

# Gatekeeping and Workflow Management for the Open Ontology Repository

A reference implementation

Kenneth Baclawski – Maximo Gurmendez

## Abstract:

This document describes the gatekeeping and workflow management component for the Open Ontology Repository. This system provides a mechanism for registering items such as ontologies, ontology mappings and ontology compositions (configurations) in a flexible manner, by allowing each registration authority to work with its own set of procedures. The component was based on a detailed set of use cases as well as the ISO 11179 standard. Procedures are specified using the Business Process Management language.

## Motivation

Ontologies and related artifacts are important for communication within a community and between communities. Communities can be very large, and the process whereby agreement is reached on the meaning of terms can be complicated. Accordingly, there must be a formal procedure for standardizing ontological artifacts. In particular, there must be defined roles as well as procedures. In order for an ontological artifact to become an official standard (according to a registration authority), the following aspects need to be considered:

- The procedures that specify the process by which an item becomes a standard.
- A repository that stores the latest version of the artifact serving as a single point of entry for every relevant actor to query the items.
- The roles that the actors play in the accreditation process.
- The provenance of an item registered on the system.
- The accessibility of items in the repository.

The gatekeeper allows any organization to become a registration authority upon the authorization of an accreditor (a singular entity that makes this decision). Representatives of a registration authority called registrars, stewards and submitters are able to flexibly define and follow through the different procedures that will regulate the processing of an artifact from its inception to the moment it becomes a standard. The gatekeeper is based on the ISO 11179, Metadata Repository specification [MDR].

## Main Concepts

As suggested in the MDR specification, each actor in the gatekeeper is either a registrar, steward, submitter or read only user. Table 1 shows the role each actor plays in the system.

<b>Actor Role</b>	<b>Description</b>
Registrar	Directly represents a registration authority. Makes higher level decisions regarding the items in the registry.
Steward	Represents a registration authority through a particular registrar. Makes the lower level decisions regarding accreditation process.
Submitter	Is able to submit items to the registry as allowed by a steward.
Read Only User	Allows access to the items in the system in a read only manner.

As can be appreciated in the table above, users form a hierarchy from the registrar to the read only user. Any user has a single “supervisor” which is the user that allowed the user into the system. For example when a registrar X allows a steward Y into the system, the supervisor of X is Y. In this manner each user responds to the user that allowed it into the system

The other main concept in the system is the actual item being subject to the registration process. We call this a “kept item” or an “administered item”. Any kept item that is submitted to the registry goes through the registration process as required for each registration authority. If there is no prescribed process, then the kept item goes through a default process. The default process is very simple and suitable for very small communities, which could even be a single individual who is developing ontologies for personal use.

Whenever an item is submitted to the system by a submitter, a process may request that the stewards or registrars take action regarding its accreditation. The actual registrars or stewards that will make these decisions will be the supervisors of the submitter as established by the hierarchy.

Actors can be created in the system in two different ways: by direct creation or by request. For example a new submitter may be created directly by the steward or a submitter can submit the request to a particular steward who will decide whether to approve it or not.

## **Use Cases**

The use cases were formalized as an instance of Use Case Description Ontology [KB2010]. The following list shows the main use cases that the gatekeeper supports:

- Accredited Registration Authority
- Steward Registration

- Submitter Registration
- Read Only User Registration
- Upload Process Definition (or procedures)
- Register an Item
- Query an Item
- Update an Item
- Complete information as required by a process definition
- Update contact information
- Query process definitions
- Query contact information

A detailed description of these use cases can be found in [KBMG2010]. For illustrative purposes, consider the following scenario:

1. Jack, a registrar, uploads a procedure that says: “stewards need to validate ontologies before they becomes a standard”
2. Submitter John uploads new ontology (the initial status is “Pending”).
3. The workflow engine notifies the steward Paul about the new ontology (according to process definition)
4. Paul, the steward, logs on to the system, and approves the ontology.
5. The workflow engine modifies the status of the ontology, now being “standard”.

Notice in the above scenario, how the system acts as a coordinator between the different actors towards the standardization of the artifact, and how the actual procedures are treated as **data** by system, modifying its behavior.

The procedures are defined using the Business Process Management (BPM) language. Here is an example of a process definition:

```
<?xmlversion="1.0"encoding="UTF-8"standalone="yes"?>
<process>
  <name>process_requires_registrar_and_steward_approval</name>
  <startState>Steward Approval</startState>
  <statesxsi:type="formTask">
    <name>Steward Approval</name>
    <instructions>
      Dear Steward, Please approve this ontology by setting approveSteward=y.
    </instructions>
    <transitions>
      <toState>Registrar Approval</toState>
```

```

        <transitionName>toRegistrarApproval</transitionName>
    </transitions>
    <assignee>steward</assignee>
    <fields>
        <length>10</length>
        <name>approveSteward</name>
        <type>java.lang.String</type>
    </fields>
</states>

<statesxsi:type="formTask"
    <name>Registrar Approval</name>
    <instructions>
Dear Registrar, Please approve this ontology by setting approveRegistrar=y.
</instructions>
    <transitions>
        <toState>Decide</toState>
        <transitionName>toDecision</transitionName>
    </transitions>
    <assignee>registrar</assignee>
    <fields>
        <length>10</length>
        <name>approveRegistrar</name>
        <type>java.lang.String</type>
    </fields>
</states>

<statesxsi:type="decisionState">
    <name>Decide</name>
    <transitionsxsi:type="conditionTransition">
        <toState>endState</toState>
        <transitionName>toEnd</transitionName>
        <condition>
            #{approveRegistrar!='y' || approveRegistrar!='y'}
        </condition>
    </transitions>
    <transitionsxsi:type="conditionTransition">
        <toState>mod</toState>
        <transitionName>toMod</transitionName>
        <condition>
            #{approveRegistrar=='y' & & approveSteward=='y'}
        </condition>
    </transitions>
</states>
<statesxsi:type="modifyStatusState">
    <name>mod</name>
    <transitions>
        <toState>endState</toState>
        <transitionName>toEnd</transitionName>
    </transitions>
    <newStatus>standard</newStatus>
</states>
<statesxsi:type="endState">
    <name>endState</name>
</states>
</process>

```

## References

[MDR] International Standards Organization, Metadata Repository 11179.

<http://metadata-stds.org/11179/>

[KB2010] Baclawski, K. Use Case Description Ontology,

<http://www.ccs.neu.edu/home/kenb/ontologies/>

[KBMG2010] Baclawski, Gurmendez 2010, OntologForum, Use Cases for the OOR.

<http://www.ccs.neu.edu/home/kenb/ontologies/oor-usecase.xml>