Linguaggi Semantici per la Rappresentazione e Gestione di Politiche di Controllo

DEIS – Università di Bologna

Firb – Web Minds: "Profili e Metadati"- Bologna, 11/12/2003

Outline

- Policy-based management: Why?
  - □ Motivation and background
  - □ Emerging policy-based management
- Policy-based management of multi-agent and distributed system
  - □ A traditional approach: Ponder
  - □ Semantic Web Languages for policy specification: KAoS and Rei
  - □ Comparison of KAoS, Rei and Ponder
    - Main benefits and drawbacks of Semantic Web Languages for policy specification, reasoning and deployment
- POEMA: Middleware for policy-controlled mobile applications

Motivations and background

Policies are constraints that dynamically regulate the behavior of a system without changing code nor requiring the cooperation of the components being governed

- ⇨ Benefits: Reusability, efficiency,extensibility, context-sensitivity,verifiability, reasoning over component behavior...

- Policies for network management
  - □ Automation of complex management task: configuration, security, recovery, QoS
- New policy management fields:
  - □ Management of full range of behavior for multi-agent and distributed systems

Policy-based management of multi-agent and distributed systems

- Technical policy categories
  - □ Authentication
  - □ Access and protection
  - □ Communication
  - □ Resource control
  - □ Monitoring and response
  - □ Mobility
- Social policy categories
  - □ Social organization
  - □ Notification
  - □ Conversation
  - □ Nonverbal expression
  - □ Collaboration and teamwork
  - □ Adjustable autonomy

Policy Representation

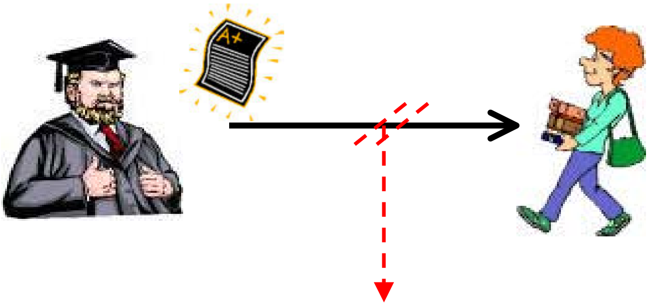
How to represent a policy

- Some current approaches to rule agent behavior…
  - □ PONDER: an object oriented and declarative language mainly adopted for Object-Oriented distributed systems
  - □ KAoS: a policy framework that uses DAML to represent policies
  - □ REI: a policy framework that uses deontic concepts together with RDF-S to represent policies

G. Tonti, J. Bradshaw, R. Jeffers, R. Montanari, N. Suri, and A. Uszok
"Semantic Web Languages for Policy Representation and Reasoning: A Comparison between KAoS, Rei, and Ponder"
In proceedings of the *2nd International Semantic Web Conference (ISWC 2003)*

Policy Specification: an example

- **Communication Policy example:**



"professors are permitted to communicate the final examination grade to their students using an encrypted communication only after the approval of the institute's director"

Ponder Policy Language

- **Declarative and Object Oriented language**
  - □ Ponder is not a Semantic Web language
  - □ Widely adopted in many object-oriented applications
  - □ Pioneer of many policy management concept

Example of policy specification

```
inst auth-  PolExample {
   subject s = people/guest;
   action print;
   target t = printer/Lab2_printer;
   when time.between(“21:00”, “08:00”);
}
```

Ponder – Policy Specification

- **Communication Policy example:**

```
domain prof = /SysEntities/Agents/ProfessorAgents;
domain stud = /SysEntities/Agents/StudentsAgents;

inst auth+ ExamGradeCommunication {
   subject  s= prof;
   target  t = /SysEntities/SysServices/CommunicationChannel;
   action t.communication (“Encrypted”, data, destination) ;
   when data.getType = “Grade”
     && destination == (stud -> select (st | st.professor == s))
     && s.receivedApproval(s.getInstituteDirector()) == ‘true’ ;
}
```

