



Policy Specification and Implementation for Security Management of Distributed Systems

Nicodemos C. Damianou

Supervisor: Professor Morris Sloman



29 February 2000

Presentation Overview

What is my Research About?

The Scope

What are the Goals?

The Research Issues

How will I Achieve the Goals?

The Past

The Present

The Future

Part I

What is my Research About?

The Scope

Research Overview

⇒ Management of Distributed Systems

Policy-based Management (PBM)

- Applications to Security Management

Many Approaches to Management

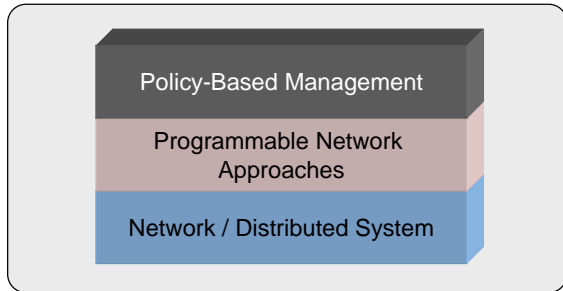
Dynamically Programming Network Elements

- Management by Delegation
- Dynamically Programmable (Mobile) Agents
- Active Networks

No single solution

PBM complementary to other approaches

PBM Vs Other Approaches!



◆ Two Interpretations of the Figure

PBM uses other approaches as mechanisms to manage the underlying Network
 PBM is used in addition to other approaches

◆ PBM more appropriate for Security Management

Intuitive specification of Access Control and Security Management functions
 Who and when is authorised to Program the Network

Security Management

What is Security Management?

Provision & Support for the Specification of Access Control Policies
 Logging, Monitoring and Auditing
 Abstracted from the Security Mechanisms
 A Distributed function by nature

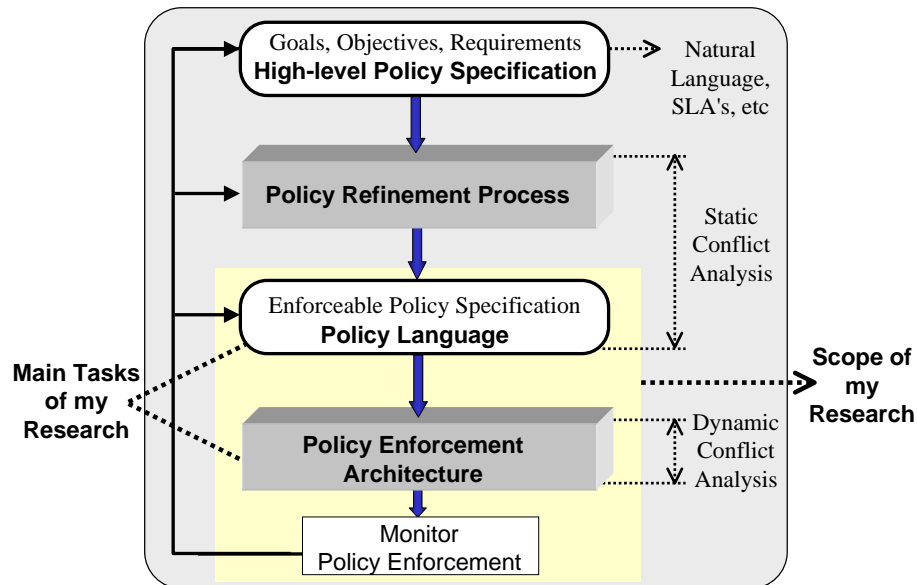
How does Policy-based Management apply to it?

