

# Workflow - Modélisation

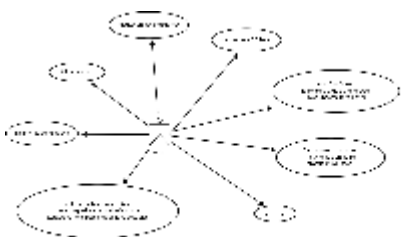
## Modélisation

Cette section présente une sorte de première analyse de Ori-Oai-Workflow. Même si par rapport à l'implémentation qui en a été faite, cette modélisation reste très conceptuelle, elle permet de mieux appréhender l'architecture de Ori-Oai-Workflow.

## Cas d'Utilisations

Ces différents cas d'utilisation seraient à affiner et compléter.

Pour une étude fonctionnelle poussée (et non pas une approche technique) des cas d'utilisations, référez-vous au document décrivant les cas d'utilisations "modèle" (auteurs: Rosa María Gómez de Regil et Romuald Lorthoir)



## Diagrammes de séquence

Dans les diagrammes de séquence présentés ci-dessous, on constate que OsWorkflow dirige l'application : en fonction de la configuration du workflow actif et de l'action demandé par l'utilisateur ("initialiser un workflow", "valider la publication", "rendre privé", "demander la validation technique", etc), de l'état dans lequel se trouve l'instance de Workflow (donc - la fiche de métadonnées), OSWorkflow va appeler les méthodes métiers d'Ori-Oai-Workflow pour vérifier les conditions ou effectuer les fonctions adéquates.

### Afficher la liste des entrées.

L'utilisateur demande à afficher les différentes entrées dans lesquelles il peut jouer un rôle sur une instance de workflow.



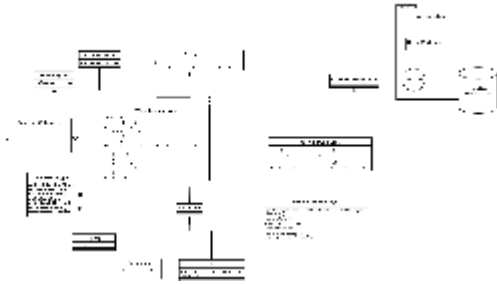
### Créer/Modifier une instance de Workflow

L'utilisateur demande à créer/modifier une fiche de métadonnée et donc une instance de Workflow.



## Diagramme de classes

Ce diagramme de classes se veut simple pour une bonne compréhension de l'ensemble. Il ne présente pas un certains nombres d'attributs et surtout de classes qui sont d'ordre architectural (architecture par couche : données/dao/domain/web). De plus certaines associations n'ont pas été implémentées directement comme des associations POJO/Java : ce sont plus des associations conceptuelles établies par les configurations Spring et les services métiers.



Quelques Explications sur certaines classes :Entry

Correspond à un ensemble d'instances de Workflows. Les instances de Workflows sont dans une même Entrée car elles décrivent la même entité. Elles sont distinctes car elles peuvent correspondre à des versions différentes, décrivent l'entité avec des schémas différents.

WorkflowInstance

Une instance de Workflow est défini par son contenu XML et son osworkflow (instance d'un OSWorkflow) associé (idworkflow + workflowName).  
Un workflowInstance représente donc La donnée principale autour de laquelle gravite l'application Spring.

MetadataType

xpathTitle est le XPath correspondant au titre d'un XML répondant au schéma présent.  
Les différents MetadataType représentent les différents types de fiches de métadonnées que support l'application. Ils sont à configurer dans un fichier de config spring spécifique (cf la section Configuration de ce présent document).  
Concrètement, les différents types de fiches sont les différents types de fiches que pourra ajouter l'utilisateur (en fonction des droits qu'il a et donc des conditions des workflows spécifiés dans le Metadata Type) :

- Ajouter une thèse
- Ajouter une ressource pédagogique
- etc.

RoleStepCategory

Les différents RoleStepCategory représentent les différentes catégories de fiches (~ dossiers virtuels) listées à l'utilisateur. Ils permettent de lister dans un même ensemble des fiches de workflows différentes dans des états différents et sur lesquels l'utilisateur a des rôles différents. Ainsi on peut par exemple présenter à l'utilisateur toutes les fiches dont il est propriétaire qui sont dans l'états creation ou saisie\_auteur issues des workflows model1 et easy par exemple.  
Ils sont à configurer dans un fichier de config spring spécifique (cf la section Configuration de ce présent document).

Les permissions sont à la fois positionnées par OSWorkflow et par Spring lors de l'initialisation des "permissions par défaut".