Ponder - Policy Specification

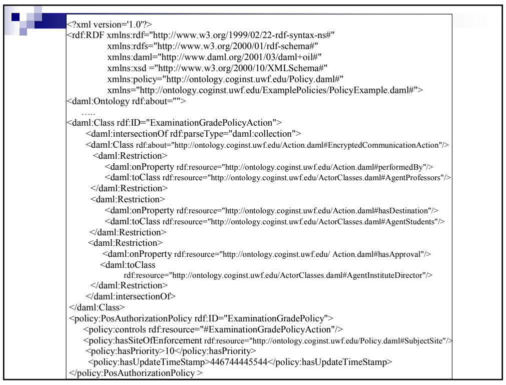
Ponder can describe any rule to constrain the behavior of components, in a simple and declarative way

…however…

- Ponder does not take care of the description of the content of the policy (e.g. description of the specified components, the system, etc.)

The adoption of a semantic web language can overcome this limitation

KAoS - Policy Specification



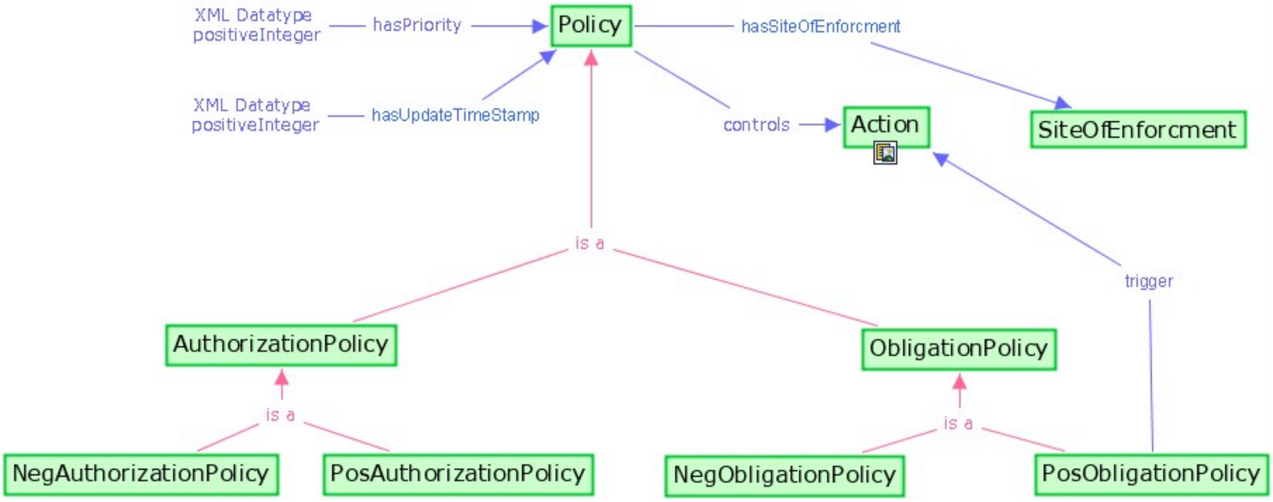
```
<?xml version='1.0'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
            xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
            xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
            xmlns:policy="http://ontology.coginst.uwf.edu/Policy.daml#"
            xmlns="http://ontology.coginst.uwf.edu/ExamplePolicies/PolicyExample.daml#">
<daml:Ontology rdf:about="">
    …..
<daml:Class rdf:ID="ExaminationGradePolicyAction">
      <daml:intersectionOf rdf:parseType="daml:collection">
      <daml:Class rdf:about="http://ontology.coginst.uwf.edu/Action.daml#EncryptedCommunicationAction"/>
        <daml:Restriction>
           <daml:onProperty rdf:resource="http://ontology.coginst.uwf.edu/Action.daml#performedBy"/>
           <daml:toClass rdf:resource="http://ontology.coginst.uwf.edu/ActorClasses.daml#AgentProfessors"/>
       </daml:Restriction>
       <daml:Restriction>
           <daml:onProperty rdf:resource="http://ontology.coginst.uwf.edu/Action.daml#hasDestination"/>
           <daml:toClass rdf:resource="http://ontology.coginst.uwf.edu/ActorClasses.daml#AgentStudents"/>
       </daml:Restriction>
      <daml:Restriction>
          <daml:onProperty rdf:resource="http://ontology.coginst.uwf.edu/ Action.daml#hasApproval"/>
          <daml:toClass
                rdf:resource="http://ontology.coginst.uwf.edu/ActorClasses.daml#AgentInstituteDirector"/>
       </daml:Restriction>
      </daml:intersectionOf>
</daml:Class>
<policy:PosAuthorizationPolicy rdf:ID="ExaminationGradePolicy">
     <policy:controls rdf:resource="#ExaminationGradePolicyAction"/>
     <policy:hasSiteOfEnforcement rdf:resource="http://ontology.coginst.uwf.edu/Policy.daml#SubjectSite"/>
      <policy:hasPriority>10</policy:hasPriority>
      <policy:hasUpdateTimeStamp>446744445544</policy:hasUpdateTimeStamp>
</policy:PosAuthorizationPolicy >
```

