Specifying Reusable Concerns using DSMLs and Connector Interfaces

Aram Hovsepyan, Stefan Van Baelen, Yolande Berbers, Wouter Joosen

5/26/2008 INRIA, Nantes
Outline

- Introduction
  - problem statement
  - goals

- Generic framework for combining GPML & DSMLs
  - Instantiation for access control in XACML

- Conclusions & Future work
## Introduction: GPML vs. DSML

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPML (e.g. UML)</strong></td>
<td><strong>DSML (e.g. XACML)</strong></td>
</tr>
<tr>
<td>• standard</td>
<td>• high-level of abstraction</td>
</tr>
<tr>
<td>• generic</td>
<td>• expressive</td>
</tr>
<tr>
<td></td>
<td>• problem/implementation gap</td>
</tr>
<tr>
<td></td>
<td>• analysis, V&amp;V possible</td>
</tr>
<tr>
<td></td>
<td>• larger number of potential users</td>
</tr>
<tr>
<td></td>
<td>• development is complex</td>
</tr>
<tr>
<td></td>
<td>• standardization &amp; learning</td>
</tr>
<tr>
<td></td>
<td>• tool support</td>
</tr>
</tbody>
</table>

- GPML: • not easy to implement domain-specific abstractions
  - V&V
  - semantics
- DSML: • development is complex
  - standardization & learning
  - tool support
Introduction
AOM: Current Approaches

- Base Concern (UML)
- Access Control (UML)
- Distribution (UML)
- Real-Time Concerns (UML)

Composition Engine

Combined View (UML)
Introduction: goals

- **AOM framework that can combine different DSMLs and a GPML**
  - GPML: base concern, design patterns, etc.
  - DSMLs: access control, real-time concerns, etc.

- **Reusable concerns**
  - Composition obliviousness
Outline

- Introduction
  - problem statement
  - goals

- Generic framework for combining GPML & DSMLs
  - Instantiation for access control in XACML

- Conclusions & Future work
Generic framework

- **Base concern - GPML**
- **Aspect concern - DSML**
- **Concern interface - GPML**
  - “lingua franca” for aspect concepts in the base
- **Composition model**
  - depends on the concern type

---

5/26/2008

INRIA, Nantes
Instantiation for Access Control Case Study

- **Screening Lab Application (UML)**
- **Integrated Care Pathway Application (UML)**
- **Access Control (XACML)**
  - same for both applications
Integrated Care Pathway Application

ICP
- closedICP : boolean
- creationTime : date
+ getClosedICP() : boolean
+ setClosedICP( v : boolean )
+ close()
+ validate()

Personnel
- name : String
- department : String

Physician
- licenseID : String

Technician

StepTemplate
+ perform( icp : ICP, c : Patient )

Step
+ planNextStep()

DecisionStep
+ execute( p : Personnel )

ActionStep
+ getStepAt( index : int ) : Step

ComposedStep

5/26/2008 INRIA, Nantes
Screening Lab Application

Radiographer

Screening
- date : date
- image
+ getDate() : date
+ screen()
+ setDate(d : date)

Reading
- outcome : int
- results : String

Radiologist
performs

Conclusion
- report : String

Doctor
- license : String

Employee
- salary : double
- name : String

GeneralPractitioner

ScreeningSubject
- sisID : String
- name : String

Secretary
Representative Health Care Requirements

- **Contact**

- **Medical logical unit, e.g., patient visit, hospitalization, surgery, etc**

- **Each contact has an assigned physician**

- **A contact is closed by the supervisor of the assigned physician**
Representative Health Care Requirements

**Policies**

- Physicians assigned to a contact can view/modify it as long as 30 days after the contact is closed.

- General practitioner can view all contact information.

- A physician can overrule an access denial, provided that a detailed reason is specified.
Representative Health Care Requirements

- GP may view his own patient’s records

```xml
<Rule Effect="Permit" RuleId="urn:e-hip:ruleid:1">
  <Target><Subjects><Subject>
    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <SubjectAttributeDesignator AttributeId="urn:e-hip:access-subject:attributes.role"/>
      <AttributeValue>general practitioner</AttributeValue>
    </SubjectMatch></Subject></Subjects>
  <Resources><Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <ResourceAttributeDesignator AttributeId="urn:e-hip:resource:target-namespace"/>
      <AttributeValue>ehipmedrecord</AttributeValue>
    </ResourceMatch></Resource></Resources>
  <Actions><Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
      <AttributeValue>view</AttributeValue>
    </ActionMatch></Action></Actions></Target>
</Rule>
```
Concern Composition Models

- Map application objects to security objects & subjects
  - <<map_object>>, <<map_subject>>

- Map operations to security actions
  - <<map_action>>

- Specify security attribute calculation
  - <<map_attribute>>
  - Map security attributes to application attributes
  - Map security attributes to application methods
Composition Model

Conclusion
- report : String

Screening
-date : date
-image
+ getDate() : date
+ screen()
+ setDate(d : date)

Reading
-outcome : int
-results : String

MedicalData
- patientID : String
- closed : boolean = false
- closingTime : date
- gp : String
- responsiblePhysician : String
+ view()
+ append()
+ close()
+ validate()

Doctor
-license : String

licenseID : String
Demo

- Demo case study
Outline

- Introduction
  - problem statement
  - goals

- Generic framework for combining GPML & DSMLs
  - Instantiation for access control in XACML

- Conclusions & Future work
Conclusions

- Generic AOM framework that enables the combination for concerns expressed in GPML & DSML
  - Instantiation on access control

- Challenges
  - Scalability towards other concerns

- Reusable concerns
  - XACML policy is reusable as it is

- Proof-of-concept implementation
Future work

- Investigate scalability
- Investigate UML profiles as a substitution for DSMLs
- Implement model composition instead of code generation engine