proc. of the 4th Conference on Telecommunications – ConfTele2003, Aveiro, Portugal, 18-20 June 2003.

VISUAL COMPOSITION of MANAGEMENT POLICIES

Vitor Roque[1], José Luís Oliveira[2], Rui P. Lopes[3]

[1] Polytechnic Institute of Guarda, ESTG, 6301-559 Guarda, Portugal
vitor.roque@ipg.pt
[2] University of Aveiro, DET, 3810-193 Aveiro, Portugal
jlo@det.ua.pt
[3] Polytechnic Institute of Bragança, ESTiG, 5301-854 Bragança, Portugal
rlopes@ipb.pt

Abstract

Policy based management have gained a crescent importance in the last years. New demands on internetworking, on services specification, on QoS achievement and generically on network management functionality, have driven this paradigm to a very important level. The main idea is to provide services that allow specifying management and operational rules in the same way people do business. Despite the main association of this technology with network management solutions, its generality allows to extend these principles to any business process inside an organization. In this paper we discuss the main proposals in the field, namely the IETF/DMTF model, and we present a proposal that allows the specification of policy rules through a user-friendly and component-oriented graphical interface.

I. INTRODUCTION

Network management has become in the last years a matter of great importance due the increased dependence of enterprises on their networked applications. This dependence has made the availability and performance of network services more critical than ever.

The evolution of network management has passed several stages, from management based on human-effort to proprietary management systems and finally to management systems based on open standards encouraged by standardization organizations mainly, like the International Organization for Standardization (ISO) and the Internet Engineering Task Force (IETF). Configuration management is a key area in any management solution and it affects directly other functional areas like security, performance, accounting and fault. Usually network configuration is an interactive task between the network administrator and the managed network equipments. If we consider that, due the crescent complexity of equipments and their management, new technologies, new network services and so on, the network administration occupy more and more time of user managers it is essential to find new solutions for network management.

In this context it is desirable that a network management system will be enriched with the ability to automatically manage the network configuration based upon high-level rules, more or less in the same way business-oriented requests are issued. For example, a management system should be capable, for a specific management situation, to offer facilities to reconfigure the whole system without the network administrator have to worry about the configuration details of network equipment.

Policy-Based Management (PBM) has emerged during the last years as the right paradigm to deal with this type of requirements [1]. The main idea of PBM is the definition of high level procedures – policies – that will rule the behaviour of the network regardless the intricate lower level equipment details. The main purpose of the PBM systems is the storage, management and the transformation of policies into configuration instructions that can be applied to the network equipment. Although the focus has been primary put on configuration management, all other management areas are suitable for the application of policies.

This paper reviews current models for policies specification and proposes a solution based on visual composition of management policies.

II. NETWORK MANAGEMENT AND POLICIES

Along the past years several network management models have been proposed, adopted, failed, redefined, tested, augmented (…). This rich and continuous work around the theme has been motivated by the increasing need for managing networks, systems, services and applications in an integrated and simple way. While the network complexity grows up new requirements were made to the management entities – for instance, better handling of internetworking processes to better deal with quality of service and security constraints. Although traditional management models are too tightly connected to the lower level instrumentation procedures and the construction of high level management rules have been outside normalization committees until recently. In this context new proposals have been presented such as COPS [2] and SNMP for Configuration [3], inside the IETF, and CIM [4] and PCIM [5] from the DMTF (in fact PCIM is a result from both organizations).

The activities on PBM standardization have been done mostly by two working groups of IETF: the Resource Allocation Protocol Working Group [6] and the Policy Framework Working Group [7]. The policy framework architecture is composed of four functional entities: the Policy Management Tool (Policy Console), the Policy Repository, the Policy Decision Point or Policy Server (PDP) and the Policy Enforcement Points (PEP). The model describes the key components but it does not prescribe any implementation details such as distribution, platform or language. As a consequence the Policy Console is the less defined component and it depends greatly on the functionality and design options taken by developers.

