

ORI-OAI version 1.0

Installation et configuration



Plan

I.	Description du document	5
A.	Technologies employées au cours de ce support.....	5
B.	Pré-requis	5
II.	Fonctionnalités de ORI-OAI version 1.0	6
A.	Gestion et publication des documents	6
1.	Référentiel unique de documents, connecté au SI	6
2.	Production de métadonnées et diffusion des ressources produites par l'établissement	6
3.	Classification pivot.....	8
B.	Partage des ressources numériques avec d'autres établissements.....	9
1.	Exposition des fiches de métadonnées des ressources de l'établissement	9
2.	Agrégation.....	9
C.	Accès aux ressources numériques à distance en fonction des droits d'accès	10
1.	Système de recherche	10
2.	Définition des droits	10
III.	Architecture de la version 1.0	11
A.	Présentation	11
B.	Le rôle de chaque module	11
1.	ORI-OAI-workflow.....	11
2.	ORI-OAI-indexing.....	12
3.	ORI-OAI-repository	12
4.	ORI-OAI-harvester	13
5.	ORI-OAI-search	13
6.	ORI-OAI-vocabulary	14
7.	ESUP-serveur-WebDAV.....	14
IV.	Installation	16
A.	Pré requis et variables d'installation	16
B.	Installation des serveurs tomcat	17
C.	Subversion	22
D.	Modules communs et obligatoires quelque soit la configuration	22
1.	ori-oai-indexing.....	22
2.	ori-oai-vocabulary	27
3.	ori-oai-search.....	28
E.	Agrégation de ressources	32
1.	ori-oai-harvester	32
2.	ori-oai-repository	36
F.	Productions de données	39



1.	ori-oai-md-editor	39
2.	ori-oai-workflow	42
3.	Solution de stockage ESUP Portail et dépôt des ressources	52
V.	Configurations avancées	57
A.	Ajout d'une métadonnée	57
1.	Ajout d'un vocabulaire statique dans ori-oai-vocabulary	57
2.	Saisie de la métadonnée dans ori-oai-md-editor	59
3.	Prise en compte du caractère obligatoire dans ori-oai-workflow	61
4.	Indexation et recherche dans ori-oai-indexing	61
5.	Recherche et affichage dans ori-oai-search	64
6.	Recherche avancée	64
7.	Affichage dans les résultats de recherche	65
8.	Affichage dans la notice.....	66
9.	Mise à jour des modifications	67
B.	Modification du Workflow	67
1.	Ajout de permissions/rôles dans ori-oai-workflow	69
2.	On modifie le workflow en 3 étapes	69
3.	On déclare un nouveau formulaire	71
4.	Création du formulaire oaidc-archive-full dans ori-oai-md-editor.....	72
5.	Affectation d'un rôle à un groupe, création d'un groupe virtuel	72
6.	Ajout d'une catégorie « Ressources publiées »	73
7.	Ajout d'une catégorie « Mes documents archivés »	73
VI.	Généralités	75
A.	Utilisation d'un frontal http Apache	75
1.	Configuration	75
2.	Protection des URL sensibles.....	75
3.	Quelles URL faut-il protéger ?	75
B.	Gestion de l'index	76
1.	Restauration de l'index des fiches de ori-oai-workflow	76
2.	Restauration de l'index des fiches de ori-oai-harvester	76
VII.	Evolutions en cours de réalisation	77
A.	Module ori-oai-workflow	77
B.	Moteur d'indexation	77
C.	Moteur de recherche.....	77
D.	Gestionnaire de vocabulaires	77
E.	Evolutions générales du projet.....	78
F.	Nouveaux types de ressources.....	78
VIII.	Liens.....	79
IX.	Annexes.....	81
A.	XML et les espaces de noms	81



1.	Rappel.....	81
2.	LOM	82
3.	LOMFR	82
4.	Conclusion	83
B.	Exploitation d'applications avec subversion - installation et mises à jour	83
1.	Checkout d'ori-oai-workflow 0.8.3	83
2.	Configurations d'ori-oai-workflow 0.8.3.....	84
3.	Mise à jour d'ori-oai-workflow : 0.8.3 -> 1.0.0	85
4.	Mise à jour d'ori-oai-workflow : conflits !	86
5.	Conclusion	88
C.	Ajout d'une métadonnée dans un formulaire d'ORI-OAI-md-editor	88
1.	Espace de noms Rennes1 et exemple d'XML LOM Rennes1	89
2.	Formulaire LOM Rennes1.....	89
3.	Prototype	90
4.	Modification du formulaire	90
5.	Termes i18n	92



I. Description du document

Ce document vous permettra une installation et une configuration pour une prise en main rapide de l'outil ORI-OAI. On y décrit diverses possibilités d'installation, de configuration et d'utilisation de ORI-OAI.

Toutefois, même si un maximum de notions est abordé dans ce support, vous n'y trouverez pas toutes les informations pour une configuration poussée des différents modules qui composent notre outil. Il est donc **fortement conseillé**, en plus de ce support, de consulter les documents décrivant des configurations plus avancées depuis la page « **Aller plus loin > Téléchargements** » du site du projet <http://www.ori-oai.org>. Vous trouverez dans ces documents en ligne toutes les informations nécessaires concernant les possibilités de configurations avancées et d'installation de ORI-OAI.

A. Technologies employées au cours de ce support

Notez que différentes technologies et notions sont installées tout au long de ce support pour l'installation et la configuration de ORI-OAI. Même si la connaissance de ces technologies n'est pas requise, il est préférable de les maîtriser en partie pour une installation plus rapide et avancée de ORI-OAI :

- Tomcat
- Ant
- XML
- Spring
- Environnement Linux

B. Pré-requis

Le projet ORI-OAI est entièrement développé dans un environnement J2EE. Aussi, le support qui vous est proposé ici se base entièrement sur un environnement Linux pour se rapprocher au mieux d'une mise en production. Les notions citées ci-après sont donc nécessaires :

- Connaître l'environnement Linux
- Avoir déjà installé une application J2EE

II. Fonctionnalités de ORI-OAI version 1.0

ORI-OAI permet de gérer l'ensemble de la production numérique institutionnelle d'un établissement. Dans sa première version, l'application est plus particulièrement configurée pour gérer les ressources pédagogiques numériques.

A. Gestion et publication des documents

1. Référentiel unique de documents, connecté au SI

ORI-OAI permet de créer un référentiel unique des documents numériques d'un établissement : il permet de gérer les documents produits par l'établissement (documents scientifiques, pédagogiques, administratifs, documentaires...), mais aussi de proposer à travers des interfaces communes ou identifiées, des documents en provenance de fonds d'autres établissements (documents agrégés par moissonnage, voir le paragraphe « **Partage des ressources numériques avec d'autres établissements > Agrégation** »).

Il est connecté au Système d'Information (SI) de l'établissement :

- à travers l'annuaire LDAP et le système CAS de l'établissement pour permettre un accès sécurisé et qui est fonction du rôle de l'utilisateur lorsque celui-ci souhaite accéder au dépôt ou à l'indexation d'une ressource,
- à travers l'ENT de l'établissement pour proposer un accès personnalisé aux interfaces de recherche et aux ressources,
- à travers le système Shibboleth pour permettre l'accès à des ressources extérieures à l'établissement.

2. Production de métadonnées et diffusion des ressources produites par l'établissement

ORI-OAI propose une gestion de workflow pour la création des métadonnées liées aux ressources et la diffusion de ces ressources et de leurs métadonnées. Chaque workflow va permettre de gérer un circuit de production de métadonnées et de diffusion d'un type de ressource. Les workflows créés permettent :

- de définir une suite d'**états** que la ressource va pouvoir emprunter jusqu'à sa publication,
- d'attribuer des droits d'accès (**permissions**) à ces états à des groupes d'utilisateurs selon des **rôles** définis,
- de définir les changements d'état possibles (**transitions**) et sous quelles **conditions** ils peuvent avoir lieu
- et de définir des événements qui se dérouleront lors d'un changement d'état (un envoi de mail par exemple) (**fonctions**).