Provides a way to specify those requirements

- Authorisation Policies - *for Access Control*
- Obligation Policies - *for Logging, Monitoring, Auditing*
- Structures to group policy specification
- Means to Analyse policy

And Enforce them in a distributed fashion

Policy Framework



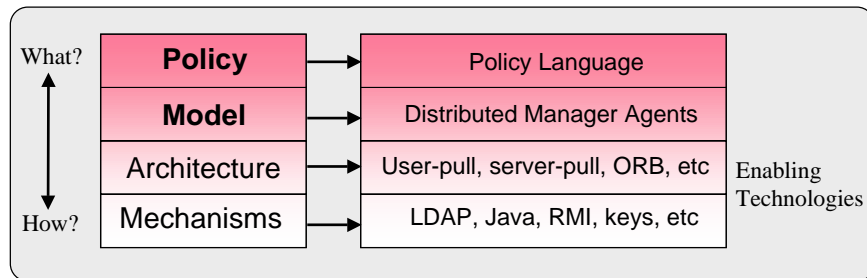
Part II

What are the Goals?
 The Research Issues

The Goals

Engineering a Security Management Framework

Ravi Sandhu



9

The Goals (2)

⇒ Policy-based Security Management Framework

Policy Specification

Policy Language

Policy Enforcement Architecture

Enforcing Obligations

Access Control Mechanisms

Evaluate with case studies

Framework also applicable for General Management

- Qos
- Configuration
- Storage

10

Policy Specification

Why do we need a Policy Language?

The Link between high-level goals and their enforcement

Why not a general purpose Language (C, Java etc)

- Experience programmers are needed
- Security concerns
- Impossible to Analyse Policies for conflicts
- More difficult to capture organisational policies in Java

Tool Support

Policy Compiler (multiple back-ends)

Policy Specification Toolkit

11

Policy Language - Design Goals

Extensibility

- New types of policy might arise

Analysability

- Conflict detection is a major aspect of policy specification

Expressiveness

- To be understandable not only by programmers

Platform Independence

- Multi-Platform

Declarative Semantics

- Simplicity. Makes conflict detection feasible

Interpretation

- Dynamically loading the policies into the agents to interpret them

12

Policy Language (2)

Builds on the old Policy Notation

Two main Policy Types

Authorisations - Specifying Access Control

- Delegation (Essential for DAC Systems)

Obligations - Specifying actions that managers must perform

- Refrains - Actions managers must not perform

Subjects / Targets explicitly identified

Domains are used to group Objects

Roles used to group policies related to positions in an organisation

⇒ **Result: Ponder (More later)**

13

Policy Language (3)

Obligation Policies

What the Managers must do

```
oblig rlogin1 { subject AC_agent; on rlogin_event;
do enable_encryption();
target */apps/transfer_protocols;
when time.between(0900,1700) }
```

Event Driven Rules

Actions are: Methods on targets, Manager Actions

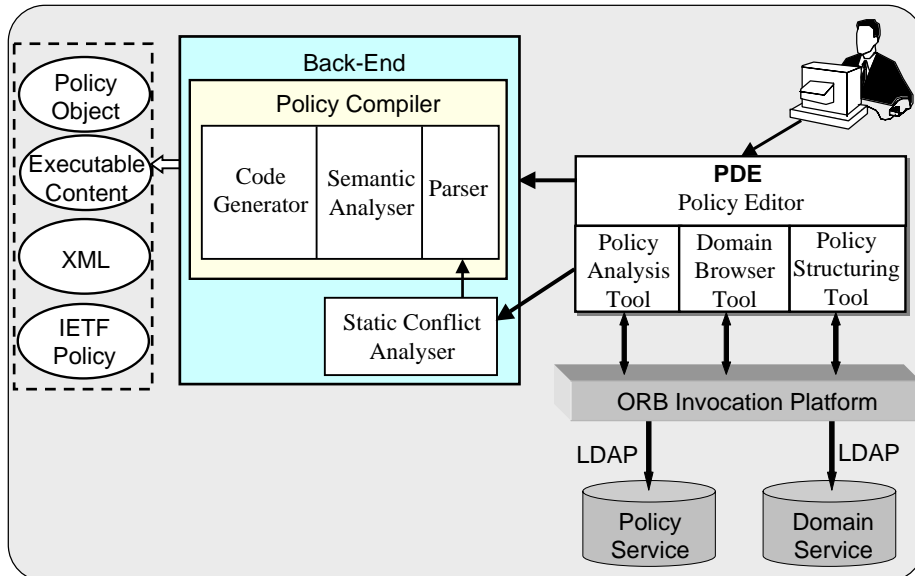
Authorisation Policies

What subjects are allowed to do

```
auth+ serviceMan { subject brServiceMan;
action resetSchedule, enable, disable
target brServices; }
```