KAoS policy

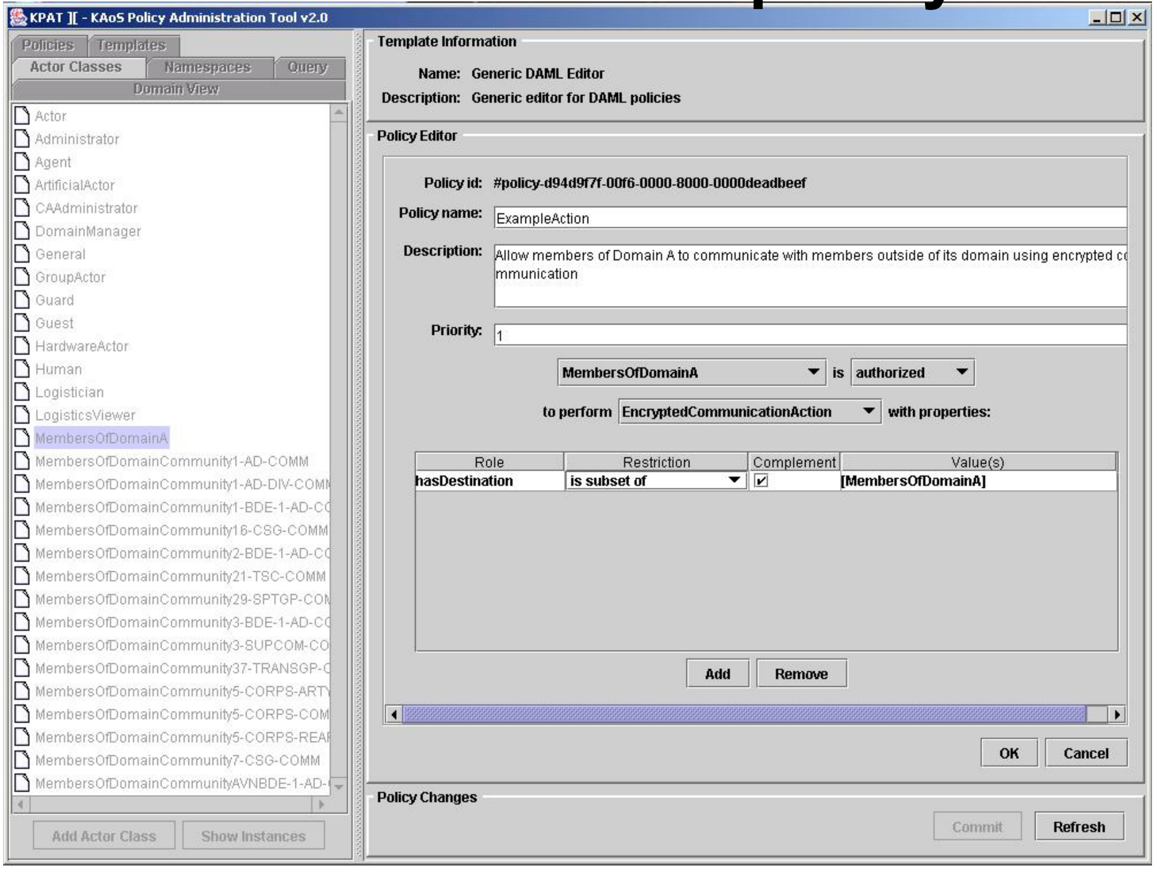
- **Policies and domains represented in DAML (soon OWL) as ontologies**
  - Classes and related properties to describe actions, actors, resources, situations, groups, and policies
- **Collection of policy management services**
  - Provides means to access the policy service from several agent and distributed computing environments (Nomads, CoABS Grid, Cougaar, Brahms, CORBA, OGSA-compliant grid computing, Web Services)