Des éditeurs peuvent être liés à tel ou tel état pour permettre la saisie de métadonnées ou d'informations parallèles au workflow.

L'application est entièrement paramétrable. Elle a cependant été pré configurée selon des besoins simples identifiés. Deux workflows sont ainsi proposés avec leur paramétrage adapté : le premier propose de manière générique un workflow en deux étapes :

- **état privé** : saisie des métadonnées en format DublinCore
- **état public** : les métadonnées et les ressources sont diffusées

Un deuxième workflow est proposé pour les ressources pédagogiques, en trois étapes :

- **état privé** : correspond à un groupe d'utilisateurs « auteur », il permet la saisie d'un ensemble de métadonnées
- **état en cours de validation** : correspond à un groupe d'utilisateurs « modérateur », il donne accès à l'ensemble des métadonnées de description des ressources pédagogiques, avec pour les champs communs aux champs proposés à l'auteur, les valeurs que celui-ci a déjà saisies
- **état publié** : les métadonnées et les documents sont diffusés

Un troisième workflow correspondant au traitement complexe de la diffusion des ressources pédagogiques a été pensé et est présent dans l'application. Sa configuration n'est cependant pas terminée.



ORI-OAI permet de proposer à travers un site web (intranet, internet, portail...) une visualisation des fiches des métadonnées produites et agrégées dans l'application et de donner un accès aux ressources à partir de cette visualisation.

i. Le Dublin Core

« La norme de métadonnées du Dublin Core est un ensemble d'éléments simples mais efficaces pour décrire une grande variété de ressources en réseau. La norme du Dublin Core comprend 15 éléments dont la sémantique a été établie par un consensus international de professionnels provenant de diverses disciplines telles que la bibliothéconomie, l'informatique, le balisage de textes, la communauté muséologique et d'autres domaines connexes.

Couverture	Collaborateur	Date
Description	Créateur	Format
Type	Éditeur	Identifiant
Relation	Droits	Langue
Source		
Sujet		
Titre		

Chaque élément est optionnel et peut être répété. Chaque élément possède également un ensemble limité de qualificatifs, des attributs qui peuvent être utilisés afin de raffiner davantage (et non pas étendre) la signification de l'élément. L'Initiative de métadonnées du Dublin Core (IMDC) a défini, en juillet 2000, des façons normalisées de "qualifier" les éléments au moyen de différents types de qualificatifs. Un registre de qualificatifs conformes aux "meilleures pratiques" de l'IMDC est en cours de construction. »

(Source: Guide d'utilisation du Dublin Core, par Diane Hillmann, traduction de Guy Teasdale, Université Laval (2001), <http://www.bibl.ulaval.ca/DublinCore/usageguide-20000716fr.htm>)

ii. Le LOM

« Le LOM est un standard international proposant un modèle de description des métadonnées associées à des objets pédagogiques quels qu'ils soient, numériques ou non. Le LOM n'est pas une norme mais une recommandation de métadonnées. Il a été adopté par les IEEE (<http://ltsc.ieee.org/wg12/>) en 2002. »

Il propose un jeu de métadonnées découpé en 9 catégories :

- général
- cycle de vie
- méta métadonnées
- technique
- pédagogique
- relations
- droits
- commentaires
- classification

« Le LOMFR est le profil français d'application du LOM. Contrairement au LOM, c'est une norme. Le LOMFR a été élaboré dans le cadre d'une convention entre l'Education nationale et l'AFNOR. Il a été procédé à un appel à commentaires du profil d'application de novembre 2005 à mars 2006. Ce profil a évolué, suite à l'appel à commentaires. Une version définitive du LOMFR a été publiée en décembre 2006.

Le LOMFR décrit des objets (ressources) pédagogiques. Est considérée comme ressource pédagogique toute entité (numérique ou non) utilisée dans un processus d'enseignement, de formation ou d'apprentissage. Il peut s'agir de livres, de films, de scénarios pédagogiques, de sites web, de logiciels, etc.

Ses objectifs sont les suivants :

- permettre une meilleure visibilité de l'offre d'objets pédagogiques, numériques ou non,
- disposer de normes à respecter pour garantir la pérennité, l'interopérabilité, et une certaine ouverture des ressources numériques produites, en lien avec les plates-formes de distribution et de consultation de ces ressources. (extrait de la norme expérimentale).

Il concerne l'ensemble de la communauté éducative :

- communauté éducative française (enseignements primaire, secondaire, supérieur)

- communauté de la formation (formation professionnelle, formation continue)
- documentalistes et bibliothécaires
- éditeurs, entreprises. »

(Source: site EducNet - <http://www.educnet.education.fr/dossier/metadata/lom1.htm>)

« A la demande de la SDTICE, le groupe de travail « SUP LOMFR » s'est réuni la première fois le 18 janvier 2007. Il est animé par l'UNT « UNIT ». Il est composé de représentants des UNT, des UNR, des Etablissements, et rassemble des compétences transversales : informatiques, documentaires et pédagogiques.

Il a pour objectif :

- De définir les éléments de la norme LOMFR jugés indispensables au référencement des ressources pédagogiques, de manière à s'adapter au contexte et aux besoins spécifiques des établissements de l'enseignement supérieur français,
- De définir des vocabulaires propres aux usages de l'enseignement supérieur,
- De rédiger un guide d'usage spécifique pour préconiser les recommandations d'utilisation au sein des établissements d'enseignement supérieur. Ce guide sera agrémenté d'exemples destinés à faciliter la mise en œuvre du LOMFR. »

(Source: site EducNet - <http://www2.educnet.education.fr/sections/superieur/unt/inter/lom>)

iii. Configuration pour la gestion de ces formats dans la V1.0.0

Pour la production de métadonnées en Dublin Core et en LOM dans ORI-OAI, des éditeurs ont été développés pour ces deux formats et sont proposés à l'utilisation.

Le Dublin Core est proposé dans le workflow le plus simple, en deux états, pour l'indexation des ressources autres que les ressources pédagogiques.

Pour ces dernières, le LOM est proposé sous la forme de deux nouveaux éditeurs dans le workflow à trois états :

- le premier propose une sélection de champs du format LOM qui sera proposée à compléter par l'auteur,
- tandis que le deuxième propose l'ensemble des champs du format LOM en ajoutant des contraintes liées aux besoins de l'application et aux travaux français sur le format LOM qui se concrétise dans la norme LOMFR et dans les travaux du groupe SupLOMFR.

A la publication des ressources, les métadonnées remplies sont indexées pour pouvoir être utilisées dans les recherches et dans les échanges avec d'autres établissements.

3. Classification pivot

ORI-OAI permet l'indexation des documents à l'aide de tout type de classification (Dewey, CDU, classification « locale »...).

Une classification « locale » peut être créée par un établissement ou un groupe d'établissements en fonction des besoins. Elle correspond à une classification sélective et personnalisée, dans un domaine particulier. Elle permet de proposer aux utilisateurs un langage qui correspond plus à leurs connaissances « métier ».

Pour permettre l'interopérabilité en dehors de ce contexte d'utilisation particulier, il peut être utile de lier cette classification « locale » à une classification générique de référence, dite « pivot ». Une table de correspondance reliant les indices de la classification « locale » aux indices de la classification « pivot » doit être créée.

ORI-OAI supporte cette organisation de classifications. Par exemple, l'application peut proposer :

- la classification « locale » aux utilisateurs non bibliothécaires lors de l'indexation du document,
- la classification « pivot » (par exemple la classification Dewey) aux utilisateurs bibliothécaires pour cette même indexation,
- la navigation dans l'une ou l'autre de ces classifications pour accéder aux documents dans le moteur de recherche.