The PDP is the entity responsible for checking when and how policies can be applied. The meaning of policy in this context is very simple: it is one or more rules that describe the action(s) to be taken when specific condition(s) exist. It can be expressed semantically as:

if (policyCondition) then (policyAction)

On the other side, the PEP is the entity point where the policy decisions are enforced when the rule condition returns a true value.

The Policy Repository is the site where all policy information is stored. The information stored here describes authorized users, applications, computers and services (objects and attributes) and their relationships. The repository is also accessed in the rule validation process to detect conflicts.

A policy protocol is used to transfer policy information among PDPs and between PDPs and PEPs. As the PDPs involved in the decision process may be located in different organizations, we can differentiate between intra-organizational and inter-organizational policy transfer protocols. COPS have been used mainly for intra-organizational policy transfer, while RSVP has been proposed to be used for inter-organizational policy transfer [8].

This policy model has been also pushed by the DMTF that has been working closed with the IETF in this area. Within the CIM context, the organization has also proposed an extension schema that deals with policy modeling. The Policy Core Information Model [5], PCIM, extends the CIM with classes to represent policy information.

III. A VISUAL APPROACH FOR POLICIES DEFINITION

The Policy Framework WG defines Policy as an aggregation of policy rules [9]. Each policy rule is made up of a set of conditions and a corresponding set of actions. The policy framework architecture defined by this WG describes the key components, but it does not prescribe any implementation details such as distribution, platform or specification language.

In policy-based management, the network administrator needs a tool to define the behaviour of the system. This definition of the behaviour must be done in an independent fashion from the network equipments, and the syntax must enable the definition of a wide set of events. In this context the use of a generic specification language permits the network administrator to represent policies independently from the management system that have to enforce it. The main problem is precisely in the way how to translate behaviours or high-level policies to a generic language, due factors like comprehensibility, integration, security and heterogeneity [10].

A main goal of policy languages has been the definition of a generic and widely used language that can be used in a universal way for any management requirement. However, several dissimilar and specific languages have been developed, and are currently being exploited to represent policies in different application areas such as routing, access control and QoS. Some of the major network policy languages are currently PFDL, RPSL/SPSL, Ponder, SRL and XML [11]. The definition of a generic widely used language seems difficult to achieve due the actual technology limitations.

The Extensible Markup Language (XML) allows describing structured information by defining specific tags [12]. This tag arrangement allows building a document that can be used to exchange information independently of the platform, programming language or application objective. These characteristics make it ideal for representing policies. In fact, a pioneer work has been done by OASIS – Organization for the Advancement of Structured Information Standards [13] – that proposes the XACML, a specification to express policies in XML formalism and allow easy exchange of data over the Internet.

We have been using XML for the definition of high level management operations [14] and yet for the provisioning of management information persistence in volatiles SNMP agents [15]. Associated with this work we have created a component-based graphical interface that allows defining operations, expressions or rules using pre-defined components (Java Beans) that can be dynamically associated with the user interface.

This solution is being exploited to the definition of roles, conditions, actions, rules and policies – we are currently working on the definition of a set of base components that characterizes the way these concepts are defined.

Another achievement from this work is the usage of a unique specification language (XML based) that can be stored both in the PDP side, or policy repository, and in the PEP side, if this entity accepts directly this specification formalism [14].

The model we have built permits the user to define policies upon a unique specification language (Figure 1). The resulting information can be transferred within the system elements using a single syntax. The requests must follow the specifications defined in a standard template (DTD/Schema) where it is defined the relevant information that requests must have. After validation, the request is subsequently processed by the PDP where a decision is made, based on the request parameters (user, policy issuer, system, policy, policy destination, etc). The decision is communicated, via a transport protocol (e.g. COPS, SNMP), in response to the corresponding system entity.

