PPSWR 2005
Dagstuhl Seminar

REWERENCE WG 12
Policy Language, Enforcement, Composition

Hannover, Linköping, Naples, St. Gallen, Turin, Zurich

September 15th, 2005
REWERSE WG I2

Mission

- **Integration of policies**
  - Security policies, Trust management
  - Business rules, Quality of service specs.

- Enhance **user control and awareness** on system behavior

- **Reduce the cost of building and maintaining** cooperative systems
Reference Scenario (I)
General Picture

- Each party has a policy to control the use of its resources
  - Service access control (security)
  - Credential disclosure control (privacy)
  - Business rules

- Decisions are based on peer properties

- Properties are established by iterative, bilateral disclosure of certificates and declarations, i.e. negotiations
Reference Scenario (II)
Example: Security and Privacy

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
Reference Scenario (III)

Explanations

Suppose Alice's request is rejected

She may want to ask questions like:

- Why didn't you accept my credit card?

Other possible queries

- How-to queries
- What-if queries
  - Would I get the special discount on financial products X if I were locally employed?
Reference Scenario (& IV)
Natural Language

We are aiming at natural rule/query formulation

- **Users can download the files in folder historical_data if the creation date precedes 1/1/2000**

Policy enforcement, negotiations, query answering should all be automatically derived from such specifications

- **Attempto Controlled English**
Current Achievements (I)
I2-D1: Rule-based Policy Specification: State of the Art and Future Work

- A revised version is currently submitted as a chapter of the book on **Security in Decentralized Data Management**
  Publisher: Springer

- Already received positive feedback from reviewers
Current Achievements (II)
I2-D2: Policy Language Specification

- **PROTUNE** Policy Language

Excerpts and extensions published in the proceedings of

- IEEE POLICY'05
- Semantic Web Policy Workshop at ISWC'05
Current Achievements (III)
I2-D3,I2-D5: ACE Extensions

- Prototype evolving as expected

- Start of a cooperation between I2 (Zurich) and A2 (Dresden) with the goal to express knowledge of protein interactions in Attempto Controlled English.
  - Started as a master thesis with the working title "Expressing Ontological Knowledge of Protein Interactions in Attempto Controlled English".
Current Achievements (IV)
I2-D4: Advanced Policy Queries (I)

- Currently being released
  - Waiting for second external review
- Advanced Query Answering
  - Why, Why-Not, How-to, What-if queries
- Two level of explanations
  - Concise
  - Full or detailed
- Constructed at client side
- Based on
  - Set of (computed) answer substitutions
  - Verbalization patterns
Policy example

[r2] allow( download(Resource) ) :-
   public(Resource).
[r3] allow( download(Resource) ) :-
   authenticated(User),
   has_subscription(User,Subscription),
   available_for(Resource,Subscription).
[r4] allow( download(Resource) ) :-
   authenticated(User),
   payed(User,Resource).

authenticated(User). explanation: [User,is,authenticated].
allow(download(Resource)). explanation: [it,is,allowed,to,download,Resource].
Current Achievements (VI)
I2-D4: Advanced Policy Queries (III)

- Why-not concise explanation for “allow(download(paper14.pdf))”

I CAN’T PROVE THAT it is allowed to download paper14.pdf

BECAUSE:
Rule [r3] is not applicable:
THERE IS NO User SUCH THAT User is authenticated [details]

AND
Rule [r4] is not applicable:
THERE IS NO User SUCH THAT
User is authenticated

MOREOVER
THERE IS NO User SUCH THAT
User has paid for paper14.pdf [details]
Current Achievements (VII)
I2-D4: Advanced Policy Queries (IV)

- Why-not concise explanation for “allow(download(paper14.pdf))”

Rule [r2]:
Everybody can download public objects
the rule is not applicable

Rule [r3]: (no summary available)
it is allowed to download paper01234.pdf IF
THERE EXIST User AND Subscription SUCH THAT
User is authenticated
AND
User has subscription Subscription
AND
paper01234.pdf is available for Subscription
the rule is not applicable

Rule [r4]:
Users can download any object if they pay for it
the rule is not applicable

[see internal format]
[details]
[details]
[see internal format]
Current Achievements (VIII)
I2-D4: Advanced Policy Queries (V)

- Policy Example

[r9] id(Cred) :-
    valid_credential(Cred),
    Cred.type:T,
    Cred.issuer:CA,
    isa(T,id),
    trusted_for(CA,T).

Id(Cred).explanation:[Cred,is,a,credential].
trusted_for(CA,Type):[CA,is,trusted,for,Type].