Dans le cas de l'utilisation de ce système classification « locale » / classification « pivot », les fiches de métadonnées peuvent être indexées avec la classification « locale », mais elles devraient l'être impérativement avec la classification « pivot ».

Seule la classification « pivot » est mémorisée lors de l'intégration des données dans les index de l'application. Lors de toute utilisation, ce sont les indices pivots qui seront retranscrits pour être présentés dans la classification « locale » à l'aide de la table de correspondance.

La Classification Décimale Dewey (CDD) est la classification utilisée comme pivot pour élaborer la classification UNIT, elle est également utilisée pour les classifications de Valenciennes. Elle a été choisie car :

- il s'agit d'une classification internationale préconisée pour les bibliothèques universitaires par la sous-direction des bibliothèques (MESR). Elle est utilisée dans un très grand nombre d'établissements français, et permet donc les échanges et l'interopérabilité avec d'autres applications françaises et étrangères,
- le groupe SupLOMFR recommande l'utilisation de la classification Dewey comme indexation minimum des ressources pédagogiques,
- elle couvre l'ensemble des domaines de la connaissance : elle permet d'avoir une classification unique comme pivot pour l'ensemble des documents publiés et/ou moissonnés dans un établissement ou un regroupement d'établissements,
- c'est une classification arborescente avec un code numérique, ce qui permet des manipulations plus aisées avec l'utilisation par exemple des tronçatures pour inclure une partie de l'arborescence, à partir d'un nœud identifié par son code. Cela permet également de rendre indépendante la valeur saisie de l'affichage qui en est fait (par exemple pour gérer les traductions),
- elle continue d'être mise à jour avec l'évolution des connaissances (on en est à la 22ème édition).

En ce qui concerne la maintenance de ces classifications, les classifications locales peuvent évoluer sans que cela pose problème puisque l'indexation prise en compte est celle effectuée dans la classification pivot. Du côté de la CDD, cette classification évolue en fonction des évolutions scientifiques et techniques. En général les modifications sont des ajouts en fin d'embranchement, mais il peut arriver qu'un élément soit supprimé. Ce cas est plus difficile à gérer, mais il est aussi plus rare.

Remarque : une autre classification peut être choisie comme pivot.

B. Partage des ressources numériques avec d'autres établissements

1. Exposition des fiches de métadonnées des ressources de l'établissement

En se basant sur le protocole OAI-PMH, ORI-OAI propose également la création d'un entrepôt pour exposer les métadonnées créées au sein de l'application. Cet entrepôt pourra ainsi être moissonné par une application utilisant le protocole OAI-PMH.

ORI-OAI permet la création de sets de diffusion pour les données proposées dans l'entrepôt : en effet, l'application permet de définir des ensembles de documents en découpant le fonds selon les valeurs données à une métadonnée particulière. Par exemple un set pourra être proposé par auteur ou par domaine pédagogique. Cela permet une sélection plus précise pour les moissonneurs des documents répondant à leur demande que de moissonner l'ensemble du fonds.

2. Agrégation

ORI-OAI se base sur le protocole OAI-PMH pour permettre d'« acquérir » des fiches de métadonnées de ressources produites par d'autres établissements et de les agréger dans son propre fonds. On parle de moissons qui sont effectuées sur des entrepôts de documents. Ces moissons peuvent être programmées pour avoir lieu de manière régulière. Dans ce cas, une vérification sera faite sur les nouveaux documents proposés par l'entrepôt enregistré.

L'agrégation s'effectue sur les métadonnées et non sur les ressources elles-mêmes qui restent dans le fonds de l'établissement qui les a produites.

Le protocole OAI-PMH demande à ce que les métadonnées proposées dans les entrepôts soient au moins présentes dans le format Dublin Core, mais il permet l'échange de n'importe quel autre format de métadonnées.

ORI-OAI peut ainsi moissonner tout format de métadonnées présenté dans un entrepôt OAI-PMH. Par contre pour indexer et diffuser ces métadonnées, des configurations de l'application doivent être faites. Dans sa première version, l'application est configurée pour moissonner des entrepôts proposant du Dublin Core et du LOM, mais aussi du CDM (Course Description Metadata) qui est « un standard européen de description de programmes d'études et de cours ».



(Source : site EducNet - http://www2.educnet.education.fr/sections/superieur/soutien/normes/juillet_2005_publi).

Le partage des ressources numériques entre les établissements va permettre

- d'enrichir l'offre numérique de chacun,
- d'augmenter la visibilité des ressources en les rendant accessibles depuis différents sites et au travers des réseaux auxquels appartient l'établissement,
- et de mutualiser les coûts en permettant la réutilisation des ressources dans d'autres contextes que celui identifié pour leur création et en travaillant à partir des besoins communs pour réaliser de nouvelles ressources.

C. Accès aux ressources numériques à distance en fonction des droits d'accès

1. Système de recherche

ORI-OAI propose de créer différents types de moteurs de recherche sur les métadonnées des documents proposées à la diffusion. Ces documents peuvent être des publications internes ou des documents agrégés.

L'application propose plusieurs moteurs déjà configurés.

- moteur de recherche simple et avancé

Un moteur de recherche simple peut être proposé. Il s'agit d'un champ textuel unique permettant de rechercher sur une ou plusieurs métadonnées en saisissant une série de termes (en suivant le même principe que tous les moteurs de recherche bien connus sur la toile).

Un moteur de recherche avancée peut également être proposé. Il permet d'interroger différentes métadonnées en attribuant des champs différents à chacune. Cela permet entre autre de préciser les valeurs d'un champ en proposant par exemple une liste particulière de valeurs ou d'attribuer un fonctionnement particulier à un champ pour la sélection d'une valeur (par exemple le choix d'un élément dans une classification).

- recherche thématique

Le moteur de recherche thématique permet de parcourir une arborescence et de sélectionner une entrée « finale » particulière pour visualiser les documents qui sont indexés avec cette entrée. Cela permet de parcourir une classification arborescente comme la Classification Décimale Dewey ou une liste alphabétique de termes comme la liste des établissements moissonnés, la liste des auteurs...

- nouveautés

La liste des nouveautés peut également être proposée comme une entrée particulière du moteur de recherche.

Ces différents moteurs sont proposés pré configurés dans l'application pour les formats Dublin Core, LOM et CDM. Il est possible de créer un moteur particulier pour un type de document présent dans l'application (les métadonnées LOM liées aux ressources pédagogiques sont plus développées que les métadonnées Dublin Core par exemple) ou de créer un moteur générique qui pourra interroger toutes les ressources, quel que soit le schéma utilisé pour les indexer.

Les moteurs de recherche peuvent être intégrés à l'ENT de l'établissement pour répondre au mieux aux besoins des utilisateurs.

2. Définition des droits

ORI-OAI propose à un auteur de définir des droits d'accès sur ses ressources lorsqu'il les dépose.

Suite à une consultation, lorsqu'un utilisateur souhaitera accéder à une ressource, une vérification sera faite pour voir s'il en a le droit et de quel accès il dispose (consultation uniquement ou écriture/modification...).

ORI-OAI permet de gérer des droits pour des groupes d'utilisateurs, en se basant sur l'annuaire LDAP de l'établissement et sur le système Shibboleth.

