Protune: Natural Language for the Specification of Rule Based Policies on the Semantic Web

Daniel Olmedilla, Telefonica R&D

Invited Talk at the Business Rules: Extraction, Modelling and Integration Session

23rd European Conference on Operational Research, July 7, 2009, Bonn

Index

- The term *policy covers:*
  - Security/Privacy policies, Trust management
  - Business rules
  - Quality of Service directives
  - Service-level agreements
  - *and more…*

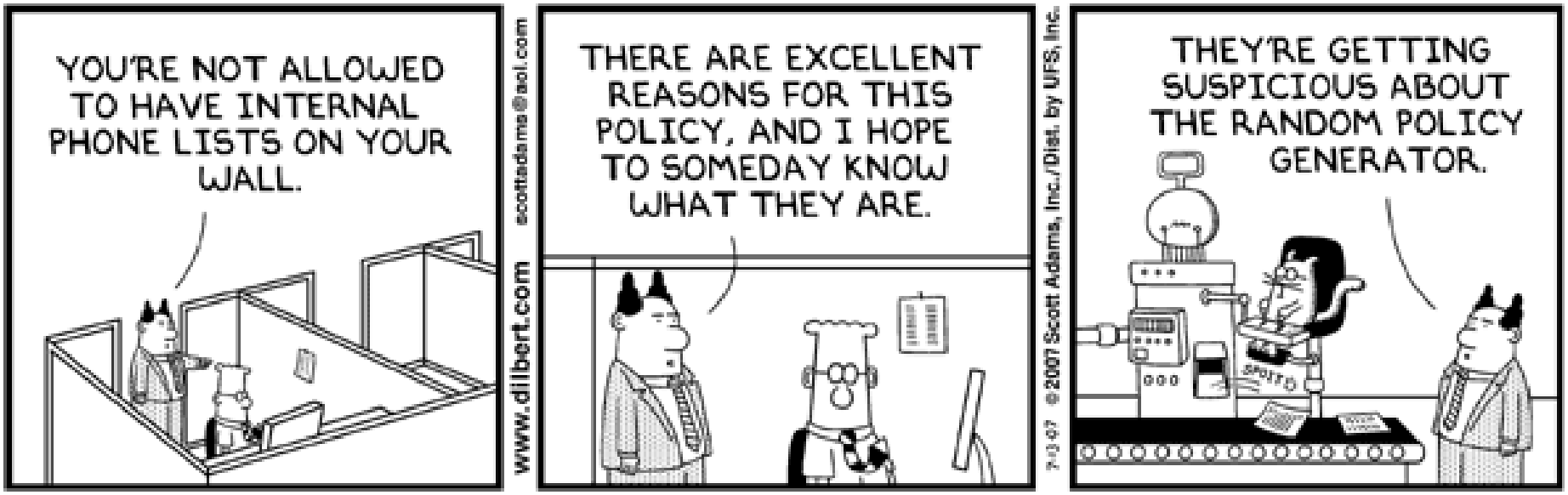
01 Introduction

Policy Examples

- Business Rules
  - *"Up to 15% of network bandwidth can be reserved by paying with an accepted credit card"*
- Pervasive Computing
  - *"My colleagues can only see the building I am in and only when they are on company premises"*
- Enterprise Collaboration
  - *"Only disclose inventory levels to customers with past due shipments"*
- DoD Scenarios (e.g., coalition forces)
  - *"Only disclose ship departure time after the ship has left"*
  - *"Only disclose information specific to the context of ongoing joint operations"*
- Homeland Security & Privacy (e.g., video surveillance)
  - *"Only allow for facial recognition when a crime scene is suspected"*

02 Challenge

User Awareness & Control



- **Encourage people to personalize their policies**
  - **Make it easy for users to write their own rules**
- **Explain policies and system decisions**
  - **Make rules & reasoning intelligible to the common user**

Natural Language Specification

Policy Explanations

03 Protune Policy Framework

Overview

- provides a **logic-based**, **declarative** policy language
- features include
  - **trust negotiation**
  - **external actions**
    - access to relational databases,
    - RDF stores,
    - file system requests,
    - time and location-aware packages
  - natural language policy **specification**
  - natural language policy **explanations**
    - "*You cannot access because …*" (in contrast to just "*Access denied.*")

04 Natural Language Policy Specification

How does a policy look like?