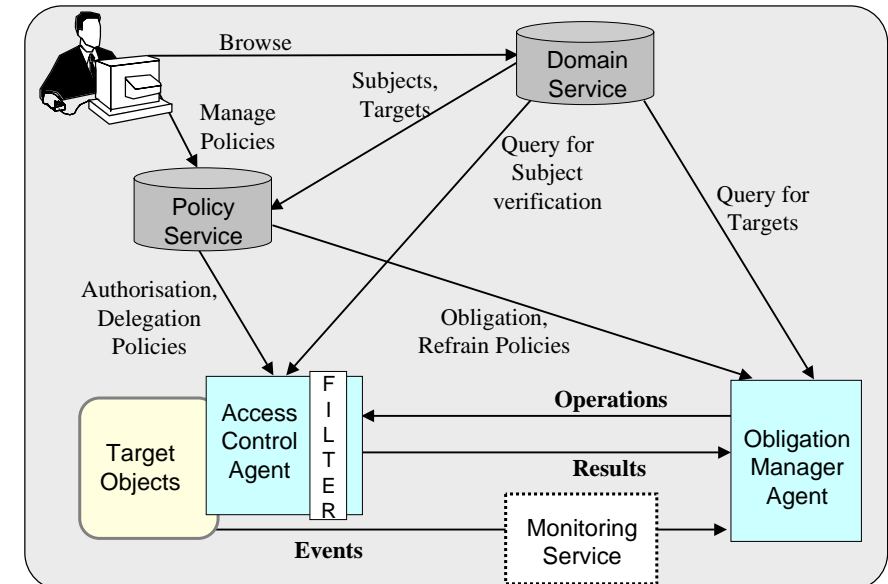
14

Policy Toolkit



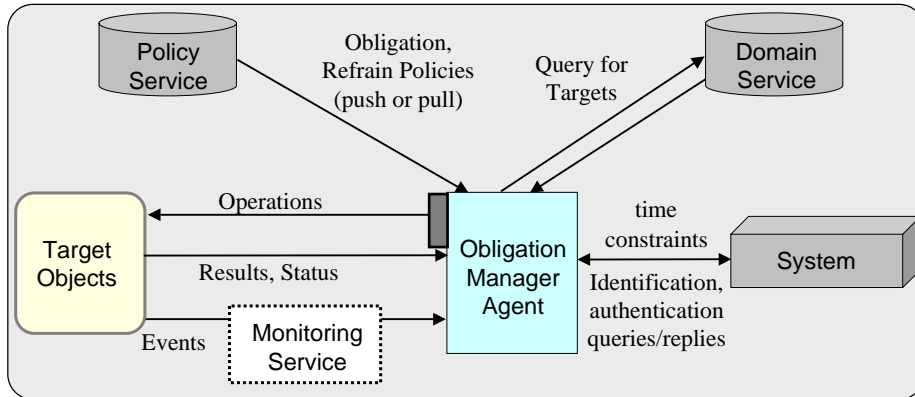
15

Policy Architecture



16

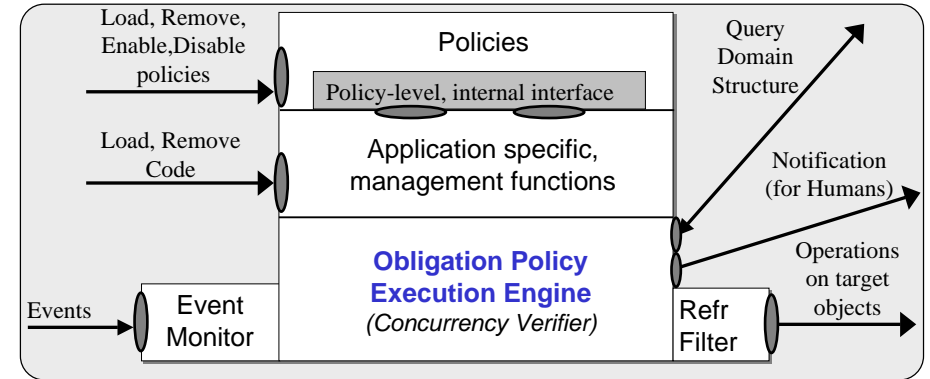
Obligation Enforcement



- ◆ Evaluate Constraints
- ◆ Query Domain Structure
- ◆ Handle Events
- ◆ Execute (Interpret) Policies
- ◆ Implement Refrains (Filters?)
- ◆ Human Managers?

17

Obligation Manager Agent

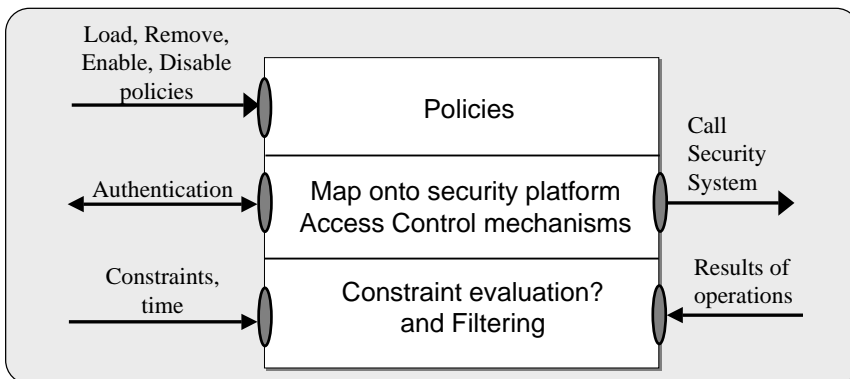


Multi-threaded

- To Manage Multiple Policies
- To Execute Multiple Policies (multiple events)

18

Access Control Enforcement



How can we map constraints?

Problem in Windows NT

How do we perform Filtering?

19

Part III

How will I achieve the Goals?

The Past, The Present, The Future

20

The Past

Policy Language: **Ponder**

New Language

- More Complete
- Conforms to the Design Goals

Grammar for the Language

- Language Specification Document

Compiler

Syntax Analyser

Ponder

Object-Oriented

Types, Instances,
Inheritance

Declarative Language

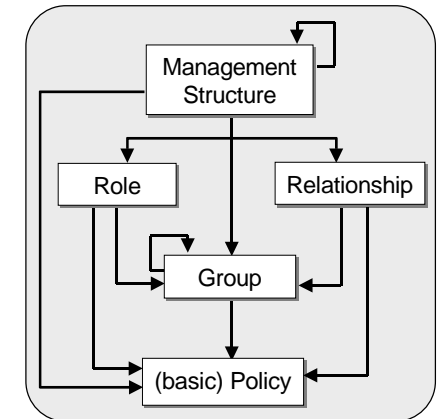
Policy Support for

Authorisations with
Information Filtering
Obligations and Refrains
Delegation policies