KAoS Ontology

- KAoS Policy Ontology distinguish between *authorization* and *obligation* policies



KAoS: KPAT Hides Complexity



Actor
Administrator
Agent
ArtificialActor
CAAdministrator
DomainManager
General
GroupActor
Guard
Guest
HardwareActor
Human
Logistician
LogisticsViewer
MembersOfDomainA
MembersOfDomainCommunity1-AD-COMM
MembersOfDomainCommunity1-AD-DIV-COMM
MembersOfDomainCommunity1-BDE-1-AD-CC
MembersOfDomainCommunity16-CSG-COMM
MembersOfDomainCommunity2-BDE-1-AD-CC
MembersOfDomainCommunity21-TSC-COMM
MembersOfDomainCommunity29-SPTGP-COM
MembersOfDomainCommunity3-BDE-1-AD-CC
MembersOfDomainCommunity3-SUPCOM-CO
MembersOfDomainCommunity37-TRANSGP-C
MembersOfDomainCommunity5-CORPS-ARTY
MembersOfDomainCommunity5-CORPS-COM
MembersOfDomainCommunity5-CORPS-REAF
MembersOfDomainCommunity7-CSG-COMM
MembersOfDomainCommunityAVNBDE-1-AD-

Template Information

Name: Generic DAML Editor
Description: Generic editor for DAML policies

Policy Editor

Policy id: #policy-d94d9f7f-00f6-0000-8000-0000deadbeef
Policy name: ExampleAction
Description: Allow members of Domain A to communicate with members outside of its domain using encrypted co
mmunication
Priority: 1

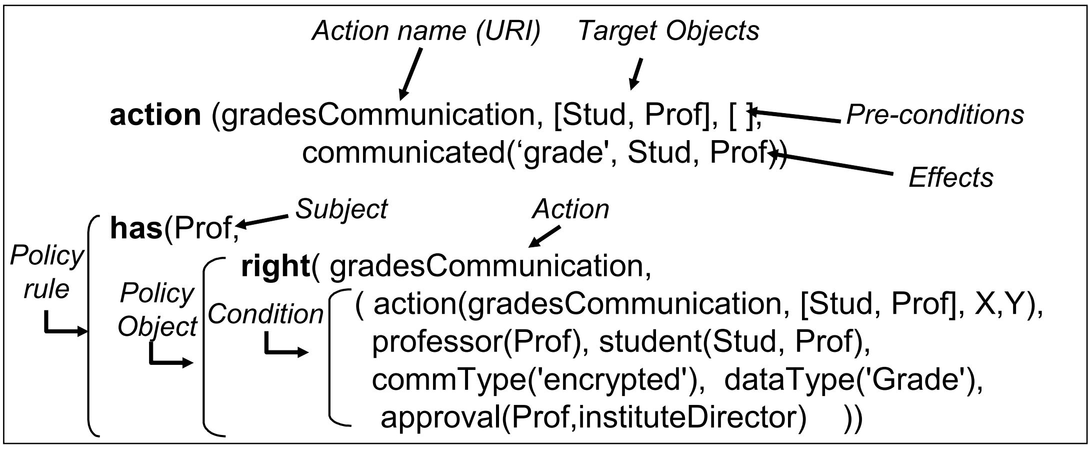
MembersOfDomainA is authorized

to perform EncryptedCommunicationAction with properties:

