

OPEN ONTOLOGY REPOSITORY

Version: 0.2

HOME

Questions?

Please email authors for questions.

System

Open Ontology Repository [OOR]

Name: Open Ontology Repository

Authors: Ken Baclawski - Máximo Gurméndez -

Description: The OOR as a participant in use cases. All communication between any user and the OOR is assumed to occur on an authenticated and secure channel. Activities such as login are therefore not included in any of the use cases. The term item is used generically for any entity that is administered by the OOR with versioning and annotations such as provenance information. This includes ontology modules, mappings, configurations, compositions, process definitions and policies. Configuration is itself a generic term that includes contexts, frameworks and situations.

Organizations: Northeastern University - OOR Initiative -

Creation Date: May 20, 2010

System Actors

Accreditation Authority [Accreditor]

Description: The singular authority that manages the

registration authorities of the OOR. One important responsibility of the accreditor is the allocation of top-level names. Consequently, the accreditor is analogous to a domain name registrar. The primary difference is that domain names can be bought and sold, while the names allocated by the accreditor must be permanent and in the public domain.

KindOf: [#ProcessManager](#)

Registration Authority [Registrar]

Description: A top-level registration authority. It has a top-level name assigned by the accreditor. Standards bodies, professional societies and publishers are examples of registration authorities. Although a registrar is an organization, the organizational structure is not represented in the OOR. Consequently, a registrar is a kind of user with specific capabilities and responsibilities that are not common to all users. One can think of a registrar as a user that represents the registration authority during interactions with the OOR.

KindOf: [#ProcessManager](#)

Community Steward [Steward]

Description: A domain-specific community within the context of a registration authority. Examples include a working group of a standards body, a special interest group of a professional society, and a journal produced by a publisher. Although a community is a collection of individuals, its internal structure is not represented in the OOR. Consequently, a steward is a kind of user with specific capabilities and responsibilities that are not common to all users. One can think of a steward as a user that represents the community during interactions with the OOR.

KindOf: [#ProcessManager](#)

Content Submitter [Submittter]

Description: An individual that submits new materials or changes to existing materials that are managed by a specific steward. Depending on the steward, there may be no restrictions on who can be a submitter or there may be only a fixed set of submitters.

KindOf: [#ProcessParticipant](#)

Process Manager [ProcessManager]

Description: The common generalization of accreditors, registrars and stewards. They share the capability of creating and managing process and policy definitions.

KindOf: [#ProcessParticipant](#)

Process Participant [ProcessParticipant]

Description: A user that participates in an OOR process. Since processes are associated with registrars and stewards, a user that participates in an OOR process is implicitly also associated with that registrar or steward. For example, a user could contribute to the OOR by providing usage information and informal evaluations.

KindOf: [#User](#)

General User [User]

Description: A user of the OOR. This includes process managers and participants as well as individuals who are searching the OOR or retrieving from it in an ad hoc manner. Unlike process managers and participants, users in general need not be associated with a registrar or steward.

Use Cases

Accredit Registration Authority [**accredit**]

Description:

The accreditation authority decides whether to include a registration authority.

Step-by-step Description:

1. **[#User]** - A user requests that a new registration authority be admitted.
2. **[#OOR]** - The system queues the request and informs the Accreditor.
3. **[#Accreditor]** - The Accreditor accredits the new registration authority and grants the registrar role to the requesting user.
4. **[#OOR]** - The system notifies the user that the request was accepted.
5. **[#User]** - The user receives the notification.

Steward Registration [**registerSteward**]

Description:

A user wants to become a Steward for a registration authority.

Step-by-step Description:

1. Includes: **[#queryAuthorities]**
2. **[#User]** - The user places the request to be a steward to a particular registrar.
3. **[#OOR]** - The system notifies the registrar that a request has been made for a user to become a steward.
4. **[#Registrar]** - The registrar approves the

request.

5. **[#OOB]** - The system notifies the user.

Submitter Registration [**registerSubmitter**]

Description:

A user wants to become a content submitter for a registration authority.

Step-by-step Description:

1. Includes: **[#queryAuthorities]**
2. **[#User]** - The user places the request to become a content submitter for a particular steward.
3. **[#OOB]** - The system notifies the steward that a request has been made for the user to become a submitter.
4. **[#Steward]** - The steward approves the request.
5. **[#OOB]** - The system notifies the user that the request has been accepted.

