Infrastructure-aware Autonomic Manager for Change Management

H. Abdel Salam
R. Mukkamala

K. Maly
M. Zubair

Department of Computer Science,
Old Dominion University
IT Management Complexity

Managing IT infrastructure for a large organization consisting of tens of networks that connect hundreds of servers is a challenging task.

Using ad-hoc and human-based methodologies to manage large size organizations can be costly, error prone and crisis oriented rather than targeted and predictable.

Different standards and best practices such as ITIL (IT Infrastructure Library) and COBIT (Control Objectives for Information and related Technology) have been proposed to describe the procedures that should be followed while managing IT environments.

In this paper, we focus on the Change Management component of IT management. ITIL defines Change Management as the standard procedures that ensure that changes to configuration items in a controlled IT infrastructure are carried out in a planned and authorized manner.
Change Management Workflow

Figure 1. A Sample change management workflow
Traditional Policy Engine Architecture

Properties (Sensors):
- Property-1 (Read-Only)
- Property-2 (Read-Write)

Getter-Setter Methods:
- Property-1.Get()
- Property-2.Set(value)

Action Methods:
- Action-Handler-1()
- Action-Handler-2()

If (Condition-1) Then Decision-1
If (Condition-2) Then Decision-2
...
If (Condition-k) Then Action-1
If (Condition-k+1) Then Action-2
...

Conditions can contain Infrastructure independent operators & vocabularies only.

Client (Managed Resource)

Infrastructure information is limited to what the client offers to the engine through its properties or methods.

Underlying Infrastructure

Policy Engine

Policy Deployer

Solicit Decision or Action

Decision

Action

Policy Evaluator

Policy1.xml
Policy2.xml

June 13-15, 2007
Policy 2007
Infrastructure Aware Policy Based System

- Infrastructure-aware Policy Engine

- Policy language constructs to represent dependencies among applications, hardware and software

- XML representation of Infrastructure
Infrastructure Aware Policy Engine

Client (Managed Resource)

Properties (Sensors):
- Property-1 (Read-Only)
- Property-2 (Read-Write)

Getter-Setter Methods:
- Property-1.Get()
- Property-2.Set(value)

Action Methods:
- Action-Handler-1()
- Action-Handler-2()

Solicit Decision or Action

Policy Engine

If (Condition-1) Then Decision-1
If (Condition-2) Then Decision-2
If (Condition-k) Then Action-1
If (Condition-k+1) Then Action-2

Policy Evaluator

Infrastructure Knowledge Base
Relevant Information about underlying infrastructure.
Information can be available in different formats (Files, XML, Database, etc)

Policy1.xml
Policy2.xml

The policy language is able to support infrastructure dependent operators (e.g., IsAvailable, IsConnected)
Policy examples

• The Blackboard application (for example) may be inaccessible only during Saturday evening from 10 PM to 12 PM; however urgent security patches may be installed as long as the application is accessible from at least 2 labs during work hours.

• No server may be taken down on three consecutive weekends.
Infrastructure Aware Operators for Complex Dependencies

Dependency between applications and hardware in a large IT infrastructure can be complex.

- OR Dependencies: Using of Database replicas to improve service availability.

- Access Type Dependencies: Many Applications offer different features based on the user privileges; these features have different dependency requirements.
Access Type Dependencies

Server S1 runs the “Blackboard” application which for “Faculty” members requires access to the database DB1 and the Apache web server.

However, when accessed by “Students”, Blackboard requires only the Apache web server.

To represent this dependencies, we introduced the “AccessType” element.

June 13-15, 2007

Policy 2007
Sample Policy

- `<odupl:Condition>`
  - `<odupl:Or>`
    - `<odupl:And>`
      - `<odupl:AtLeastN N="1">`
        `<odupl:IsAvailable Resource="Blackboard" AccessPoint="Faculty" AccessLevel="1" />`
        `<odupl:IsAvailable Resource="Banner" AccessPoint="Faculty" AccessLevel="1" />`
      </odupl:AtLeastN>
    - `<odupl:Equal>`
      `<odupl:Parameter ParameterName="Severity" DataType="Integer" />`
      `<odupl:IntegerConstant>1.0</odupl:IntegerConstant>`
    </odupl:Equal>
  </odupl:And>
- `<odupl:Greater>`
  `<odupl:Parameter ParameterName="Severity" DataType="Integer" />`
  `<odupl:IntegerConstant>1.0</odupl:IntegerConstant>`
</odupl:Greater>
</odupl:Or>
</odupl:Condition>`
Sample IT Management Policy

- <odu:Condition>
  - <odu:Or>
    - <odu:And>
      - <odu:AtLeastN N="1">
        <odu:IsAvailable Resource="Blackboard" AccessPoint="Faculty" AccessLevel="1" />
        <odu:IsAvailable Resource="Banner" AccessPoint="Faculty" AccessLevel="1" />
      </odu:AtLeastN>
    </odu:And>
    - <odu:Equal>
      <odu:Parameter ParameterName="Severity" DataType="Integer" />
      <odu:IntegerConstant>1.0</odu:IntegerConstant>
    </odu:Equal>
  </odu:Or>
  - <odu:Greater>
    <odu:Parameter ParameterName="Severity" DataType="Integer" />
    <odu:IntegerConstant>1.0</odu:IntegerConstant>
  </odu:Greater>
</odu:Condition>
Sample Infrastructure Representation

Objects File

```xml
<Component Name="DLIBRouter" ObjectType="Router" Type="Router" id="1001" />
<Component Name="MyPC" ObjectType="GenericComputerSystem" Type="Machine" id="18365" />
<Component Name="128.82.7.11" ObjectType="IpDevice" Type="Machine" id="13415" />
<Component Name="taddm-client3.seven.research.edu.edu" ObjectType="IpDevice" Type="Machine" id="13449" />
<Component Name="D-Lib Main Switch" ObjectType="Bridge" Type="Router" id="13463" />
```

Connectivity File

```xml
- <Node Name="DLIBRouter" ObjectType="Router" Type="Router" id="1001">
  <Link ToNode="13463" />
  <Link ToNode="1002" />
  <Link ToNode="1003" />
</Node>
- <Node Name="BBServer" Type="Machine" id="1002">
  <Link ToNode="1001" />
  <Link ToNode="1003" />
</Node>
```

Dependency File

```xml
- <DependencyGraph>
  - <Node id="2032" Name="Banner" Type="Application" RunsOn="1011">
    <DependsOn Node="2012" AccessType="Faculty" />
    <DependencyGroup Min="1">
      <Option Node="2021" AccessType="Faculty" />
      <Option Node="2022" AccessType="Faculty" />
    </DependencyGroup>
  </Node>
  - <Node id="2021" Name="Apache-c10" Type="Application" RunsOn="12281">
    <DependsOn Node="2011" AccessType="Faculty" />
  </Node>
  <Node id="2004" Name="Oracle-nasa" Type="Application" RunsOn="10584" />
</DependencyGraph>
```
Conclusion

• Current policy languages do not enable managers to express policies that involve the connectivity and dependencies within the infrastructure managed by an IT organization. We have implemented a language and a policy engine that will allow Infrastructure-Aware rules.

• We have demonstrated the feasibility of the engine and have tested the engine with an infrastructure that is being discovered by an automated tool.

• Future Work
  – Need to integrate the engine with a workflow system that represents change management.
  – Test the scalability of the engine with regard to number of servers, switches, PCs in the infrastructure as the algorithms to implement connectivity predicates are polynomial in nature.