Policies (typically) are

- machine-understandable
- declarative
- formal syntax
- unintuitive and hard to grasp for users

An example:

allow(access(Requester,Resource)) ←
friend(Requester), family_picture(Resource).

What could be translated to

If the requester is a friend and the resource is a family-picture then the requester can access the resource.

04 Natural Language Policy Specification

Controlled Natural Language

Is a subset of natural language (NL)

- They are formal languages
  - CNLs have a formal, machine-understandable semantics that is unambiguous

    "Bob sees the girl with the telescope."

    "Bob and John are friends. He is my friend, too."

  are always given the same semantics.
- But They do not look like formal languages
  - They are more user-friendly

We use ACE (Attempto Controlled English)

04 Natural Language Policy Specification

ProACE: a Controlled Natural Policy Language

- A subset of the controlled natural language ACE
- Each ProACE can be translated into a Protune policy

  - *"If the requester is older than 18 and she is Bob's friend then she can access everything which is in 'adult-content-folder'."*
  - *"If the requester is a friend and the resource is a family-picture then the requester can access the resource."*
  - *"If the company of a credit-card is "VISA" then the credit-card is accepted."*

04 Natural Language Policy Specification

ProACE Mapping Examples

John waits.	`wait('John').`
John greets Sue.	`greet('John', 'Sue').`
John is old.	`old('John').`
Every resource is private.	`private(X).`
John is the murderer.	`murderer('John').`
The murderer is John.	`murderer('John').`
John is in London with Sue.	`inWith('John', 'London', 'Sue').`
John is with Sue in London.	`inWith('John', 'London', 'Sue').`
John is as old as Sue.	`asOldAs('John', 'Sue').`
John is fond-of Sue.	`fondOf('John', 'Sue').`
John is the son of Sue.	`'Sue'.son:'John'.`
John is the son of Sue with Bill.	`sonOfWith('John', 'Sue', 'Bill').`
John's mother waits.	`'John'.mother:X, wait(X).`
If the user is the owner of the file then the user can read the metadata of the file.	`allow(read(User, X)) :-` `  File.metadata:X, File.owner:User.`