- Why-not concise explanation for “id(Credential)” (with a disclosed credential)

I CAN’T FIND ANY Cred SUCH THAT Cred is an id BECAUSE:
c012 is a credential with type id and issuer Open University [details]
id is an id_type
BUT
IT IS NOT THE CASE THAT Open University is trusted for id [details]
Questions

WG I2 Coordinator: Piero A. Bonatti (Naples)
PROTUNE Policy Language (I)
The Language

A logic programming language with “reserved predicates”

Types of predicates:

- **Decision Predicates:** can be queried by users
- **Abbreviation/Abstraction Predicates**
- **Constraint Predicates:** comprise usual equality and disequality predicates
- **State Query Predicates:** read the state without modifying it
- **Provisional Predicates:** may be made true by means of associated actions that may modify the current state
PROTUNE Policy Language (II)

Example

Authorization rules

allow(enter_site()) :-
    declaration(usr=U,pwd=P),
    has_passwd(U,P)

allow(buy()) :-
    credential(type=visa, num=N, exp=E),
    not blocked(N), E>today
Authorization rules

allow(enter_site()) :-
    declaration(usr=U,pwd=PWD),
    has_passwd(U,P)

allow(buy()) :-
    credential(type=visa, num=N, exp=E),
    not blocked(N), E > today

- To be provided by the client
- Similar to an abduction problem
Authorization rules

allow(enter_site()) :-
    declaration(usr=U,pwd=P),
    has_password(U,P)

allow(buy()) :-
    credential(type=visa, num=N, exp=E),
    not blocked(N), E>today

- May be associated to actions
- E.g. Contact VISA web site
PROTUNE Policy Language (V)

Local State Predicates

Authorization rules

allow(enter_site()) :-
   declaration(usr=U,pwd=P),
   has_passwd(U,P)

allow(buy()) :-
   credential(type=visa, num=N, exp=E),
   not blocked(N), E>today

- Local state predicates – sensitive
- Requires policy filtering
Authorization rules

allow(enter_site()) :-
    declaration(usr=U,pwd=P),
    has_passwd(U,P)

allow(buy()) :-
    credential(type=visa, num=N, exp=E),
    not blocked(N), E>today

- Time dependent state terms/predicates
- Including the **negotiation state**
## PROTUNE Policy Language (VII)
### Metapolicies (I)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Domain</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>provisional predicates</td>
<td>commands</td>
</tr>
<tr>
<td>actor</td>
<td>provisional predicates</td>
<td>self, peer</td>
</tr>
<tr>
<td>aggregation_method</td>
<td>cost and sensitivity attributes</td>
<td>max, min, sum, adopt(Predicate)</td>
</tr>
<tr>
<td>cost</td>
<td>provisional predicates</td>
<td>number</td>
</tr>
<tr>
<td>evaluation</td>
<td>state predicates</td>
<td>immediate, delayed, concurrent</td>
</tr>
<tr>
<td>expected_outcome</td>
<td>provisional predicates</td>
<td>success, failure, undefined, unknown</td>
</tr>
<tr>
<td>explanation</td>
<td>literals and rules</td>
<td>string expression</td>
</tr>
<tr>
<td>ontology</td>
<td>abbreviation predicates, credentials, declarations, actions</td>
<td>URI</td>
</tr>
<tr>
<td>predicate</td>
<td>literals</td>
<td>predicate names</td>
</tr>
<tr>
<td>selection_method</td>
<td>negotiator</td>
<td>certain_first, order(attribute_list), adopt(Predicate)</td>
</tr>
<tr>
<td>sensitivity</td>
<td>predicates, literals, rules</td>
<td>public, private, not_applicable</td>
</tr>
<tr>
<td>type</td>
<td>predicates, literals</td>
<td>abbreviation, constraint, decision, state_predicate, provisional, state_query</td>
</tr>
</tbody>
</table>
table(Key,Data).evaluation:immediate ←
ground(Key).

logged(Msg,File).action:‘echo’+Msg+‘>’+File.

credential(_).ontology:URI.

abbrev(_).explanation:“this condition checks...”
Current Achievements (III)
I2-D4: Policy Language Specification (I)

- Policy example

```
allow(download(Resource))

public(Resource) authenticated(User) payed(User,Resource) authenticated(User) has_subscription(User,Subscription) available_for(Resource,Subscription)

id(Credential), Credential.name:User,
challenge(K), declaration([username=User,password=P]),
password(User,P), doc('http://lol.com/register.php')

accepted_credential(Cred), Cred.type:T, Cred.issuer:CA
isa(T,id), trusted_for(CA,T)
```

```text
authenticated(User).explanation: [User,is,authenticated].
allow(download(Resource)).
explanation: [it,is,allowed,to,download,Resource].
```
Promotional Activities

- Position paper at Semantic Web Policy Workshop at ISWC’05

- Invitation as a panelist at COMPSAC'05 in Edinburgh in a Panel on Security and Privacy in Distributed Collaborative Systems (July 2005))

- Course on Attempto Controlled English at the University of Tartu (Estonia), Nov. 2005

- Invited Presentation of the Attempto/REWERSE work at the University of Tartu (Estonia), Nov. 2005

- Invitation to present the Attempto/REWERSE work in the Turing Center's Distinguished Lecture Series at the University of Washington (Seattle), Dec. 2005.