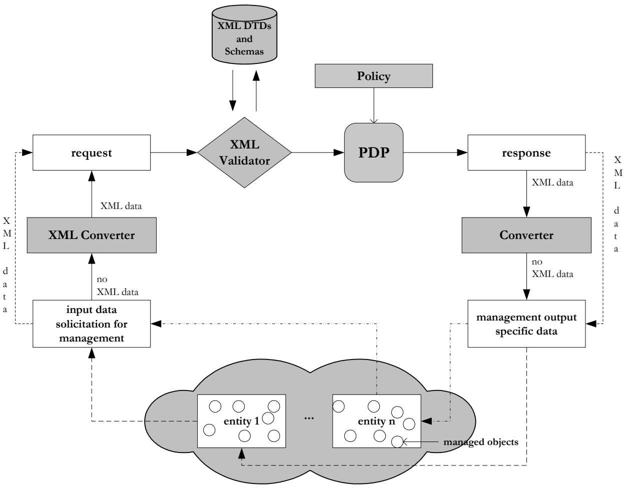

Figure 1 - XML policies system model.

With the use of XML we can handle agents to the enforcement of policies in a simpler way. Considering that data is structured and sent to the agent in XML, it will be much easier for the agent to understand exactly what the data (policy) means and how does it relates to other pieces of data it may already know (installed policies). Also the use of XML can bring more "intelligence" to the agents since we have the possibility to enable smart searches with the use of standard templates (DTDs/Schemas). With smart searches the possibility of choosing the wrong information (wrong policies) from a repository is lower than with another, unstructured, language.

IV. USAGE SCENARIO

The Policy Editor prototype permits to represent in a graphical way the definition of policies. The usage of Java and XML provides this tool with great flexibility and scalability that can be used by any kind of policies system.

The tool consists of a graphical editor with the following functions (Figure 2):

1. Definition/import of policies;
2. Interactive construction of policies by choosing and positioning the different elements in the Policy editor;
3. Conflict detection and validation;
4. Policies storage.

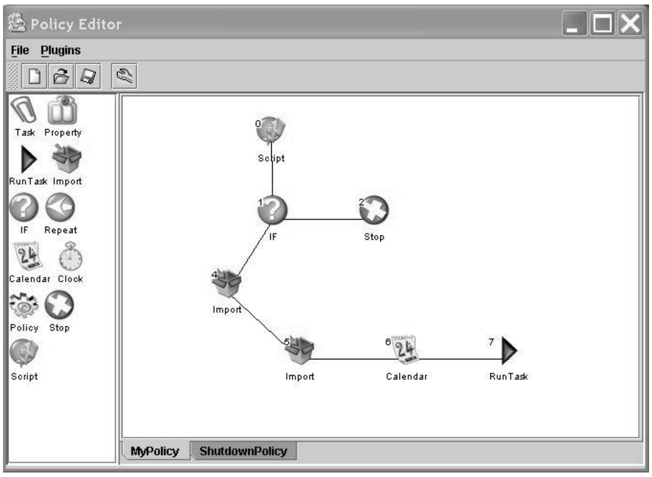

Figure 2 – Policy Editor, a composition framework for the definition of management policies.

Figure 2 also represents a simple policy construction. In this case, `MyPolicy` is the composition of a `script` element (the request in Figure 1, a validation element), the `if` that serves to verify if it is possible for the system to satisfy the request, the `import` element (the rules to be applied), and the element `calendar`, that define the policy scheduling. When executing `MyPolicy`, the system reads the XML document and performs the described operations in sequence (Figure 3).

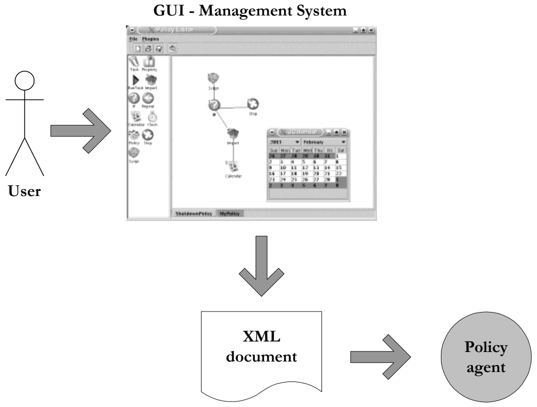

Figure 3 – MyPolicy definition.