III. Architecture de la version 1.0

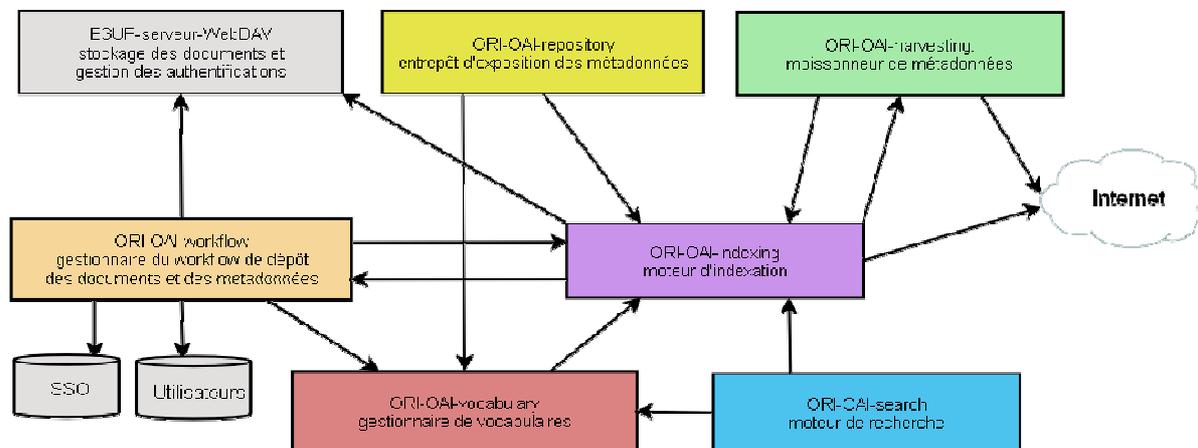
A. Présentation

L'architecture du projet ORI-OAI se présente sous la forme de sept modules développés indépendamment. Chacun de ces modules a un rôle bien défini dans le système et communique avec les autres au travers de Web Services exposés en frontal de chaque composant.

Ce choix permet une grande souplesse dans d'éventuelles déclinaisons d'architectures et de langages de programmation. En effet, cette architecture technique a été notamment pensée pour permettre à des logiciels extérieurs de dialoguer avec différents éléments du système. Nous pouvons par exemple prendre le cas d'une bibliothèque qui souhaiterait faire des recherches de documents depuis son portail documentaire sans passer par l'interface de recherche ORI-OAI-search. Il est donc possible qu'il y ait une connexion directe entre leur logiciel et l'index de recherche ORI-OAI-indexing.

B. Le rôle de chaque module

La figure suivante montre l'ensemble des modules et leurs interconnexions :



Voyons maintenant en détail le rôle de chaque module qui compose le système ORI-OAI :

1. ORI-OAI-workflow

Module **optionnel** dans la version 1.0

Dans quels cas l'utiliser : pour la saisie et le dépôt de fiches de métadonnées dans le système local

Modules obligatoirement pré requis :

- **ORI-OAI-vocabulary** pour fournir les vocabulaires lors de la saisie des métadonnées
- **ORI-OAI-indexing** pour indexer les fiches de métadonnées saisies
- Base de données **eXist** pour le stockage des fiches de métadonnées
- Base de données **MySQL**
- Un annuaire **LDAP** pour la gestion des utilisateurs

Modules optionnels suivant le cas d'utilisation :

- **ESUP-serveur-WebDAV** si vous voulez utiliser ce serveur pour y déposer les ressources
- **Canal stockage ESUP Portail** (ou tout autre client WebDAV) pour déposer les ressources sur le serveur ESUP-serveur-WebDAV
- Un serveur SSO **CAS** pour l'authentification (authentification LDAP si CAS non utilisé)

Description :

Ce composant est utilisé pour la saisie des métadonnées. Les technologies utilisées par ce moteur de publication sont diverses et permettent un paramétrage très fin, complet et puissant en fonction des besoins de l'établissement ou du consortium qui le met en place.



Il sera aussi utilisé dans la version 2 pour le dépôt des documents dans le système et l'affectation des droits d'accès aux ressources. En attendant cette version, il est possible uniquement de renseigner le lien vers le(s) document(s) depuis l'interface de saisie des métadonnées ; le dépôt devant se faire en dehors du workflow. Dans le cas où vous utilisez le serveur WebDAV pour gérer vos documents, il est donc nécessaire d'utiliser n'importe quel client compatible avec la norme WebDAV en dehors du workflow pour déposer la ressource et copier le lien vers cette ressource dans la fiche de métadonnées. Pour les établissements utilisant l'ENT ESUP Portail, vous avez la possibilité d'utiliser le canal stockage (<http://www.esup-portail.org/Projets/Fichedescriptive/?contentId=183>) pour faire le dépôt de la ressource, gérer les droits d'accès (lecture/écriture pour des usagers, groupes d'usagers de votre établissement ou externes en utilisant la fédération d'identités avec shibboleth).

Le moteur de workflow OsWorkflow permet une configuration avancée de toutes les actions effectuées à chacune des étapes de la publication. En effet, par de la configuration, il peut répondre aux besoins des établissements exigeant du déposant la saisie de tous les champs de métadonnées, mais aussi des établissements plus structurés où la saisie des métadonnées s'effectue en différentes étapes, par différents intervenants. Chaque établissement peut ainsi configurer le module suivant ses besoins.

Les formulaires de saisie des métadonnées sont eux aussi configurables. La technologie XForms associée au moteur de transformation Orbeon offre des formulaires de saisie dynamiques en fonctions de fichiers de configuration écrits en XML. Le support des différents formats de métadonnées repose donc sur la possibilité d'enrichir le module ORI-OAI-workflow de nouvelles configurations XForms.

Les autres technologies utilisées dans ce module sont Acegi Security pour gérer toutes les sécurités d'accès à l'application, eXist comme base de données XML Open Source pour le stockage des fiches de métadonnées, Hibernate pour rendre les accès à la base de données relationnelle transparents et enfin JSF pour la gestion des interfaces graphiques.

2. ORI-OAI-indexing

Module **obligatoire** dans la version 1.0

Description :

Une fois le dépôt de ressources et la saisie de métadonnées validés, ceux-ci sont indexés par le module ORI-OAI-indexing. Ce module a pour rôle l'indexation des fiches de métadonnées ainsi que des documents associés.

Pour cela, il utilise le moteur d'indexation Lucene. Celui-ci permet l'indexation de différentes sources offrant une recherche puissante et rapide en se reposant sur différents analyseurs. L'analyseur de la langue française permettra notamment la gestion des verbes conjugués, des pluriels ou encore des accents et caractères spéciaux. Un système de pondération permet aussi de rendre une métadonnée plus pertinente qu'une autre. Par exemple, on préférera retrouver en premier les documents dont l'élément recherché se trouve dans le titre plutôt que dans la description.

Lius est un framework d'indexation basé sur le projet Lucene. Il permet une indexation de différents formats de fichiers comme XML, PDF, OpenOffice, ZIP, MP3, etc. Il est utilisé dans notre projet pour offrir une configuration avancée des champs à indexer dans les différents formats de fiches de métadonnées XML et, par la suite, pour indexer les documents associés en plein texte.

En plus de l'aspect indexation, ORI-OAI-indexing offre un service de recherche de documents via Web service en se reposant sur la syntaxe des requêtes Lucene. Il est utilisé par différents composants dans le système.

3. ORI-OAI-repository

Module **optionnel** dans la version 1.0

Dans quels cas l'utiliser : pour exposer via le protocole OAI les fiches de métadonnées locales ou provenant de la moisson d'autres entrepôts

Modules obligatoirement pré requis :

- **ORI-OAI-indexing** pour la recherche des fiches de métadonnées à exposer

Modules optionnels suivant le cas d'utilisation :



- **ORI-OAI-vocabulary** pour créer des « sets OAI » en fonction de différents vocabulaires
- **ORI-OAI-workflow** pour exposer les fiches de métadonnées locales
- **ORI-OAI-harvester** pour exposer les fiches de métadonnées provenant de moissons OAI

Description :

Le module ORI-OAI-repository se charge, via le protocole OAI-PMH, de l'exposition des fiches de métadonnées saisies dans le module ORI-OAI-workflow et/ou de celles provenant de moissons OAI. Utilisant le logiciel OAICat, il expose les fiches dans le but d'être moissonnées par tout moissonneur OAI.