Query the registration authorities [**queryAuthorities**]

Description:

A user queries the system to obtain information on relevant registration authorities, including the accreditor, registrars and stewards. This will only retrieve publicly released information about the

authorities.

Step-by-step Description:

1. **[#User]** - A user requests information on registration authorities that satisfy some specified criteria.
2. **[#OOR]** - The system returns the information about all relevant registration authorities, including all registrars and stewards associated with each registration authority.

Upload Process or Policy Definition [**uploadProcessDefinition**]

Description:

Upload a process or policy definition to the system. This may be a version of an existing process or policy. Such definitions are administered items, so this is a special case of item update in general, but such updates are performed by following a process. Modifying the process that one is using to modify the process could be result in an anomolous situation, so a separate use case was introduced.

Step-by-step Description:

1. **[#ProcessManager]** - The process manager requests the process and policy definition schemas.
2. **[#OOR]** - The system returns the schemas and explanations for instantiating process and policy definitions. A GUI would normally present the schema using a wizard.
3. **[#ProcessManager]** - The process manager uploads the process or policy definition to

the system.

4. [\[#OOR\]](#) - The system notifies the process manager that the process or policy definition has been uploaded successfully.

Register Item [[registerItem](#)]

Description:

A submitter registers an administered item. This is a new item, not a version of an existing item. Items include ontology modules, mappings and contexts. Process and policy definitions are administered items, but they have their own separate use case, and the participant is process manager rather than a submitter. Submission of a item is the first step in a process whereby an item progresses through a series of levels. The default process is very simple. Many communities will have more complex processes.

Step-by-step Description:

1. [\[#Submitter\]](#) - The submitter uploads an administered item.
2. [\[#OOR\]](#) - The system returns a confirmation message and a version number for the newly created administered item.
3. [\[#OOR\]](#) - The system enqueues a task for validating the uploaded item using the appropriate policies.

Validate Item [[validateItem](#)]

Description:

Validation tasks can be time-consuming so they are

enqueued for asynchronous processing.

Step-by-step Description:

1. [\[#00R\]](#) - The system performs a validation task in the queue according to the current policies for items belonging to the registration authority and steward.
2. [\[#Submitter\]](#) - The system notifies the original submitter of the result of the validation task.
 - a. Exception: [\[#validationFails\]](#)
3. [\[#Steward\]](#) - The system notifies the steward of the result of the validation task.
 - a. Alternative Flow:
[\[#notifyRegistrar\]](#)

Validation Failure [[validationFails](#)]

Description:

A validation task fails.

Step-by-step Description:

1. [\[#00R\]](#) - A validation task fails.

Query Administered Item [[queryItem](#)]

Description:

A user queries the system to find an administered item. Items include ontology modules, mappings and contexts. Process and policy definitions have their own use case.

Step-by-step Description:

1. **[#User]** - The user submits a combination subset of query component values. These can include title, release date and status. The user also specifies how to sort the results, and a desired rank range of results. For example: the user might search for "Virus Infection", released before 01/10/2009, with STANDARD status, sorted by release date, and returning the ranked list that goes from entry 20 to 30. Optionally, the specific version can also be included in the query.
2. **[#OOR]** - The system returns a list of administered items that match the user query and satisfy any access constraints.

Navigate Administered Item **[navigateItem]**

Description:

An elementary step during browsing. Unlike a query, the user has the item identifier rather than some search criteria. The result of the navigation is a set of attributes of the item that provide a minimal description of it as well as the item identifiers of related items.

Step-by-step Description:

1. **[#User]** - The user submits an item identifier.
2. **[#OOR]** - The system returns the list of attributes of the item that the user has the right to access.

Retrieve Administered Item **[retrieveItem]**

Description:

Retrieve the entire administered item. Some ontology modules, mappings or contexts are very large, so this should not be a common operation. Note that there is no requirement that the administered item be physically stored by the OOR instance. However, it should be able to retrieve items from wherever they are stored when requested.

Step-by-step Description:

1. [#User] - The user submits an item identifier.
2. [#OOR] - The system returns the administered item that the user has the right right to access.

