Policy-Based Management Overview

by:

Vladimir Tosic

Outline

- Introduction to service-level management
- Definitions and examples
  - Policies, policy rules, types of policy rules
  - Policy conflicts, meta-policies
  - Roles and related concepts
- Emerging standards
  - The IETF/DMTF Policy Framework
- Potential benefits and problems of policy-based management

Motivation for Service-Level Management

- End users want to have some **control** of IT services (and underlying systems) they use
- However, they want management reports in **business-oriented** terms they understand
- Network and system management systems provide **technical** management information
- A **mapping** between technical and business-oriented management views is necessary, but not enough

Service-Level Management

- Adopts service- and business-oriented view
- The **process** of planning, negotiating, implementing, and controlling services in order to ensure that service customers (users) get the required **service** availability, performance, quality, security, and price
- **Higher-level concepts** than in network and system management
- Reports in **business-oriented terms**

Service-Level Agreements and Service-Level Objectives

- A **service-level agreement (SLA)** is a high-level, business-oriented view of services that humans easily understand and express
- An SLA specifies (in **business-oriented terms**) the expected operational characteristics of the relationship between service customers and service providers
- **Service-level objectives (SLOs)** define metrics to enforce, police, and/or monitor SLA

SLA Example

- Duration of the service
- Description of the service
  - Service overview
  - Priority
  - Critical and peak periods
  - Impact and cost of outage
  - Availability
  - Accuracy
  - Security
  - ...
- ...

SLA Issues

- SLAs **cannot** be directly used in management because they do not contain detailed technical descriptions of management activities that have to be performed
- Definition of SLAs that can be translated into descriptions of corresponding management activities **need not** always be straightforward
- Policies are **the link** between SLAs and actual technology-oriented management activities

Policies and Policy Rules

- **Policies** define, in **technical terms**, desired states and behaviour of managed entities
- Often specified as **systems of policy rules**
- Policy rule format:

```
IF <set of conditions to be met (states
 of managed resources, events in the
 managed system, time, etc.)>
THEN <set or sequence of actions to be
 taken when the conditions are met>
```

- Various **levels of abstraction**

Policy Rule Example

- Provide high QoS for traffic to or from the AccountingSubnet during the last 10 days of the month and the first 15 days after the end of a fiscal quarter

```
IF (((IPsubnet 192.168.12.0/255.255.248.0)
     && (dayOfMonth in last10days)) ||
    ((IPsubnet 192.168.12.0/255.255.248.0)
     && (monthIn [Apr, Jul, Oct, Jan])
     && (dayOfMonth in [1-15])))
THEN priority = 6
```

Policy Framework

- A **policy framework** contains all underlying mechanisms, methods, protocols, and tools used for policy-based management activities:
  - policy rule definition and modification by users
  - policy rule storage and retrieval (usually in a **policy repository**)
  - policy rule interpretation, implementation, and enforcement
- Various suggested architectures

Types of Policy Rules

- **Positive authorization**
- **Negative authorization**
- **Obligation** (positive obligation)
- **Refrain** (negative obligation)
- **Positive delegation**
- **Negative delegation**

Types of Policy Rules - Examples

- Positive authorization: “TAs may enter marks into the marks processor and correct them.”
- Negative authorization: “Students may not enter/correct marks in the marks processor.”
- Obligation: “TAs must enter marks into the marks processor after every assignment or midterm.”
- Refrain: “TAs must not correct marks in the marks processor after the professor has corrected them.”

Policy Conflicts

- A **policy conflict** occurs when conditions in two or more policy rules are simultaneously satisfied but not all of the corresponding actions can be performed together
- **Modality policy conflicts**
  - Positive authorization / negative authorization
  - Negative authorization / obligation
  - Obligation / refrain
- **Application-specific policy conflicts**

Meta-Policies

- **Meta-policies** (policies about policies, e.g., precedence rules) are **policy constraints** used to resolve policy conflicts
- Example: “Rules for TAs always have higher precedence than rules for students.”
- Other types of policy constraints (limiting particular policy rules) are also possible