Ce module gère également le concept de « sets OAI-PMH ». Cet aspect du protocole OAI permet d'exposer les fiches de métadonnées sous forme d'ensembles distincts. Ces ensembles sont souvent liés à une thématique particulière. Nous pouvons par exemple identifier l'ensemble de toutes les fiches pédagogiques au format LOM associées aux notions de mathématiques. Pour identifier les fiches correspondant aux différents ensembles, le module ORI-OAI-repository construit des requêtes suivant les critères associés aux différents sets et les envoie au module ORI-OAI-indexing.

4. ORI-OAI-harvester

Module **optionnel** dans la version 1.0

Dans quels cas l'utiliser : si vous voulez construire un index à partir de fiches de métadonnées distantes récupérées via le protocole OAI

Modules obligatoirement pré requis :

- **ORI-OAI-indexing** pour indexer les fiches de métadonnées saisies
- Base de données **eXist** pour le stockage des fiches de métadonnées

Description :

Ce composant du système correspond au moissonneur OAI-PMH. Utilisant le logiciel OAIHarvester2, il permet le moissonnage de fiches de métadonnées sur tout entrepôt OAI. Les fiches moissonnées sont alors stockées localement dans une base de données XML eXist.

Tout comme le moteur de workflow, ce module fournit toutes les fiches de métadonnées moissonnées au moteur d'indexation dans le but d'être indexées. L'index de recherche Lucene est alors composé des documents locaux, mais aussi des fiches de métadonnées moissonnées par ce module.

Via une interface graphique conviviale, l'administrateur du système peut programmer les différentes moissons qui seront lancées par le gestionnaire de tâches Quartz.

5. ORI-OAI-search

Module **obligatoire** dans la version 1.0

Modules obligatoirement pré requis :

- **ORI-OAI-indexing** pour faire des recherches sur l'index
- **ORI-OAI-vocabulary** pour la construction des interfaces de recherche et la traduction de certains termes dans les résultats

Description :

Ce module offre une interface graphique pour la recherche de documents dans le système. Dialoguant avec le module ORI-OAI-indexing, il génère des requêtes dans la syntaxe Lucene et affiche les documents retrouvés.

Ce composant est entièrement configurable en ce qui concerne les formats de documents que l'on souhaite rechercher, et les types de recherches que l'on veut proposer à l'utilisateur. Il existe trois types de recherche :

- **Par date** : on propose à l'utilisateur de rechercher des documents suivant leur date de création, modification, etc. Ce type est utilisé pour afficher les nouveautés.
- **Avancée** : il est possible de configurer différents formulaires de recherche avec des critères plus ou moins avancés. On pourra proposer par un exemple un formulaire de recherche composé d'un seul champ



permettant une recherche sur le document complet et les métadonnées associées, ou encore un formulaire de recherche avancée où l'on pourra rechercher indépendamment sur chaque métadonnée d'un format donné.

- **Thématique** : dans ce type de recherche, on ne demande aucune saisie à l'utilisateur. Elle est mise en place pour faire des recherches suivant des classifications de documents, des auteurs, des mots-clefs, etc. Elle peut être sollicitée par les utilisateurs souhaitant découvrir les documents référencés dans le système n'ayant aucun critère de recherche particulier. Par ce type de recherche, on guide également l'utilisateur dans ses recherches en ne proposant que les valeurs réellement indexées comme par exemple dans le cas de la recherche par mots-clefs.

Notons également que le module peut être décliné en deux versions : *servlet* pour une installation standard sur un serveur Web et *portlet* pour une intégration dans un Environnement Numérique de Travail.

6. ORI-OAI-vocabulary

Module **obligatoire** dans la version 1.0

Description :

Le composant ORI-OAI-vocabulary est celui qui gère les vocabulaires utilisés dans différents modules. On entend par vocabulaire un ensemble fermé de valeurs disponibles pour un critère donné.

Les vocabulaires peuvent être statiques et configurés via des fichiers XML comme par exemple des classifications de documents ou des valeurs strictes de champs de métadonnées LOM. Ils peuvent aussi être dynamiques comme dans le cas de toutes les valeurs disponibles dans l'index d'une métadonnée précise. Par exemple, on pourra constituer dynamiquement, en interrogeant le module ORI-OAI-indexing, la liste des mots-clefs libres ou des auteurs qui ont déjà été saisis dans les documents pédagogiques.

Ces vocabulaires sont utilisés par :

- Le module ORI-OAI-workflow pour proposer des listes de valeurs lors de la saisie des métadonnées.
- Le moteur de recherche ORI-OAI-search pour les recherches thématiques ou les valeurs disponibles pour certains champs de la recherche avancée.
- L'entrepôt ORI-OAI-repository pour générer dynamiquement des sets OAI en fonction par exemple d'une thématique donnée.

7. ESUP-serveur-WebDAV

Module **optionnel** dans la version 1.0

Dans quels cas l'utiliser : pour la gestion et le stockage des documents locaux

Modules obligatoirement pré requis :

- Annuaire **LDAP**

Modules optionnels suivant le cas d'utilisation :

- **ESUP Portail** si vous utilisez cet ENT et que vous voulez utiliser le même gestionnaire de groupe dans l'ENT et le serveur WebDAV
- **Canal stockage** si vous voulez passer par votre ENT ESUP Portail pour déposer les ressources et gérer les droits d'accès
- Tout autre **client WebDAV** si vous ne passez pas par ESUP Portail

Description :

Initialement développé par le consortium ESUP Portail, ce serveur permet le stockage des documents en ligne en utilisant le protocole WebDAV. Le socle de ce composant est le serveur Jakarta/Slide. Via différentes méthodes du protocole WebDAV, il est possible de gérer des documents distants (dépôt, téléchargement, suppression, etc.). De plus, le protocole ACP (Access Control Protocol) offre une gestion dynamique des droits (lecture, écriture, etc.) sur les documents.



Les développements qui ont été faits autour de ce serveur sont :

- Le support d'authentications via le SSO (Single Sign-On) CAS.
- La gestion de fédération d'identités par Shibboleth.
- L'externalisation de la gestion des groupes d'utilisateurs.
- Le support des quotas.

IV. Installation

A. Pré requis et variables d'installation

Nous verrons dans ce document que certains composants extérieurs sont nécessaires au bon fonctionnement de ORI-OAI :

- base de données MySQL
- base de données XML eXist
- annuaire LDAP

Nous définissons ici les différentes variables utilisées dans ce support :

- **ORI_HOME** le dossier dans lequel vous ferez l'installation sur votre machine. (Note : lors des captures d'écran de ce support, le dossier ORI_HOME correspondait au dossier /usr/local/ori de notre machine)
- **JAVA_HOME** le dossier d'environnement du JDK de votre machine
- **URL_MYSQL** le serveur sur lequel se trouve la base de données MySQL
- **BASE_MYSQL** le nom de la base de données que vous utilisez
- **URL_EXIST** le serveur sur lequel est installée la base de données eXist. (Note : référez-vous à la documentation eXist pour l'installation de ce serveur : <http://exist.sourceforge.net/>)
- **COLLECTION_EXIST** le nom de la collection que vous utiliserez sur ce serveur eXist. (Note : référez-vous à la documentation eXist pour l'installation de ce serveur)
- **LDAP_ETABLISSEMENT** l'adresse de l'annuaire LDAP que vous utiliserez au cours de l'installation

Il faut aussi connaître :

- Le nom de la machine sur laquelle vous installez ORI-OAI (ex : ori.insa-lyon.fr). Nous nommerons cette adresse **NOM_SERVEUR_ORI** dans ce document.
- L'adresse du serveur SMTP de l'établissement (ex : smtp.insa-lyon.fr). Nous nommerons cette adresse **SMTP_ETABLISSEMENT** dans ce document.

Vous devrez remplacer toutes les variables citées ci-dessus par leur valeur réelle chaque fois que celles-ci apparaissent dans le document.