All the editor policy editor icons are dynamic generated based on XML Schemas. We expect, in a near future, to improve considerably the prototype by adding Web interfaces. With this improvement it will be possible the access from any browser and from any location in the internet.

V. CONCLUSIONS AND FUTURE WORK

Policy-Based Management means managing or configuring network elements based upon a set of business rules or business objectives.

While traditional configuration management enables a device-by-device configuration of network elements, increased size (more devices to configure) and complexity (devices are of different types, from different vendors, with different technologies and perform far more operations) are turning the configuration into a more difficult task. The main goal of Policy-Based Management is to go beyond these difficulties.

To achieve this goal, the definition of policies is a subject of a great importance. Within this paper we made some considerations about what a policy language must implement to be usable in PBM systems. Concepts like user, policy, role, role hierarchy, permission, constraint, history and application and their interactions were discussed.

Although the general idea of using policies for managing network is powerful and appealing, policy management products and generalized implementations and usage are still far a way from consensual solutions. Improving policy languages with graphical and user-friendly interfaces can help to change this slowly evolution.

In this paper we proposed a solution for the definition of policies in a graphical oriented way. This high level semantic allows composing rules upon visual components. Moreover, the underlying policies definition provides a universal syntax, in XML, that allow easy transfers, storage and even edition using a common XML editor.

VI. REFERENCES

[1] Sloman, M. (1994), “Policy Driven Management for Distributed Systems”, *Journal of Management Information Systems* **2**(4): 333-360.

[2] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R. and Sastry, A. (2000), “The COPS (Common Open Policy Service)”, IETF, RFC2748, January.

[3] MacFaden, M., Partain, D., Saperia, J. and Tackabury, W. (2002), “Configuring Networks and Devices With SNMP”, IETF SNMPCONF Working Group, draft-ietf-snmpconf-bcp-10.txt.

[4] CIM-Core (2002), Common Information Model - Core Model v2.6, DMTF.

[5] Moore, B., Ellesson, E., Strassner, J. and Westerinen, A. (2001), "Policy Core Information Model Specification v1", IETF, RFC3060, February.

[6] Rap (2002), Resource Allocation Protocol (rap) WG, IETF, http://www.ietf.org/html.charters/rap-charter.html

[7] Policy (2002), Policy Framework (policy) WG, IETF, http://www.ietf.org/html.charters/policy-charter.html

[8] INTAP (2001), "Survey on Policy-Based Networking", Interoperability Technology Association for Information Processing (INTAP).

[9] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J. and Waldbusser, S. (2001), "Terminology for Policy-Based Management", IETF, RFC3198, November 2001.

[10] Goh, C. (1998), "Policy Management Requirements", *HP OpenView University Association (HP-OVUA) Plenary Workshop*, ENST de Bretagne, Rennes, France,

[11] Stone, G. and Xie, G. (2001), "Network Policy Languages: a survey and a new approach", *IEEE Network* **15**(1).

[12] XML (1998), "Extensible Markup Language (XML) 1.0", XML World Wide Web Consortium,, W3C REC-xml-19980210, February.

[13] OASIS (2002), "OASIS eXtensible Access Control Markup Language (XACML)", Organization for the Advancement of Structured Information Standards, Committee Specification 1.0.

[14] Lopes, R. and Oliveira, J. (2002), "A Multi-protocol architecture for SNMP entities", *IEEE Workshop on IP Operations and Management (IPOM 2002)*, Dallas, USA.

[15] Lopes, R. and Oliveira, J. (2001), "A new mechanism for distributed managers persistence", *Third International Conference on Enterprise Information Systems (ICEIS 2001)*, Setúbal, Portugal.