Role	Restriction	Complement	Value(s)
hasDestination	is subset of	☑	[MembersOfDomainA]

Policy Changes

Rei – Policy Specification



Action name (URI)

Target Objects

Pre-conditions

Effects

```
action (gradesCommunication, [Stud, Prof], [ ],
              communicated('grade', Stud, Prof))

has(Prof,
    right( gradesCommunication,
        ( action(gradesCommunication, [Stud, Prof], X,Y),
          professor(Prof), student(Stud, Prof),
          commType('encrypted'),  dataType('Grade'),
           approval(Prof,instituteDirector)    ))
```

Subject

Action

Policy rule

Policy Object

Condition

- Prolog-like syntax for policy specification
- A policy framework that supports policy specification analysis and reasoning in pervasive computing applications

Rei Ontology

- **Policies and domains represented in RDF-S as ontologies**
  - Domain-independent ontologies include description of 'Policies', 'Rules', 'Conditions', 'Entities' and 'Actions'
  - Rei accepts also domain-dependent ontologies, in any language that can be converted into the form of triple recognizable by the Rei Policy Engine

Example:

```
<rdfs:Class rdf:ID ="CommunicationAction">
   <rdfs:subClassOf  rdf:resource="DomainAction"/>
 </rdfs:Class>

 <rdf:Property rdf:ID="hasDestination">
    <rdfs:domain rdf:resource="#CommunicationAction"/>
 </rdf:Property>

 <rdf:Property rdf:ID="carriesMessage">
    <rdfs:domain rdf:resource="#CommunicationAction"/>
 </rdf:Property>
```

Comparison

	KAoS	Rei	Ponder
Ontology-based	Yes	Yes	No
Specification language	DAML/OWL	Rei: (Prolog-like syntax + RDF-S)	Ponder (declarative specification)
Tools for policy specification	KPAT – Graphical editor for ontology and policy management	No** *** a GUI is being developed for the next Rei version*	Graphical editor and compiler
Reasoning support	Java Theorem Prover	Prolog engine	Event calculus representation
Enforcement mechanisms	Need to write the code of appropriate enforcers and to insert them in entities to control ** *** Policy automation being explored for the next version*	Action execution is outside the Rei engine	Java interfaces for enforcement agents are provided

Semantic Web Languages for policy Specification: why?

	Semantic web languages for policy specification	**Ponder **** ** **used as example of non-semantic web language**
Expressiveness	Capable of representing concepts and behavior of any complex environment	Capable of controlling specific sorts of behavior within object-oriented systems
	Multiple levels of abstraction	Low level of abstraction: object level
	Easy to extend policy ontology at runtime with new concepts	Extensibility supported by object-oriented inheritance at compile-time
Analyzability	Ontology representation simplifies and directly supports policy reasoning, conflict detection and harmonization	Conflict detection requires transforming policy specification into an event calculus representation
	Simplified access to policy information by querying the ontology	Access to single policy object by API – Access to policy repository to be designed
Ease-of-use	Need of specialized tools to assist unskilled users with policy specification and interpretation	Language specifically designed for simple policy specification and direct readability
Enforceability	High-level specification requires skilled programmers or sophisticated policy automation mechanisms for enforcement	Detailed specifications can be directly mapped into policy enforcement mechanisms
	Policy sharing among heterogeneous systems requires an agreement on a common ontology	Policy sharing among heterogeneous systems requires agreement on interfaces