Remarque générale : dans ce tutoriel, nous utilisons des mots de passe simples. Lors d'un déploiement en production d'ORI-OAI, utilisez des mots de passe complexes et en phase avec la politique de mot de passe (si elle existe) mise en place dans votre établissement.

Remarque concernant eXist : la configuration par défaut de la base de données eXist a besoin d'être modifiée pour mieux répondre aux besoins de ORI-OAI :

- Lorsque les volumes de fiches moissonnées par le moissonneur (ori-oai-harvester) excèdent 10000, il faut modifier la configuration de la base eXist (WEB-INF/conf.xml) en fonction :

```
<watchdog output-size-limit="10000" query-timeout="-1"/>
```

À remplacer par exemple par

```
<watchdog output-size-limit="20000" query-timeout="-1"/>
```

20000 étant alors la limite du nombre de fiches par moisson.

- Par défaut eXist valide les fichiers XML qu'il stocke via le schéma identifié dans la fiche XML. Cela a un coût en terme de performance, de plus la validation des fiches LOM via le schéma standard du LOM lom.xsd n'est pas supporté par le parser de eXist. Désactivez cette validation en éditant le fichier **WEB-INF/conf.xml** et en modifiant :



```
<indexer caseSensitive="no" index-depth="5" preserve-whitespace-mixed-content="no"
stemming="no" suppress-whitespace="both"
tokenizer="org.exist.storage.analysis.SimpleTokenizer" track-term-freq="yes"
validation="auto">
```

par

```
<indexer caseSensitive="no" index-depth="5" preserve-whitespace-mixed-content="no"
stemming="no" suppress-whitespace="both"
tokenizer="org.exist.storage.analysis.SimpleTokenizer" track-term-freq="yes" validation="no">
```

(validation est passé de **auto** à **no**)

- Les modules ori-oai-workflow ainsi que ori-oai-harvester (dans leurs versions 1.0.*) s'appuient sur la Version 1.1.1 de exist.

Enfin, on notera qu'on abandonnera eXist dès la version 1.1.0 de ori-oai-workflow et très rapidement également concernant ori-oai-harvester, cela au profit de l'usage plus usuel d'une BD SQL type MySQL.

Remarque concernant la base de données MySQL :

- ori-oai-workflow requiert l'utilisation d'une base de données transactionnelle. On recommande l'usage de MySql avec le moteur InnoDB.

Aussi, sous linux, dans my.cnf, pour les versions de mysql récentes, on pourra vérifier que l'on a :

```
default-storage_engine = innodb
```

Pour une version plus anciennes, l'option est de la forme :

```
default-table-type = innodb
```

Commandes pour vérifier en ligne de commande (client mysql) le storage utilisé sur une table de votre base :

```
mysql> use `ori-oai-workflow`;
Database changed
mysql> show table status like 'ORI_WORKFLOW_ACL_ROLE';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Name          | Type      | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length | Data_free | Auto_increment | Create_time | Update_time | Check_time | Create_options | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ORI_WORKFLOW_ACL_ROLE | InnoDB   | Dynamic    | 5    | 3276           | 16384       | 16384           | 16384       | 0         | 6              | NULL        | NULL        | NULL       | NULL          | REFER `ori-oai-workflow/ORI_WORKFLOW_ACL_OBJECT_IDENTITY`(`oriAclObjectId`) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

B. Installation des serveurs tomcat

Pour des raisons de performances et d'indépendance des modules, nous préconisons dans un premier temps l'utilisation d'un serveur Tomcat pour chaque Module **ORI-OAI**.

Ouvrez un navigateur web et allez à l'adresse : <http://tomcat.apache.org/download-55.cgi>



Installation et configuration de ORI-OAI version 1.0

Téléchargez dans le répertoire **ORI_HOME/src** le fichier **apache-tomcat-5.5.25.tar.gz** (ou une version 5.5 de Tomcat plus récente)

En tant que **root** (commande **su**), allez dans le répertoire **ORI_HOME/src**.

```
cd ORI_HOME/src
```

Décompressez le fichier **apache-tomcat-5.5.25.tar.gz**

```
tar -xvf apache-tomcat-5.5.25.tar.gz
```

Pour accéder au « Tomcat Manager » à partir d'un navigateur web, il faut définir un utilisateur.

Ouvrez le fichier **tomcat-users.xml**

```
vi ORI_HOME/src/apache-tomcat-5.5.25/conf/tomcat-users.xml
```

Ajoutez la ligne de l'utilisateur **admin** :

```
<tomcat-users>
  <user name="tomcat" password="tomcat" roles="tomcat" />
  <user name="role1" password="tomcat" roles="role1" />
  <user name="both" password="tomcat" roles="tomcat,role1" />
  <user name="admin" password="admin_ori" roles="admin,manager" />
</tomcat-users>
```

Enregistrez le fichier.

Pour s'assurer que l'application fonctionne avec l'encodage UTF-8, ouvrez le fichier **catalina.sh**

```
vi ORI_HOME/src/apache-tomcat-5.5.25/bin/catalina.sh
```

Ajoutez (après la grande section de commentaires)

```
export CATALINA_OPTS="-Dfile.encoding=UTF-8 $CATALINA_OPTS"
```

Enregistrez le fichier.

Allez dans le répertoire **ORI_HOME/src**

```
cd ORI_HOME/src
```

Créez un serveur Tomcat pour chaque module dans **ORI_HOME** en exécutant les commandes ci-dessous:

```
cp -R apache-tomcat-5.5.25 ../tomcat-repository
cp -R apache-tomcat-5.5.25 ../tomcat-harvester
cp -R apache-tomcat-5.5.25 ../tomcat-indexing
cp -R apache-tomcat-5.5.25 ../tomcat-vocabulary
cp -R apache-tomcat-5.5.25 ../tomcat-search
cp -R apache-tomcat-5.5.25 ../tomcat-workflow
cp -R apache-tomcat-5.5.25 ../tomcat-editeur
```

Vous devriez obtenir l'affichage suivant dans le répertoire **ORI_HOME** :

```
ori@localhost: /usr/local/ori - Terminal - Konsole
Session  Édition  Affichage  Signets  Configuration  Aide

drwxr-xr-x 11 ori root 4096 2007-12-01 20:23 tomcat-editeur/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:24 tomcat-exist/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:21 tomcat-harvester/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:21 tomcat-indexing/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:20 tomcat-repository/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:22 tomcat-search/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:21 tomcat-vocabulary/
drwxr-xr-x 11 ori root 4096 2007-12-01 20:22 tomcat-workflow/
[ori@localhost ori]$
```

Voici un résumé des ports qui seront utilisés par les différents modules dans Tomcat

La colonne en gras représente les ports qu'il faudra utiliser pour accéder aux différents modules avec un navigateur web.

(ex : http://NOM_SERVEUR_ORI:8185/ori-oai-workflow pour accéder au module ori-oai-workflow, par exemple).

MODULE / PORT	Shutdown	Non SSL	SSL	AJP 1.3	Proxy	Nom du contexte de déploiement du module
Repository	8280	8180	8480	8380	8580	<i>ori-oai-repository</i>
Harvester	8281	8181	8481	8381	8581	<i>ori-oai-harvester</i>
Indexing	8282	8182	8482	8382	8582	<i>ori-oai-indexing</i>
Vocabulary	8283	8183	8483	8383	8583	<i>ori-oai-vocabulary</i>
Search	8284	8184	8484	8384	8584	<i>ori-oai-search</i>
Workflow	8285	8185	8485	8385	8585	<i>ori-oai-workflow</i>
Editeur	8286	8186	8486	8386	8586	<i>ori-oai-md-editor</i>

Modifiez le fichier **server.xml** qui se trouve dans **chaque** répertoire **tomcat-xxx/conf** en utilisant les ports du tableau ci dessus.

Par exemple, pour le module ori-oai-repository, modifiez le fichier **server.xml** comme suit :

