Using Policy-Based Concepts to Provide Service Oriented Accounting Management

Igor Radisic

Department of Informatics, University of Munich
Email: radisic@informatik.uni-muenchen.de
Motivation

- **Scenario:** outsourcing of IT services
  - Customized services (Intra-/Extranets, WWW/Email Hosting,…) and management of services incl. accounting management
  - Usually long term contracts (3-5 years)
  - Many change management operations during service operation

- **Today:** accounting realized by combining heterogeneous components, but no standard interfaces regarding:
  - Management functionality \(\Rightarrow\) no integrated, unified management view
  - Usage functionality \(\Rightarrow\) implementation of gateways between components

- **Problem:** static, hardly adaptable realization of accounting management
  - Change activities mainly carried out by human interactions
  - Adjustments result often in reimplementing parts of the customized accounting management
Requirements

• Support of a customized service accounting (tariffing, billing, etc.)
• Seamless integration of various components needed for accounting of services
• Support of change management activities
• One integrated view on accounting management
• Automation of management tasks as far as possible

Main Goal: supporting every activity involved in service accounting as far as possible
Approach

• **Process oriented management** of service accounting instead of component oriented management
  – Investigation of dynamic aspects in service accounting
  ➔ Development of
    • A **process model** using service life cycle to identify relevant sub processes
    • A **service model** containing relevant accounting entities

based on
  • Existing specifications by ITIL, TOM, OSI, ...
  • In-depth analysis of specific scenarios

• **Using policy-based concepts** to manage accounting process
  – **Policies** for managing single sub processes and process transitions
  – **Meta-Policies** to increase additionally automation of management tasks
  ➔ Development of a **Policy Description Language** and a **Policy Management System** based on existing approaches
Accounting Process Model

Used for:

- Identifying range of needed management actions
- Analyzing effects of change management activities
Relevant Accounting Entities

- Identified during analyzing activities within sub processes

- Combined to accounting service model
  ➔ Adopting prior specified entities by standardization organizations where possible

- Reveals additional dimension of associations between management relevant entities:
  - Aggregation, containment associations
  - Relations between roles
Excerpt of Accounting Service Model

- <<role>> User (client) uses <<role>> User (customer) to conclude the service agreement.
- The service agreement specifies the accountable unit, which is transferred to the invoice recipient.
- The provider domain includes a <<role>> Provider who designs and directs the service template and sends the tariff to the accounting manager.
- The accounting manager analyzes and compiles the data.
- The customer domain includes <<role>> Customer who receives the report and invoice.
- The provider side includes <<role>> Provider who designs and directs the service template and sends the tariff to the customer.
- The customer side includes <<role>> Customer who receives the report and invoice.

Igor Radisic
Managing the Accounting Process

• Process oriented view by applying policies
  + Integrated view on managed entities
  + Automation of management activities

• Introducing **mediation layer** representing complete accounting process
  ⇒ Management independent of realizing components
  + No need for respecifying existing policies in case of changing/adding components to accounting system
  + Proxy entity can implement lacking functionality of “real” components

• Overall: Replacing today's patchy implementations of
  – Gateways with mediation layer
  – Management scripts with policy specification
Specifying Policies for Accounting Management

- Existing grammars for policy description languages adopted:

  \[
  \text{Policy} \quad \text{ID} \quad \text{For Subject+} \quad \text{On Target+} \quad \text{On Event Event+} \\
  \text{Do Action+} \quad \text{Constraint ConstraintExpression+}
  \]

- Policies: managing sub processes and transitions

- Meta-Policies: supporting change activities regarding respecification of (existing) policies

- Previously performed analysis needed for specifying policy fields:

<table>
<thead>
<tr>
<th>Policy</th>
<th>Meta-Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject</strong></td>
<td>Role (set) or proxy agent</td>
</tr>
<tr>
<td><strong>Target</strong></td>
<td>Role (set) or proxy agent</td>
</tr>
<tr>
<td><strong>Event</strong></td>
<td>Start/End of activity processing</td>
</tr>
<tr>
<td><strong>Action</strong></td>
<td>Operation called on CORBA interface of proxy agent (default: target), runtime or development environment</td>
</tr>
<tr>
<td><strong>ConstrExpr</strong></td>
<td>Boolean Expression evaluated prior executing specified actions</td>
</tr>
</tbody>
</table>

Igor Radisic
Conclusion

• Need for support of dynamic aspects in accounting management
  – Development of a process-oriented, policy-based management solution
  – Prototypical implementation:
    • Mediation agents for NeTraMet, Apache Log Files, CPU Metering, etc.
    • DTD of policy description language currently ported to XML schema
    • Installation of test bed on cooperation partner’s site till end of year

• Current Work:
  – Investigation of scenario-independent reusability of (meta-)policies
    • Many ways to specify a policy with identical semantics
    • Goal: Development of a methodology or policy patterns
  – Development of a tool-based support for pricing on basis of a knowledge-based system