POEMA: Policy Enabled MObile Applications

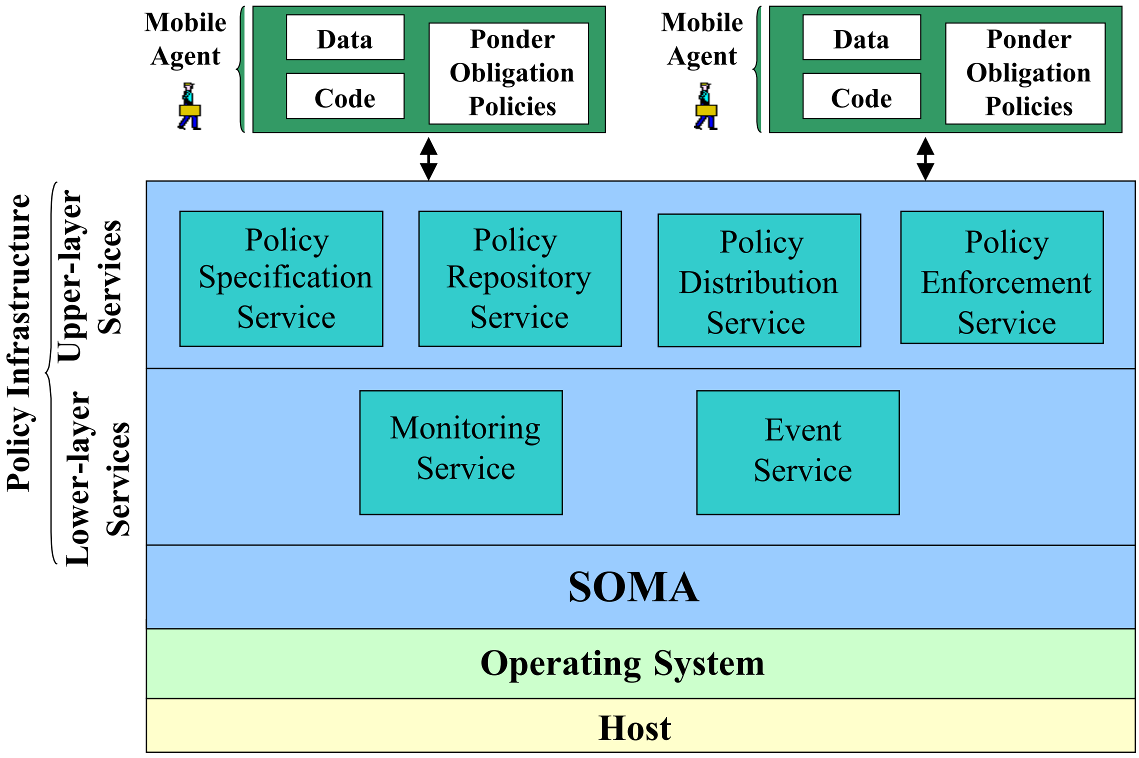
- **Policies for governing the mobility behaviour of mobile agents**
  - Separation of Concerns
  - Mobility Policies specify: ***When***, ***Where*** and ***Which Unit*** of mobility must migrate
  - Directly implementable policies represented in Ponder

```
inst oblig  MobPol1 {
  on CPULoad(90);
  subject s = agents/Manager;
 do s.go(G1.toString(), “run”);
 when
MonitoringSystem.isReachable(G1);
}
```

```
inst oblig  MobPol2 {
  on CPULoad(90);
  subject s = System/Relocator;
  target t = agents/Manager;
  do s.relocate(t, G1.toString(), “run”);
  when
  MonitoringSystem.isReachable(G1);
}
```

http://www.lia.deis.unibo.it/Research/POEMA/

POEMA Architecture



Mobile Agent
Data
Code
Ponder Obligation Policies
Mobile Agent
Data
Code
Ponder Obligation Policies
Policy Infrastructure
Upper-layer Services
Lower-layer Services
Policy Specification Service
Policy Repository Service
Policy Distribution Service
Policy Enforcement Service
Monitoring Service
Event Service
SOMA
Operating System
Host

Linguaggi semantici per la rappresentazione e gestione di politiche di controllo

DEIS – Università di Bologna

Firb – Web Minds: "Profili e Metadati"- Bologna, 11/12/2003