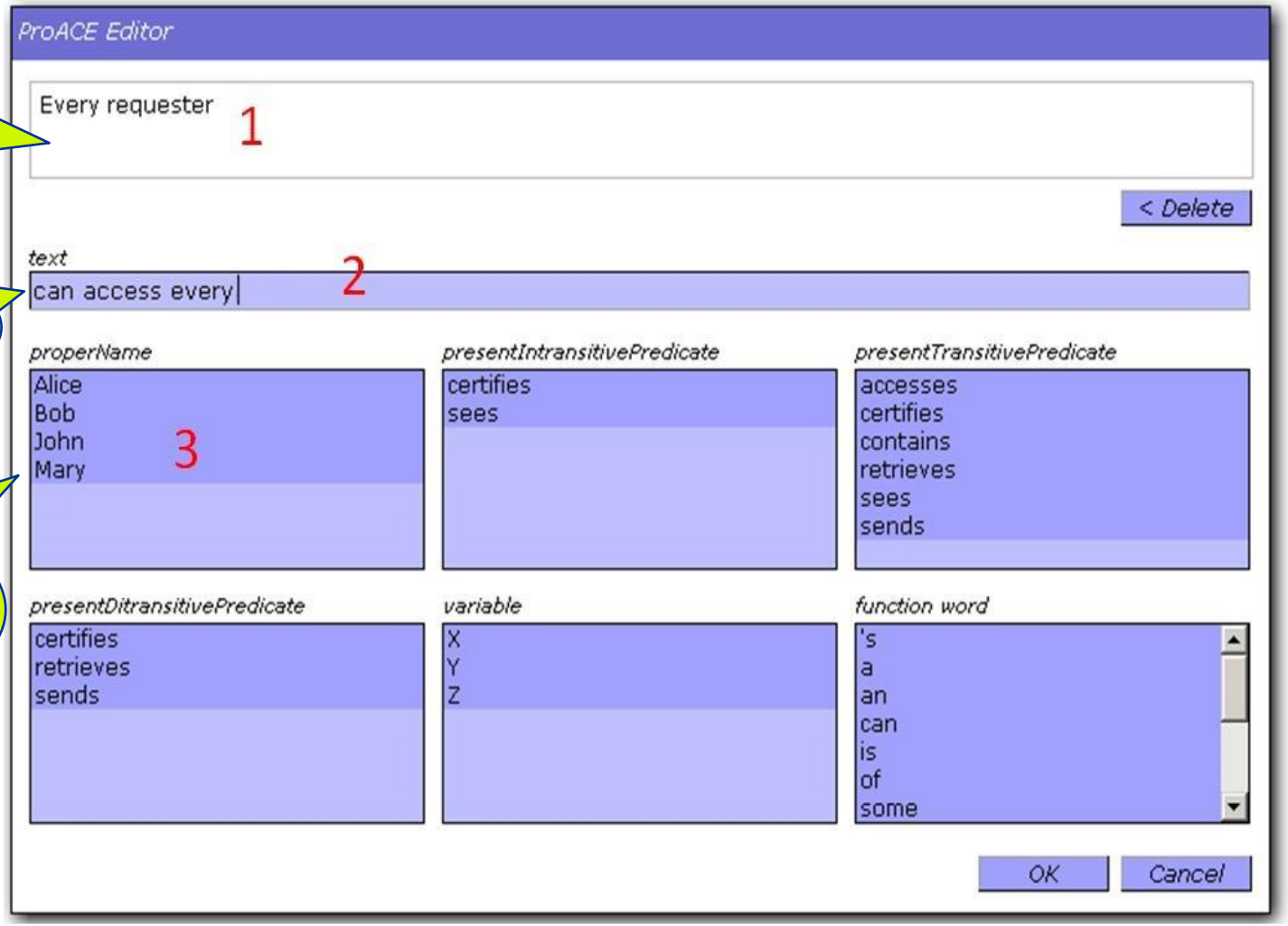
04 Natural Language Policy Specification