Meta-Policies

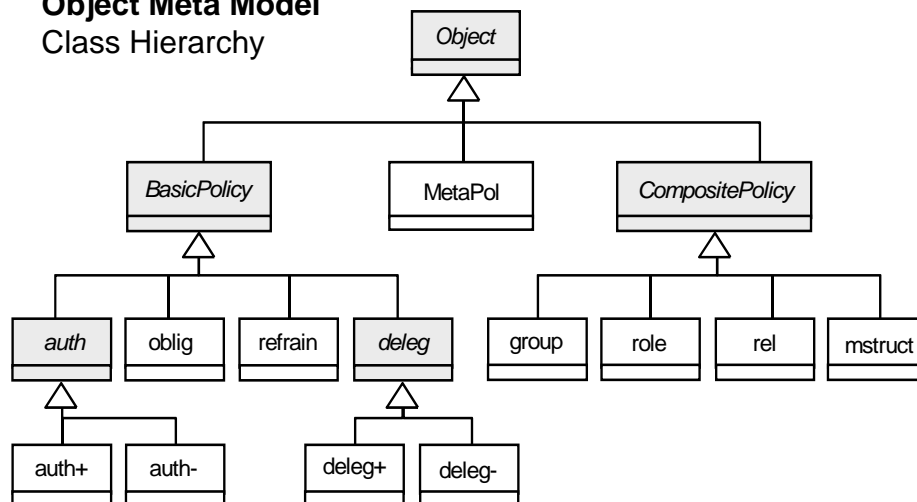
Object Constraint
Language

Structured Specification of Policy



Ponder (2)

Object Meta Model
Class Hierarchy



Ponder (3)

Types and Instances

```
type
  oblig allocBwT(subject m, target o) {
    on perfDegradation(bw, source)
    do bwReserve(bw+10)
  } // allocBwT

inst
  oblig site1/perf = allocBwT(site1/netOp, site1/edgeRtr)
  oblig site2/perf = allocBwT(site2/netOp, site2/edgeRtr)
  oblig allocBW {
    subject netOp; target edgeRtr
    on perfDegradation(bw, source)
    do bwReserve(bw+10)
  } // allocBW
```

Ponder (4)

Structuring: Roles - Relationships

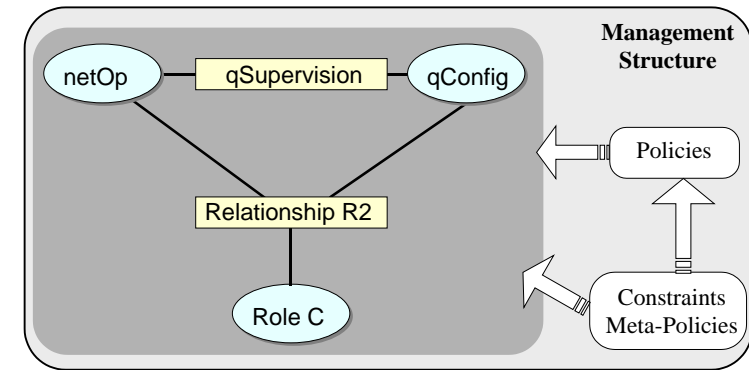
```
import /net/oam/*
inst rel qSupervision {
  role netOperator          /* from /net/oam */
  role /net/edge/qConfig

  inst
  oblig report {
    subject /net/edge/qConfig.subject
    on time.at(1800); do report(q_info)
    target netOperator
  } // report
  auth+ config {
    subject /net/oam/netOperator.subject
    action setStrategy; target qConfig
  } // config
} // qSupervision
```

25

Ponder (5)

Management Structures



⇒ <http://www-dse.doc.ic.ac.uk/policies/ponder.html>

26

The Present

Policy Compiler

Semantics Analysis

- Consistency Analysis
- Scope Analysis
- Type Analysis

Policy Toolkit

Policy Editor

- With Syntax Highlighting

27

The Future

Policy Language

Formal Semantics (Operational Semantics)

Policy Compiler - Code Generation

Policy Toolkit

Graphical Notation

Run-time API - Dynamically change structures

Policy Enforcement Architecture

Policy Service

Domain Service

Obligation Policy Agent

Run-time API to control Obligation Agents + Tool

Evaluate the Framework: Case Studies

28