

# OOOR-KEEPER INTEGRATION PROPOSAL FOR HACKATHON

KENNETH BACLAWSKI, MÁXIMO GURMÉNDEZ  
NORTHEASTERN UNIVERSITY

## 1. ABSTRACT

KEEPER is an environment for provenance and entity registration of artifacts in a repository [1]. The Open Ontology Repository (OOR) is a proposed API and architecture for a federated collection of ontology repositories [2].

The purpose of this proposal is to define the goals of the OOR-KEEPER integration project. Any ontology repository should place a strong emphasis on the evaluation of the resources found within it. It is therefore necessary to provide a way for users with different roles to review and make standardized assertions on the quality and status of the items kept within the repository.

The process by which users review and make qualitative assessments on the ontologies may vary for different organizations and for different kinds of ontologies. The KEEPER project implements a flexible workflow system that allows authorized users to define and save such processes by formalizing the different stages of the review process and what the responsibility of each kind of actor is. Once these processes are defined, KEEPER will track the status of each item, hence enforcing that the defined process is executed as planned.

The goal of this project is to integrate the current OOR reference implementation with KEEPER so that any item stored in the repository are able to define a process by which the ontologies become a standard.

## 2. TECHNICAL CHALLENGE AND DELIVERABLES

The Open Ontology Repository (OOR) [2] consists of a code base forked from BioPortal [3]. The requirements for BioPortal and OOR differ. See [4] for a detailed analysis of the differences between these two. See [5,6] for the OOR requirements. The main programming language used in OOR is Ruby. KEEPER, on the other hand is a JEE application that exposes it's basic API via a WSDL/SOAP web service. The main challenge is to integrate these two systems so that:

1. An ontology in the OOR defines the process it will use for standardization.
2. Once the ontology is assigned to a process, the OOR will invoke the KEEPER service to start an instance of the process definition.
3. The OOR, shows for each user, the pending tasks that the different processes have assigned to it.

4. The OOR renders the forms to the appropriate user based on the process definition and submits to KEEPER all the information filled in on these forms.
5. Integrate authentication across OOR and KEEPER

### 3. COLLABORATORS

Kenneth Baclawski, Máximo Gurméndez  
Northeastern University

Additional collaborators are welcome.

### 4. SOURCES

The OOR Code source can be obtained in: to be advised  
The sources for KEEPER can be found in: to be advised

### 5. REFERENCES

1. K. Baclawski and M. Gurmendez. KEEPER: Knowledge Engineering Environment for Provenance and Entity Registration, A reference implementation, <http://www.ccs.neu.edu/home/mgurmend/pubs/keeper.pdf>
2. Open Ontology Repository, <http://oor.net>
3. BioPortal, <http://bioportal.bioontology.org>
4. OOR Sandbox and OOR Requirements - Comparative Analysis (October 15, 2009). [http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall\\_2009\\_10\\_15](http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2009_10_15)
5. K. Baclawski and M. Gurmendez. Quality and gatekeeping use cases for the Open Ontology Repository. Ontolog Forum. (April 1, 2010), <http://www.ccs.neu.edu/home/kenb/pub/2010/27/public.pdf>
6. K. Baclawski and M. Gurmendez. The OOR Requirements, <http://www.ccs.neu.edu/home/kenb/ontologies/oor-usecase.xml>