Extract Administered Item Part [extractFromItem]

Description:

Retrieve part of an administered item. Some ontology module, mappings and contexts are very large, so partial retrieval may be necessary. This is the preferred use case for internal browsing of an ontology module or mapping.

Step-by-step Description:

1. [#User] - The user submits an item identifier and specification of a part of an item to be retrieved.
2. [#OOR] - The system returns the request part of the administered item that the user has the right to access.

Use Administered Item [useItem]

Description:

Register that the administered item is being used. This can be the basis for maintaining usage statistics.

Step-by-step Description:

1. [\[#User\]](#) - The user submits an item identifier.
2. [\[#OOR\]](#) - The system confirms the usage of the item.

Update Administered Item [[updateItem](#)]

Description:

A submitter or steward updates annotations associated with an administered item. The item itself is not updated. Item update is handled by versioning.

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#Steward\]](#) - The steward selects an administered item for updating its annotation.
3. [\[#OOR\]](#) - A template for administered item annotations with known values filled in is given to the steward. A GUI may present this using a wizard.
4. [\[#Steward\]](#) - The steward modifies values of the administered item annotations.
5. [\[#OOR\]](#) - The system returns a confirmation message and a new version number for the administered item annotations.

Upload Administered Item Version [**versionItem**]

Description:

A submitter or steward creates a new version of an existing item.

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#Submitter\]](#) - The submitter selects an administered item as the previous version of the item.
3. [\[#Submitter\]](#) - The submitter uploads an administered item.
4. [\[#OOR\]](#) - The system returns a confirmation message and a new version number for the administered item.
 - a. Alternative Flow: [\[#mapVersion\]](#)

Upload Administered Item Map [**mapItems**]

Description:

A submitter registers a mapping from one item to another item.

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#Submitter\]](#) - The steward selects two administered items for the mapping.
3. [\[#OOR\]](#) - A template for the mapping is given to the submitter. A GUI may present this using a wizard.

4. [\[#Submitter\]](#) - The submitter modifies values of the mapping annotations and uploads a mapping from one item to the other item.
5. [\[#OOR\]](#) - The system confirms that the mapping has been uploaded.

Compose Administered Items [[composeltems](#)]

Description:

A submitter composes a number of ontology modules or contexts and registers the composition. One would expect that an application will use more than one ontology. In many cases one can define the composition by importing ontologies and adding additional axioms (such as sameAs relationships). However, there are cases where this is not possible, so a use case is available for more complex compositions.

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#User\]](#) - The user selects administered items, usually ontology modules, contexts and mappings.
3. [\[#OOR\]](#) - A template for the composition is given to the user. A GUI may present this using a suitable graphical editor.
4. [\[#User\]](#) - The user defines the compositional diagram.
5. [\[#OOR\]](#) - The system stores the compositional diagram and returns a composition identifier.

Process Instance Form Completion [processInstanceForm]

Description:

A user needs to input some information as required by one step in a process definition.

Step-by-step Description:

1. [\[#OOR\]](#) - The system notifies a process participant that some information needs to be provided for a process associated with an administered item.
2. [\[#ProcessParticipant\]](#) - The user queries the system for the list of tasks to be completed.
3. [\[#OOR\]](#) - The system returns all the pending tasks for the user.
4. [\[#ProcessParticipant\]](#) - The user select one of the pending tasks.
5. [\[#OOR\]](#) - A template for the information required by the task is given to the user. A GUI can present this to the user using a wizard.
6. [\[#ProcessParticipant\]](#) - The user submits the information required for the task.
7. [\[#OOR\]](#) - The system returns a confirmation message.

Update Contact Information [updateContact]

Description:

A user changes contact information. This can include retiring from the OOR.

Step-by-step Description:

1. Includes: [\[#queryContact\]](#)
2. [\[#User\]](#) - The user submits new contact information (i.e., email address, name or surname).
3. [\[#OOR\]](#) - The system returns a confirmation message.

Review an administered item [[reviewItem](#)]

Description:

A user informally reviews the administered item, for example, by providing comments and number rating. Formal reviews and evaluations are performed as part of a process. This use case is only concerned with ad hoc informal reviews.

