The REWERSE View on Policies

Piero Bonatti, November 7th, 2005
Contents

- The view of REWERSE on policies for the Semantic Web
  - Important features
  - Technical challenges

- REWERSE is one of the 2 EU NoE devoted to the SW
  - focused on rule-based techniques
  - policies identified as crucial area
  - WG I2 devoted to policy specification, composition, conformance
What is a Policy

For us the term policy covers:

- Security policies & Trust management
- Business rules
- Quality of Service directives
- ...

*more about this in the panel*

All these policies make decisions based on similar pieces of information (evidence)

- user age, nationality, customer profile, identity, reputation...
Many Policies, One Framework

It is appealing to integrate all these notions in one framework

- One common infrastructure
  - for interoperability and decision making
- Where policies can be harmonized & coordinated

**Challenge:** harmonize and integrate different requirements

- procedural (ECA) vs. declarative semantics
- top-down, bottom-up derivation strategies
- deduction vs. abduction ...
Strong, Soft, and Lightweight Evidence

How can individuals *prove* their eligibility?

- Strong evidence, e.g. **digital credentials**
- Soft evidence, e.g. **numerical reputation measures**
- Lightweight evidence, e.g. **unsigned declarations**

They should be integrated for balancing:

- trust level
- risk level
- computational costs
- usability (fetching credentials, personal assistants)

(see also the paper in the workshop proceedings)
Strong, Soft, and Lightweight Evidence

Challenges:

- Research on reputation models still in early stage
  - new models keep being introduced
  - vulnerabilities (e.g., to coalitions)
  - adopt parametric frameworks? (current choice of REWERSE)

- Interoperability
  - lightweight evidence can be based on any web contents
  - how to explain requirements in a machine-understandable way?
  - a standard semantic web issue – ontologies
  - still lightweight?...
Trust Management (TM)

- Encode trust
  - already discussed in a more general setting (evidence)
- Acquire / submit evidence, encode requests
  - negotiations
- Make decisions, execute actions
  - rule-based
  - provisional policies

There exists much work on TM

- Are (semantic) web scenarios different from those considered by the TM community?
- Any difference in their requirements?
Formulating information requests

A Semantic Web approach:

- **Publish policies** (rules) describing which information is needed to get services, facilities, credentials, ...
  - eligibility criteria, user classes, and other auxiliary concepts
  - in fact peers are exchanging their ontologies

- A new idea?

  - **No**: see CCS 2000
  - Ontologies are based on minimal shared knowledge: X.509+rule semantics
    - This makes the approach plausible even in short-mid time
    - Very important for Semantic Web ideas
Publishing rules

Not *all* rules!

- Policies are sensitive even in “personal” scenarios
  - “Picture 12.jpg can be seen only by my best friends”
  - How would an “ordinary” friend react to a denial?...

- Personal information sharing scenarios are not different from standard TM scenarios

- More policy rules can be disclosed as more information about the peer is gathered
  - it encourages multiple iterations
Negotiations

Multiple steps motivated by

- The need for incremental information release
  - sensitive policies
  - minimizing personal information disclosure
- Counter-requests of the peer asking for a resource
  - “how are you handling the information I'm giving you?”
  - proving certifications
Negotiations

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
Stateful Negotiations?

Some researchers argue they are undesirable on the web

- Saving states on the peers is not strictly necessary
  - disclosed information can be replicated in each message
  - although messages get longer
- In practice important web sites adopt stateful transactions
  - despite heavy traffic load
- Would it be better to have stateful web protocols?
  - probably lead more robust and secure
  - think of cookie-related vulnerabilities
Summarizing

Trust Management: What's new?

- Apparently, only a more courageous approach to lightweight evidence

**Challenges:** all open challenges in TM, including:

- “policy interoperability”: guaranteeing negotiation success when policies “theoretically” permit it

- optimal negotiations: which strategies / policy features minimize the amount and the sensitivity of disclosed information

- more generally: making concrete decisions in the choice space given by regulations
Cooperative policy enforcement

Crucial for the success of a web service

- Never say “no”!
- Encourage first-time users
- Explain policy decisions
  - Advanced queries: Why / why not
- Guide users in acquiring missing permissions
  - Activate registration procedures
  - Provide instructions
  - Advanced queries: how-to, what-if
- Enhance users awareness about the policy applied by their system
  - A necessary precondition for fully exploiting the system's security mechanisms
Explanation mechanism