```
vi ORI_HOME/tomcat-repository/conf/server.xml
```

```
<Server port="8280" shutdown="SHUTDOWN">
<!-- Define a non-SSL HTTP/1.1 Connector on port 8180 -->
<Connector port="8180" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8480" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" />
<!-- Define a SSL HTTP/1.1 Connector on port 8480 -->
<!--
<Connector port="8480" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
```

```
-->
<!-- Define an AJP 1.3 Connector on port 8380 -->
<Connector port="8380"
enableLookups="false" redirectPort="8480" protocol="AJP/1.3" />

<!-- Define a Proxied HTTP/1.1 Connector on port 8580 -->
<!-- See proxy documentation for more information about using this. -->
<!--
<Connector port="8580"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" acceptCount="100" connectionTimeout="20000"
proxyPort="80" disableUploadTimeout="true" />
-->
```

Enregistrez le fichier.

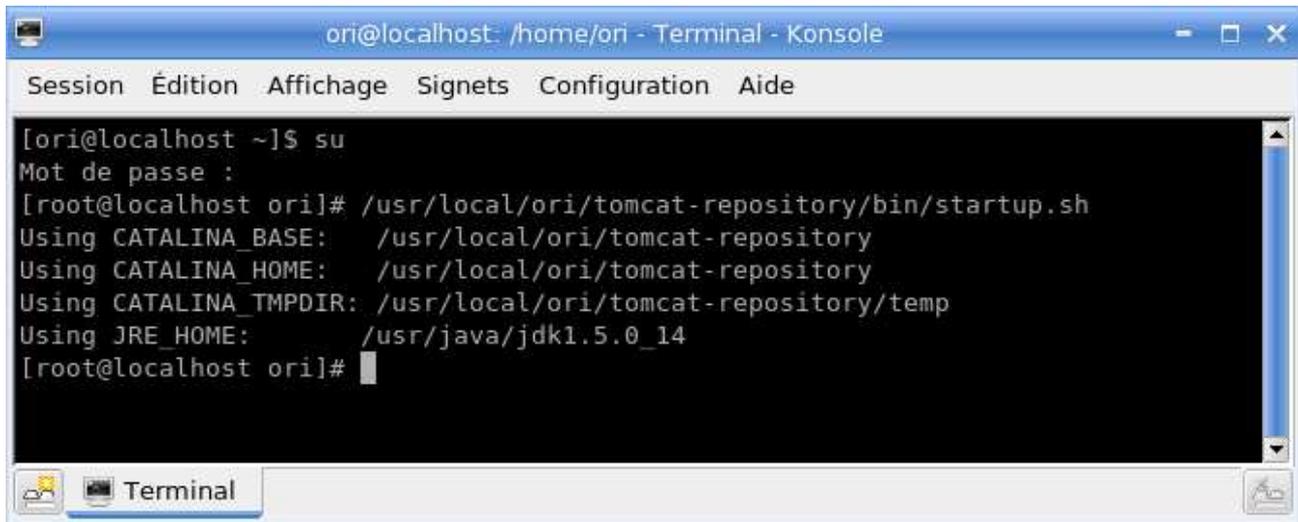
La partie installation des tomcats est terminée.

Vous pouvez maintenant lancer et tester les serveurs Tomcat pour savoir si l'installation s'est bien déroulée:

Exécutez les commandes suivantes:

```
ORI_HOME/tomcat-repository/bin/startup.sh
```

Vous devriez obtenir l'affichage suivant :



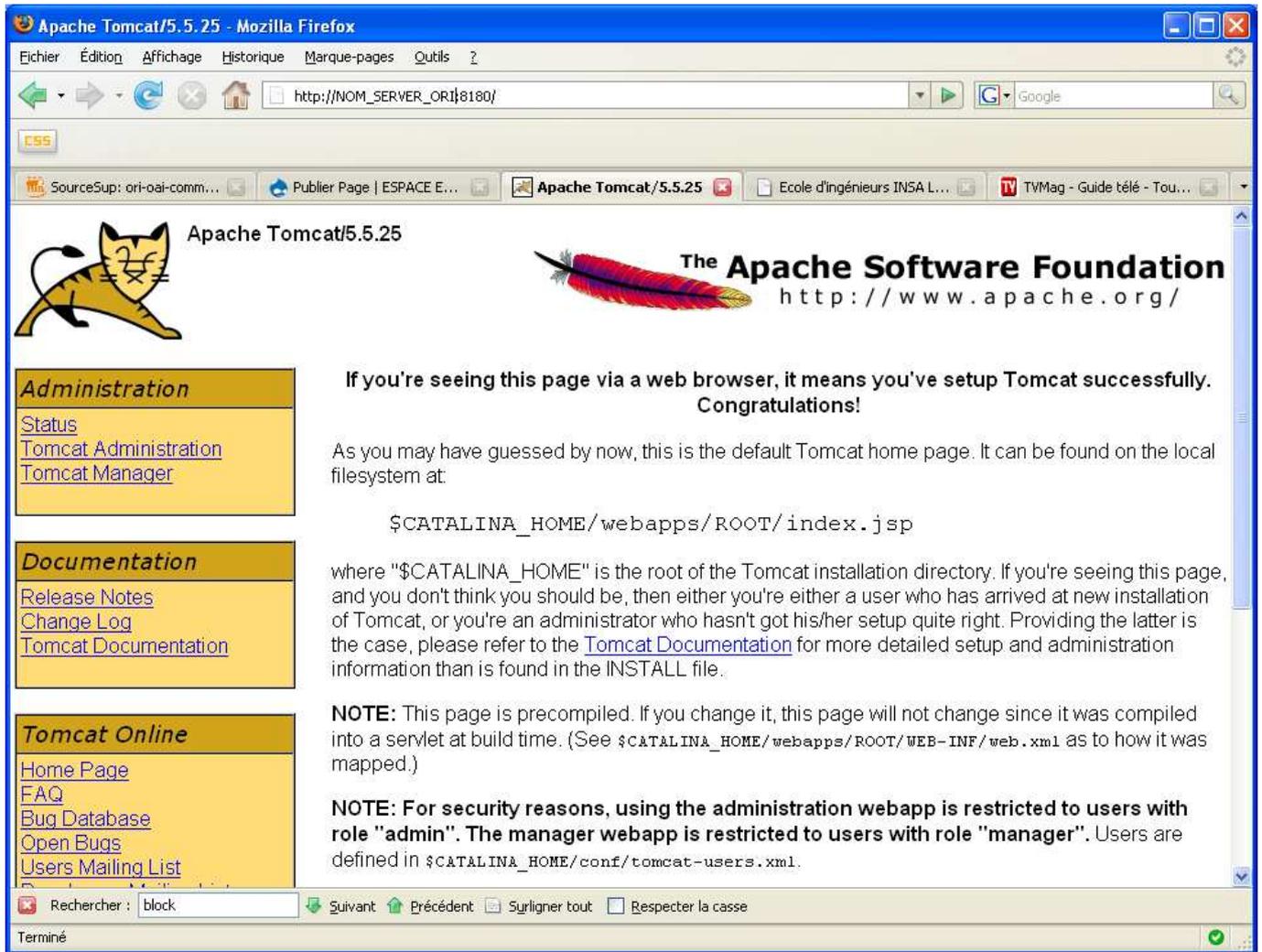
```
ori@localhost: /home/ori - Terminal - Konsole
Session  Édition  Affichage  Signets  Configuration  Aide