Roles

- Entities in a managed environment play some **roles** (possibly more than one at a time)
- Specification of policies for roles is much **more convenient and flexible** than specification for particular entities
- **Dynamic** change of which roles are played by particular entities can be done without changing the related policies
- **Role classes** can be used for convenient specification of particular roles

Some Related Concepts

- A **policy domain** groups managed entities for which a common policy applies
- Policy domains can **overlap** and can be **nested** (the concept of **sub-domains**)
- A role defines a policy domain
- **Policy templates**
- **Role relationships**
- **Management structures**
- **Policy groups**

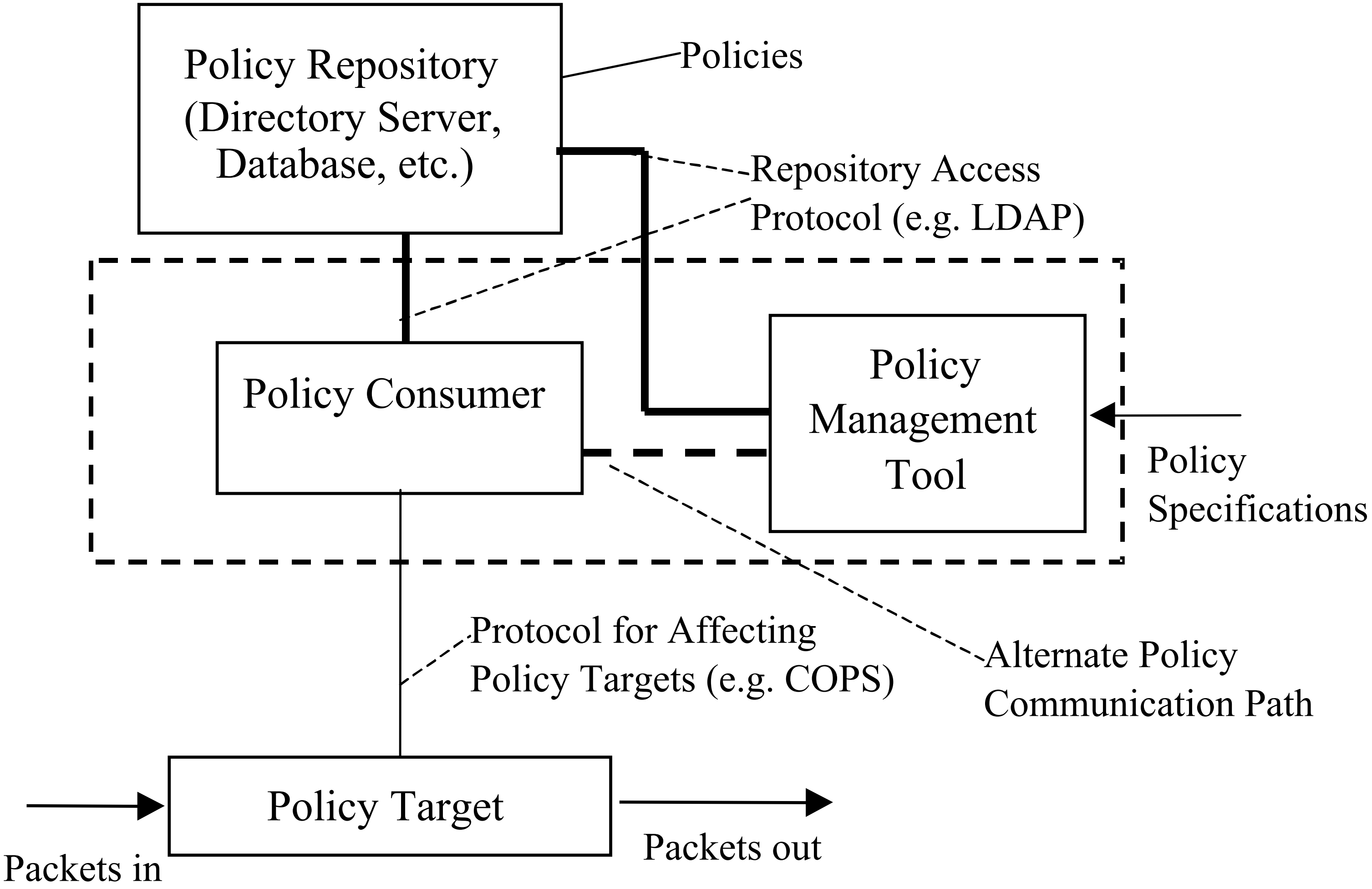
Policy-Based Management Standardization Efforts