Main challenge:

- Finding the right tradeoff between
  - Explanation quality
  - Framework instantiation effort
    - The framework needs to be adapted to each application domain
    - Reduce the need for specialized staff
Explanation mechanism (how to)

TO MAKE SURE THAT User is allowed to download Resource

NOTHING NEEDS TO BE DONE IF Resource is public

[details]

ALTERNATIVELY

PLEASE MAKE SURE THAT

User is authenticated

AND

User has subscription

[details]

ALTERNATIVELY

PLEASE MAKE SURE THAT

User is authenticated

AND

User has paid for Resource

[details]

POLICY
allow(User,download(Resource)) ←
public(Resource).
allow(download(Resource)) ←
authenticated(User),
hasSubscription(User).
allow(download(Resource) ←
authenticated(User),
paid(User,Resource).

METAPOLICY
allow(X,download(Resource)).explanation:
[X,is,allowed,to,download,Resource].
public(Resource).explanation:
[Resource,is,public].
authenticated(User).explanation:
[User,is,authenticated].
hasSubscription(User).explanation:
[User,has,subscription].
paid(User,Resource).explanation:
[User,has,paid,for,Resource].
Explanation mechanism (why not)

“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
Pruning: User is not authenticated so it makes no sense to inspect her subscriptions

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

“authenticated” depends on a credential. “hasSubscription” depends on “authenticated”

POLICY
[r3]: allow(download(Resource)) ← authenticated(User),
hasSubscription(User).
[r4]: allow(download(Resource)) ← authenticated(User),
paid(User, Resource).

METAPOLICY
allow(download(Resource)).explanation:
[it,is,allowed,to,download,Resource].
public(Resource).explanation:
[Resource,is,public].
authenticated(User).explanation:
[User,is,authenticated].
hasSubscription(User).explanation:
[User,has,subscription].
paid(User, Resource).explanation:
[User, has, paid, for, Resource].

Pruning: User is not authenticated so it makes no sense to inspect her subscriptions

authenticated(User).explanation:
[User,is,authenticated].
hasSubscription(User).explanation:
[User,has,subscription].
paid(User, Resource).explanation:
[User,has,paid,for,Resource].

Piero Bonatti, University of Naples
The REWERSE View on Policies

SWPW, Nov.7 2005
Controlled natural language specifications

- Our goal is formulating rules such as:
  - “Academic users can download the files in folder historical_data whenever their creation date precedes 1942”
  - Internal format: rules

- Very important for giving users greater control on the policy applied by their system
  - A necessary precondition for fully exploiting the system's security mechanisms
REWERENCE's policy framework

**Protune**

- First attempt at tackling all the aforementioned issues simultaneously
- **Metapolicies** for driving negotiations declaratively
  - e.g. Rule sensitivity, action execution time
- and for instantiating the framework in application scenarios
  - New actions and responsible actors
  - Verbalization directives
- Integrating **legacy software and numerical reputations**
- Explanation mechanism
Implementations

Extensions and improvements of

- The Trust Management System **PeerTrust**

- Attempto Controlled English system (ACE)
  - for natural language specification
  - for query answering
  - [http://www.ifi.unizh.ch/attempto/](http://www.ifi.unizh.ch/attempto/)
Metapolicy examples

table(Key,Data).evaluation:immediate ← ground(Key).

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

credential(_).ontology:URI.

abbrev(_).explanation:”this condition checks…”
Publications
Publications

Important REWERSE related ...


Publications

Important REWERSE related …


Note: A foundational study related to forthcoming deliverables on policy validation and composition.

- More publications on rewerse.net.
Mission

Working Group I2 aims at designing **policy languages** and **policy-driven systems** that exploit **semantic web techniques** to enhance user privacy, web service usability and protection, and improve user control on the policies applied by open systems and services.

**Security & Privacy Protection**
- often in conflict with system **usability**
- providing & gathering security-related inform. or certificates
  - goal: progressively moved from users to machines

**Enhancing User Control & Awareness** on System Behavior
- common users
  - specify their own rules
  - understand the automated decisions of the system
    ⇒ are given high-level tools (e.g. **natural language parsers**)
      - to formulate policies and to ask systems for explanations

**explanation facilities**
- attract occasional users
  ⇒ may make a web service more competitive