network administrators, and thorough knowledge of a policy specification language like CIM Query Language (CQL) [23] or PONDER [11] or CIM-SPL [24]. Two main reasons exist for the difficult learning curve: i) the operator needs to master the language syntax, ii) and he has to have thorough knowledge of policy semantics in order to apply the correct actions and evaluate the correct conditions. Moreover, as was illustrated by Westerinen *et al.* in [25], some policy specifications can lead to very large policy sentences, which could increase the error factor.

Our strategy for the policy editor development was to make use of a graphical metaphor [26] for policy composition. The policy editor tool enables a dual policy specification process: (i) the operator creates a new blank policy, and then has to enter the correct policy components, or (ii) he can just use a graphical wizard to guide him filling in the policy components.

The wizard process is composed of a set of dialog windows that ask for the policy conditions, as well as the condition values that should be added to the policy being defined. Once this two-step wizard policy definition is complete, the data is inserted in the CIMOM. After creation of policies the administrator can change values, as illustrated in Figure 6, where the conditions and actions for each policy are presented graphically.



Policies:
- Conference server
  - Conditions
    - AS address
    - AS port
    - AS protocol
  - Actions
    - Open firewall
    - Packet Mark
- Traffic conditioning
  - Degrade P2P
  - Prohibit skipe
  - MPLS labels

Managed elements:
- PBM
  - RACSs
    - RACS1
      - gwdeti
      - gwestga
      - gtav
    - RACS2
    - RACS3
    - RACS4
  - HSS

Condition list: ○ ORed conditions ● ANDed conditions

	Attribute	Operator	Value
☑	IPClassifier.DestinationIP	=	2001:690:2380:778f:20c:29ff:fe7e:4727
☑	IPClassifier.DestinationPort	=	5981
☑	IPClassifier.Protocol	=	UDP
□			
□			

Action list:

	Order	Action	Parameters
☑	1	Firewall	
☑	2	Mark	Conversational
□			
□			
□			

Policy applies to elements:

Name	Include descend.
gwdeti	□
gwestga	□
RACS2	☑
RACS4	☑
P-CSCF	□

Time period component: ☑ Enable time component
Start 14:00 End 18:00
□ Monday □ Tuesday □ Thursday □ Wednesday □ Friday ☑ Saturday ☑ Sunday

Figure 6 – Graphical editor view

The policy editor is composed of two windows: a left side tree view control that lists the policies and policy groups that exist in the CIMOM repository and their components. Within the same panel a tree control window shows the managed entities that belong to the management scenario. The right side window presents the policy content (conditions and actions). Parent nodes of the tree view are CIM policies organized in two nodes, conditions and actions. The application allows composition of conditions in the form of a logical expression (conjunctions and disjunctions). The policy actions are grouped as descendants of actions node and are ordered according to the action order value.

The right side window changes its content according to the node selected in the tree control window. If the root node is selected, a panel listing all the policies' names is shown allowing the administrator to uncheck a policy, changing its state. Only the policies in active state are executed by the CIMOM, inactive policies being simply stored in the database and ignored during the CIMOM operating process. If the policy's name node is selected, a panel listing all the components of the selected policy is shown. The condition grid control includes the condition attribute, the condition value and a check box that enables or disables the sub-condition verification process. In the action grid the editor includes the action order value, the action name and a check box for the action disabling process.

VI. CONCLUSIONS

This paper presents a PBM management architecture for a NGN - IMS network scenario. The proposed management platform includes a complete flow for policy delivery from the human operator up to the lower management components. The proposed architecture was demonstrated by specification of several rules for the NGN QoS management domain.

A policy specification user interface application is proposed based on a graphical metaphor. The editor allows policy specification in an easy and user-friendly way by non-expert professionals saving human operators from a costly learning process of the language details. The graphical components used in the editor liberate the manager from the semantic and syntactical details of policy specification language. The editor includes an assisted policy creation wizard, which allows the user to fill in policy components in the successive dialog windows.

