Ontology-Based Policy Specification and Management

W. Nejdl, D. Olmedilla  L3S Research Center
M. Winslett, C. Zhang  University of Illinois

2nd European Semantic Web Conference
Heraklion, Greece, 1st Jun. 2005
Outline

- Motivation
  - Trust Negotiation
  - Policy Specification & Management
- Using Ontologies for Policy Specification
- Protégé Plug-in
- Conclusions
- Further Work
Traditional Access Control for Decentralized Systems

Assumption: I already know you---you have a local account!
Trust Negotiation

- Trust is based on parties’ properties
- Every party can define access control policies to control outsiders’ access to their sensitive resources
- Establish trust iteratively and bilaterally by the disclosure of certificates and by requests for certificates
**Step 1:** Alice requests a service from Bob

**Step 2:** Bob discloses his policy for the service

**Step 3:** Alice discloses her policy for VISA

**Step 4:** Bob discloses his BBB credential

**Step 5:** Alice discloses her VISA card credential

**Step 6:** Bob grants access to the service
Protection of Credentials
Problems

- Management of policies is difficult
  - Many resources to protect
  - Big administration effort

- Correctness of our policies
  - Real-world policies are complex
  - Errors might provide inappropriate access
Sharing Policies for Common Attributes (I)
Sharing Policies for Common Attributes (& II)

Similar to method inheritance in object oriented programming languages

Ease policy management
- Reduce administration effort
- Reduce number of errors

But what happen if
- there are exceptions?
- policies must be refined at lower levels?
Composing and Overriding Policies (I)

Mandatory

P1 = {Signed by authorized authority}

P2 = {Non-commercial}

P3 = {Category not F}

P4 = {4 years experience}

Default

P5 = {3 years experience}

Illinois DL = \{P1, P2, P3\}

P2, P3, P5

Texas driver = \{P1, P2, P4\}
Composing and Overriding Policies (& II)

- Mandatory policies cannot be overridden
  - They are enforced at lower levels

- Default policies can be overridden
  - In order to deal with exceptions

- In any case, policies can be refined with extra constraints at lower levels
Conclusions

■ Ontologies can be used to
  ▪ Make easy management of policies
  ▪ Reduce errors at the specification

Composition and overriding of policies
  • Mandatory and default policies

■ Prototype in Protégé
Further Work (I):
Avoiding Information Leakage Request
Further Work (& II)

- Multiple inheritance
- Conflicting policies in ontology hierarchies
- Disjunction on policy composition?
- Integration with PeerTrust language
References

- Protégé Plug-in
  
  http://www.l3s.de/~olmedilla/projects/trust/trust.html

- PeerTrust Project
  
  http://www.l3s.de/peertrust/

- REWERSE NoE
  
  http://www.rewerse.net/
Thanks

Questions?