Quality and Gatekeeping Use Cases for the OOR

Ken Baclawski and Máximo Gurméndez
College of Computer and Information Science
Northeastern University
Motivation

- Quality is a major concern
  - Emphasized at the Ontology Summit 2008
  - Ontologies are similar to standards and publications
  - A mechanism for review is necessary

- Gatekeeping
  - Ensures minimum level of quality
  - Allows for many policies and workflows
Sources of Requirements and Use Cases

- ISO 11179: Metadata Repository (MDR)
- XMDR Project: Extended MDR
- Ontology Summit 2008
  - Several sessions devoted to requirements
  - Wiki page for requirements and use cases
  - What are the requirements?
Management Requirements

- Version management
- Configuration management
- Provenance and other metadata
- Metadata validation/authentication
  - Related to policy issues
- Ontology ownership and access control
Administration and Policy Issues

- Gatekeeping policies and enforcement
  - Overall policies or left to individual repositories

- Policy-making bodies
  - How are they constituted?
  - How can they interoperate?
Development of OOR Use Cases

- Consulted existing examples
- Assignment in software engineering course
- Classified all of the examples
- Developed Use Case Description Ontology
- Developed OOR use cases as instances of UCDO
- Available at [http://www.ccs.neu.edu/home/kenb/ontologies/](http://www.ccs.neu.edu/home/kenb/ontologies/)
Classification of Use Cases

- Administer user authorizations and privileges
- Assign identifiers
- Define workflows and policies
- Federate OOR instances
- Harmonize and map ontologies
- Publicize ontologies
- Query metadata
- Register ontologies
- Review and evaluate ontologies
Prototype Implementation

- Experimental
- Web Services Based
- ISO 11179 Foundations
- JBPM as Workflow Engine
Prototype: User Hierarchies

- Accreditor
- Registrar
- Steward
- Submitter
- User
Prototype: Process Definitions

Diagram of Process Definitions:

- **ProcessDefinition**
  - name: String
  - startState: State
  - states: List

- **State**
  - id: long
  - name: String
  - transitions: List

- **Transition**
  - id: long
  - transitionName: String
  - toState: String

- **ConditionTransition**
  - condition: String

- **DecisionState**

- **EndState**

- **FormTask**
  - asignee: String
  - instructions: String
  - fields: List

- **ModifyStatusState**
  - newStatus: String

- **ScriptTask**
  - script: String

- **Field**
  - id: long
  - type: String
  - name: String
  - defaultValue: String
  - length: int
Prototype: Sample Scenario

- Registrar uploads process definition:
  “Stewards validates model before it becomes a standard”
- Submitter uploads new model (Status=PENDING)
- Workflow Engine notifies Steward (according to process definition)
- Steward Validates Model (Form Task)
- Workflow Engine Modifies Status (Status=STANDARD)