The proposed hierarchical management architecture is based on the vision that the entire network could be mapped to a set of policy functions. This characteristic gives us the opportunity to satisfy NGN configuration requirements in a controlled way.

REFERENCES

[1]. S. Jongtae, et al., “Overview of ITU-T NGN QoS Control,” *Communications Magazine, IEEE*, vol. 45, no. 9, 2007, pp. 116-123.

[2]. TISPAN, "NGN Functional Architecture", ETSI ES 282 001, vol. V2.0.0, Mar 2008.

[3]. 3GPP, "IP Multimedia Subsystem (IMS) Stage 2", vol. Release 8, Jan 2008.

[4]. “Broadband Forum”, http://www.broadband-forum.org/, 2009-02-24.

[5]. TISPAN, "Policy and charging control architecture - version 7.8.0", October 2008.

[6]. TISPAN, "Resource and Admission Control Sub-System (RACS): Functional Architecture", ETSI ES 282 003, vol. V2.0.0, May 2008.

[7]. TISPAN, "Network Attachment Sub-System (NASS)", ETSI ES 282 004, vol. V2.0.0, Feb 2008.

[8]. DMTF, “Web-Based Enterprise Management (WBEM) Initiative”, http://www.dmtf.org/standards/wbem/, 2008-12-14.

[9]. J. Strassner, “DEN-ng: Achieving Business-Driven Network Management,” Proc. Network Operations and Management Symposium, 2002- NOMS 2002., Florence, Italy, 2002.

[10]. K.H. Chan, et al., "RFC 3084 - COPS Usage for Policy Provisioning (COPS-PR)", T. I. E. T. F. (IETF), Mar. 2001.

[11]. D. Nicodemos, et al., “The Ponder Policy Specification Language,” Proc. Proceedings of the International Workshop on Policies for Distributed Systems and Networks, Bristol, UK, Springer-Verlag, 2001, pp. 18 - 38.

[12]. DMTF, "CIM Simplified Policy Language (CIM-SPL)", vol. Version: 1.0.0a, 2007-01-10.

[13]. DMTF, "CIM Policy Model White Paper - CIM Version 2.7", vol. 2.7.0, 2003/06/18.

[14]. DMTF, "Common Information Model (CIM) Specification - Version 2.9", 2005.

[15]. DMTF, "Specification for the Representation of CIM in XML", 02/05/2002.

[16]. DMTF, "Specification for CIM operations over HTTP version 1.1".

[17]. “OpenWBEM project”, www.openwbem.org, 2006-04-05.

[18]. “C++ CIM/WBEM Manageability Services Broker”, http://www.openpegasus.com, 2006-04-05.

[19]. “Java™ Web Based Enterprise Management”, http://wbemservices.sourceforge.net/, 2006-04-05.

[20]. DMTF, "Web Services for Management (WS-Management)", 2006-04-05.

[21]. S. Omari, et al., “Enterprise directory support for future SNMPv3 network management applications,” Proc. Global Telecommunications Conference, 1999. GLOBECOM '99, 1999, pp. 2010-2014 vol.2013.

[22]. B. Moore, "RFC 3460 - Policy Core Information Model (PCIM) Extensions", January 2003.

[23]. DMTF, "CIM Query Language Specification", p. 28, 13/08/2008.

[24]. D. Agrawal, et al., “Issues in Designing a Policy Language for Distributed Management of IT Infrastructures,” Proc. Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on, 2007, pp. 30-39.

[25]. A. Westerinen and J. Schott, “Implementation of the CIM Policy Model using PONDER,” Proc. Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on, 2004, pp. 207-210.

[26]. R. Lopes, et al., “Executable Graphics for PBNM ” Proc. 5th IEEE International Workshop on IP Operations and Management - IPOM 2005, Barcelona, Spain, Springer, 2005, pp. 108-117.