- **Directory Enabled Network (DEN)** -integrated into the DMTF Common Information Model (CIM) standard
- **Common Open Policy Service (COPS)** protocol by the IETF Resource Allocation Protocol Working Group
- The joint **IETF/DMTF** work on the **Policy Framework** and the **Policy Core Information Model (PCIM)**

Logical Architecture of the IETF/DMTF Policy Framework

- Four main logical (functional) elements:
  - **Policy management tool** – policy rule definition and update, translation, validation for mutual consistency and global conflicts
  - **Policy repository** – storage, search, and retrieval of policy rules
  - **Policy consumer** – acquires and deploys policy rules, and optionally translates them into a form usable by policy targets
  - **Policy target** - operates as specified by a policy rule, carries out policy actions

Application to Network Management


Policy Repository (Directory Server, Database, etc.)
Policies
Repository Access Protocol (e.g. LDAP)
Policy Consumer
Policy Management Tool
Policy Specifications
Protocol for Affecting Policy Targets (e.g. COPS)
Alternate Policy Communication Path
Policy Target
Packets in
Packets out

Policy Decision vs.
Policy Enforcement

- **Policy decision** is the process of evaluating conditions in policy rules. It may occur in a **policy consumer**, in a **policy target**, or in **both**
- **Policy enforcement** is the process of executing the appropriate (device-specific) policy rule actions that are determined according to the previous policy decision. It occurs in one or more **policy targets**

Global Conflict Detection vs. Local Conflict Detection

- **Global conflict detection** is done in **policy management tools**. It checks whether a new policy rule statically conflicts with policy rules that are already in the policy repository
- Time-based and dynamic conflicts **cannot** be discovered with global conflict detection
- **Local conflict detection** is done in **policy consumers** (in some cases partially in **policy targets**). It checks for policy conflicts that apply to controlled policy targets

The Policy Core Information Model (PCIM) - I

- **Declarative** information model (does not address execution of policy actions) that will be part of the **standard CIM schemas**
  - Policy conditions and actions are modeled with **separate objects** containing **opaque byte arrays in an arbitrary encoding**
  - Policy conditions and actions can be defined in the scope of a **single policy rule** or in the scope of the **policy repository** (in the latter case, they can be reused across many policy rules)

The Policy Core Information Model (PCIM) - II

  - Policy rules are not associated with the policy repository, but can be organized into hierarchies of **policy groups**
  - **Only** policy conditions (not policy actions) are associated with **managed entities**
  - Addresses a number of issues (e.g., ordering of policy actions, precedence of policy rules, policy constraint composition, roles, ...) through new standard CIM **classes** and their **data members**
- Flexible, but too complex and unconstrained with the possibility of significant problems

Policy-Based Management - Potential Benefits

- **Better** distribution of management control (resulting in potentially improved management efficiency, robustness, and scalability)
- Enables **dynamic** deployment of management functionality
- **Might reduce** interoperability and platform-dependence problems
- There are some **emerging standards**

Policy-Based Management -Potential Problems

- Policy refinement and policy conflict detection/resolution might be **bottlenecks**
- Might be too **static** and **centralized** for dynamic (e.g., active) and autonomous self-configuring systems
- **Not yet mature** technology, drastic differences in some adopted solutions (also applies to emerging standards)
- **Performance** issues have yet to be explored

Conclusions

- Service-level management is a **necessity**
- Policies are **the link** between abstract business-oriented SLAs and actual technology-oriented management activities
- There are some **emerging standards** like the IETF/DMTF Policy Framework and PCIM
- There are still a number of **technical issues** related to policy-based management that have yet to be solved