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#User\]](#) - The user selects an item and submits a comment and/or a rating for a particular version of an administered item.
3. [\[#OOR\]](#) - The system records the information and sends the user a confirmation message.

Query reviews [[retrieveReviews](#)]

Description:

The user wants to see reviews and evaluations for an item. This use case deals only with informal reviews. Formal reviews are part of a process and are normally confidential.

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#User\]](#) - The user selects an administered item and specifies which reviews or evaluations are to be returned.
3. [\[#OOR\]](#) - The system returns the requested reviews or evaluations.

Advertise Administered Items [[advertise](#)]

Description:

The user retrieves a list of some of the administered items that are to specially advertised. This differs from a query because the query is OOR generated rather than user generated.

Step-by-step Description:

1. [\[#User\]](#) - The user submits a request to see the advertised administered item (e.g., when opening the home page of the system).
2. [\[#OOR\]](#) - The system returns the advertised administered item using some OOR instance-specific criteria (e.g., the items which have a STANDARD status and a high rating).

Retire Process Participant [retireProcessParticipant]

Description:

A process participant desires to retire from their system responsibilities.

Step-by-step Description:

1. [\[#ProcessParticipant\]](#) - The process participant submits a request to retire from the system.
2. [\[#OOR\]](#) - The system responds with a confirmation message.
 - a. Alternative Flow: [\[#reassignProcessParticipant\]](#)

Query procedures [queryProcedures]

Description:

The user wants to retrieve the procedures regarding the standardization process. This is only relevant for users who might participate in a process.

Step-by-step Description:

1. [\[#ProcessParticipant\]](#) - The user submits a request to see the procedures that satisfy specified criteria.
2. [\[#OOR\]](#) - The system returns all the procedures that satisfy the criteria.

Query metrics [queryMetrics]

Description:

The user wants to see metrics about an administered item. Examples include:

1. number of statements (terms, relationships) in an item
2. number of statements referencing a given URI
3. number of revisions to a given context or statement
4. number of users/organizations using the item
5. number of contexts/terms accessed (served)
6. number of times a given context or term has been accessed

Step-by-step Description:

1. Includes: [\[#queryItem\]](#)
2. [\[#User\]](#) - The user selects an administered item and requests its metrics.
3. [\[#OOR\]](#) - The system returns the metrics for the item.

Query Contact Information [[queryContact](#)]

Description:

The user wants to retrieve contact information for users. This information is normally confidential so access constraints will be enforced.

Step-by-step Description:

1. **[#User]** - The user submits a request for the contact information of users satisfying search criteria.
2. **[#OOR]** - The system returns contact information relevant to the query and satisfying access constraints.

Alternative Flows:

Notify Registrar [**notifyRegistrar**]

Preconditions: The validation task is relevant to the registrar as specified in the relevant policies.

Description:

The registrar is notified of the result of a validation task.

Step-by-step Description:

1. **[#Registrar]** - The system notifies the registrar of the result of a validation task.

Map Administered Item Version [**mapVersion**]

Preconditions: The new version of the administered item is backwardly compatible with the previous version.

Description:

A mapping is defined for a new version of an administered item.

Step-by-step Description:

1. **[#Submitter]** - The submitter uploads a mapping from the previous version to the new version.

2. [OOB] - The system confirms that the mapping has been uploaded.

Reassign Process Participant [reassignProcessParticipant]

Preconditions: The user is a process participant.

Description:

A process participant has retired and any responsibility must be taken over by some other process participant.

Step-by-step Description:

1. [OOB] - The system sends a request to other process participants (i.e., registrars, stewards and submitters) in the same registration authority and at the same level (i.e., steward, registrar or submitter). The request is to take over the responsibility of the process participant that has retired, with a request number.
2. [ProcessParticipant] - A user sends the petition number and an acknowledgement that is willing to accept the petition. We will call this user the candidate.
3. [OOB] - The system informs the immediate superior that a candidate exists for a user to accept the petition.
4. [ProcessParticipant] - The immediate superior accepts the candidate.
5. [OOB] - The system informs the candidate, that it has now inherited the responsibilities of the retired user.

DESIGN: © COPYRIGHT 2008 BLACKER

DESIGN BY REALITY SOFTWARE AND MAXIMO GURMENDEZ