ProACE Editor

Accepted partial sentence

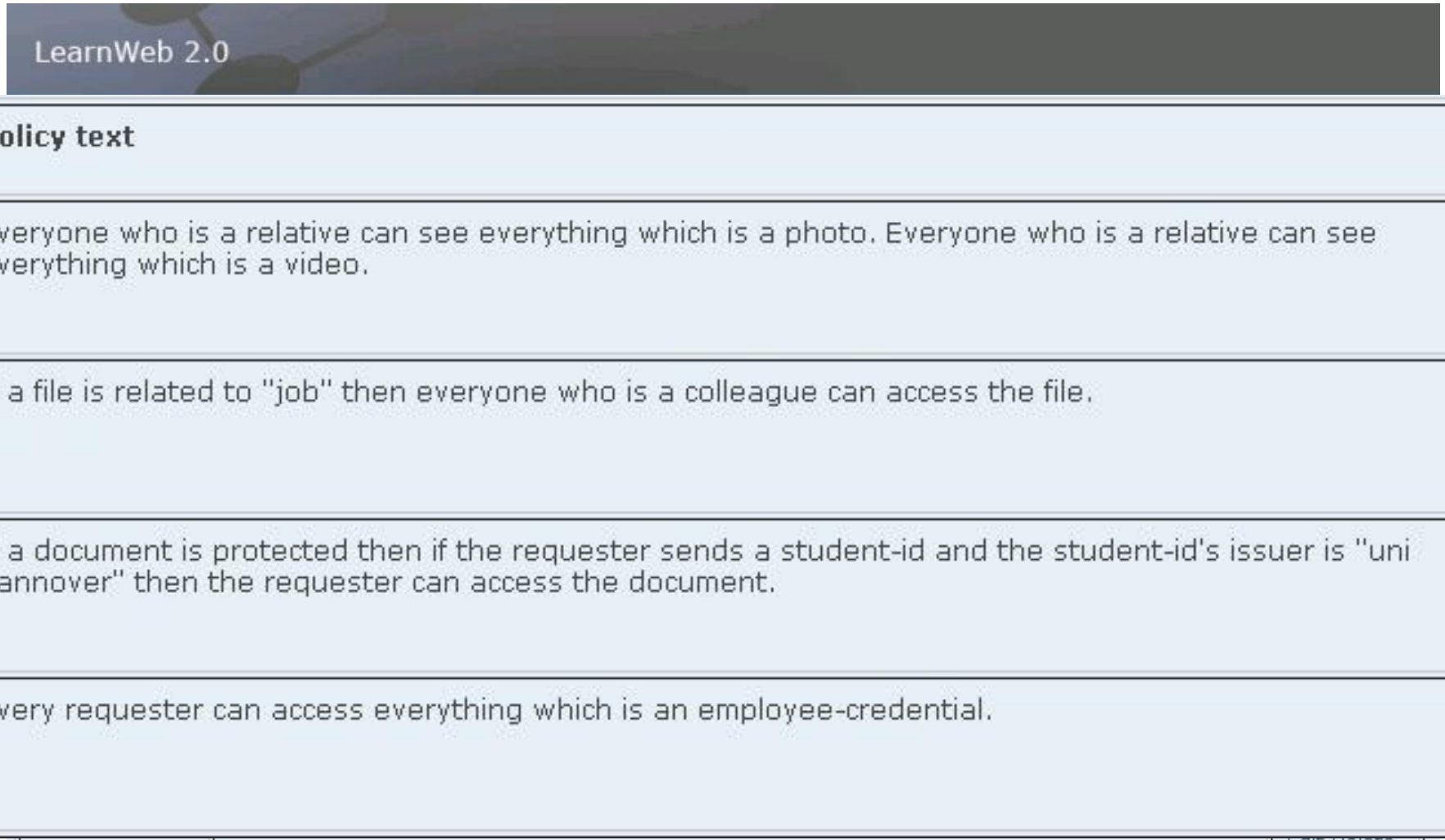
Field for direct text input

Lists for one-by-one word insertion



04 Natural Language Policy Specification

LearnWeb 2.0: Natural Language Policies in a social platform



05 Conclusions
Summary

- Protune is a framework where
  - Behavior is flexible
  - Covers many different policy types
  - Policies are rule based, and allow for reasoning
  - Allows for semantic and privacy aware negotiations
    - No previous shared knowledge required
    - Allows for semantic and privacy aware negotiations
  - Provides user awareness and control, i.e.,
    - Natural language policy specification
    - Natural language policy explanations

06 References
Demonstration & Prototype

- Demonstration
  - http://policy.L3S.uni-hannover.de/
- Prototype available
  - http://skydev.L3S.uni-hannover.de/gf/project/protune/
  - Freely distributed
  - All in java
    - Easily configurable, multi-thread
    - Legacy systems integration: RDBMS, LDAP, RDF repositories, …

06 References

Some Publications

- Bonatti, Olmedilla. **Driving and monitoring provisional trust negotiation with metapolicies**. In *6th IEEE Policies for Distributed Systems and Networks (POLICY 2005)*. IEEE, 2005.
- Bonatti, Olmedilla, Peer. **Advanced policy explanations on the web**. In *17th European Conference on Artificial Intelligence (ECAI 2006)*. IOS Press, 2006.
- Antoniou et al., Rule-based policy specification. **Secure Data Management in Decentralized Systems**. Springer, 2007.
- Bonatti, Olmedilla. **Rule-based policy representation and reasoning for the semantic web**. In *Reasoning Web, Third International Summer School 2007*. Springer.
- De Coi et al., **Controlled Natural Language Policies**, submitted for publication, 2009.

Thanks!

Questions?

http://www.olmedilla.info/
danieloc@TID.es

Work performed in collaboration with
Juri L. De Coi - L3S Research Center, Germany
Tobias Kuhn - University of Zurich, Switzerland