[ori@localhost ~]$ su
Mot de passe :
[root@localhost ori]# /usr/local/ori/tomcat-repository/bin/startup.sh
Using CATALINA_BASE:   /usr/local/ori/tomcat-repository
Using CATALINA_HOME:   /usr/local/ori/tomcat-repository
Using CATALINA_TMPDIR: /usr/local/ori/tomcat-repository/temp
Using JRE_HOME:        /usr/java/jdk1.5.0_14
[root@localhost ori]#
```

Ouvrez un navigateur web à l'adresse :

http://NOM_SERVEUR_ORI:8180

Vous devriez obtenir l'affichage suivant :



Recommencez la manipulation en exécutant successivement les commandes:

ORI_HOME/tomcat-editeur/bin/startup.sh

puis testez l'url : **http://NOM_SERVEUR_ORI:8186**

ORI_HOME/tomcat-workflow/bin/startup.sh

puis testez l'url : **http://NOM_SERVEUR_ORI:8185**

ORI_HOME/tomcat-search/bin/startup.sh

puis testez l'url : **http://NOM_SERVEUR_ORI:8184**

ORI_HOME/tomcat-vocabulary/bin/startup.sh

puis testez l'url : **http://NOM_SERVEUR_ORI:8183**

ORI_HOME/tomcat-indexing/bin/startup.sh

puis testez l'url : **http://NOM_SERVEUR_ORI:8182**

ORI_HOME/tomcat-harvester/bin/startup.sh

puis testez l'url : **http://NOM_SERVEUR_ORI:8181**

Une fois que tous les tests sont validés et que tous vos serveurs Tomcat sont maintenant opérationnels, vous pouvez les stopper avant de commencer l'installation des modules **ORI-OAI** :

```
ORI_HOME/tomcat-editeur/bin/shutdown.sh
ORI_HOME/tomcat-workflow/bin/shutdown.sh
ORI_HOME/tomcat-search/bin/shutdown.sh
ORI_HOME/tomcat-vocabulary/bin/shutdown.sh
ORI_HOME/tomcat-indexing/bin/shutdown.sh
ORI_HOME/tomcat-harvester/bin/shutdown.sh
```

C. Subversion

On propose lors de l'installation des différents modules d'ORI-OAI d'utiliser l'outil Subversion pour récupérer chaque module (en mode anonyme donc : lecture seule).

Même si Subversion est un outil plutôt utilisé par les développeurs, il permet ici à l'exploitant de faciliter les mises à jours des versions des modules d'ORI-OAI tout en préservant les configurations personnelles qui peuvent, selon le module, être conséquentes.

Les différents entrepôts subversion sont hébergés par la plateforme de gestion de projets sourcesup (<http://sourcesup.cru.fr/>). Dans l'espace web d'un projet donné, les informations concernant l'entrepôt associé au module se situent dans la page associée à l'onglet « Subversion ».

Pour le module de workflow (et l'éditeur de métadonnées qui partage le même projet) par exemple, l'entrepôt complet est

<http://subversion.cru.fr/ori-workflow>

La version (« taguée ») 1.0.0 a pour url (vous pouvez naviguer avec un navigateur web dans l'entrepôt pour retrouver cela) :

<http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-1.0.0/>

Une fois que vous aurez suivi les instructions d'installation des modules, que vous les aurez configurés, utilisé ... nous vous invitons à mettre à jour vos modules en vous appuyant sur subversion. Pour ce faire, lisez attentivement et mettez en pratique les procédures de mise à jour décrites dans l'**annexe** (donnée en fin de ce document) : **Exploitation d'applications avec subversion - installation et mises à jour.**

D. Modules communs et obligatoires quelque soit la configuration

1. ori-oai-indexing

ORI-OAI-indexing permet de gérer l'indexation et la recherche de documents locaux et distants. Il est appelé par le workflow via le frontal d'indexation pour indexer des fiches locales. En les indexant il reçoit un identifiant ainsi que d'autres éléments pour chacune d'elles. ORI-OAI-Indexing appelle le workflow pour lui demander des fiches à partir de leur identifiant. Ceci est nécessaire dans le cadre d'une recherche.

Le moissonneur fait également appel à lui, toujours par l'intermédiaire du frontal, en ce qui concerne l'indexation. ORI-OAI-indexing indexera une chaîne de caractères contenant la fiche. Le moissonneur envoie d'ailleurs un identifiant qui sera indexé avec la fiche pour faire le lien entre les deux modules. D'autre part si le module d'indexation lui envoie des identifiants de fiches, le moissonneur lui retournera alors toutes les fiches correspondantes. Cette procédure sera appliquée suite à une recherche.

Les recherches dans l'index sont effectuées par le module ORI-OAI-Search. Ce composant envoie une requête de type Lucene au module d'indexation. Il récupère ensuite de la part de ce dernier les attributs qui correspondent à sa recherche tels que le titre ou l'auteur par exemple.

L'installation d'ORI-OAI-Indexing se déroule en plusieurs étapes :

i. Installation

Création des répertoires

Les répertoires suivants doivent être créés sur le serveur :

- **ORI_HOME/indexes/index** : c'est dans ce répertoire que l'index sera créé.
- **ORI_HOME/indexes/tmp** : ce répertoire contiendra les fichiers temporaires nécessaires à la recherche.



La création des répertoires se fait grâce aux commandes suivantes :

```
mkdir -p ORI_HOME/data/indexes/index
mkdir ORI_HOME/data/indexes/tmp
```

Téléchargement du module

Depuis le répertoire **ORI_HOME/src**, lancez la commande suivante pour récupérer les sources de la version 1.0.1 de ce module:

```
svn checkout http://subversion.cru.fr/orioai-indexing/tags/1.0.1 ori-oai-indexing-svn
```

ii. Configuration

Modification du fichier build.properties

Ouvrez le fichier **build.properties** afin d'éditer les paramètres de l'installation:

```
vi ORI_HOME/src/ori-oai-indexing-svn/build.properties
```

et modifiez :

```
#Repertoire de l'index
tomcat.home = ORI_HOME/tomcat-indexing

#Repertoire de déploiement
tomcat.deploy = ORI_HOME/tomcat-indexing/webapps/

#URL vers tomcat
tomcat.URL.deploy = http://NOM_SERVEUR_ORI:8182
```

Enregistrez le fichier

Modification du fichier configIndexing.xml

Ouvrez le fichier **configIndexing.xml** afin d'éditer les paramètres de configuration du module d'indexation :

```
vi ORI_HOME/src/ori-oai-indexing-svn/properties/configIndexing.xml
```

et modifiez :

```
<!-- Repertoire de index -->
<indexDir>ORI_HOME/data/indexes/index</indexDir>

<!-- Repertoire des fichiers temporaires -->
<tmpDir>ORI_HOME/data/indexes/tmp</tmpDir>

<!-- Adresse du Workflow et du Harvester -->
<workflowUrl>http://NOM_SERVEUR_ORI:8185/ori-oai-
workflow/xfire/OriWorkflowService</workflowUrl>

<harvesterUrl>http://NOM_SERVEUR_ORI:8181/ori-oai-
harvester/ws/xfire/HarvesterWebService</harvesterUrl>

<!-- Proxy -->
<proxy>
  <proxyHost></proxyHost>
  <proxyPort></proxyPort>
</proxy>
```

Enregistrez le fichier

NB : Si votre machine passe par un proxy, il faut paramétrer **proxyHost** et **proxyPort** en conséquence.

NB : Un système de cache est présent dans le module d'indexation. Celui-ci permet d'améliorer les performances de recherche en conservant en mémoire vive les résultats des recherches les plus récentes ce qui évite de requêter l'index à chaque recherche.



Configuration des logs

Ouvrez le fichier **log4j.properties** afin d'éditer les paramètres de journalisation du module d'indexation :

```
vi ORI_HOME/src/ori-oai-indexing-svn/properties/log4j.properties
```

et modifiez :

```
# ----- Definition des logs console
log4j.rootLogger=INFO,stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%5p [%c] - %m%n

# ----- Definition des logs fichier
log4j.logger.org.orioai.indexing=INFO,fichier
#log4j.appender.fichier=org.apache.log4j.ConsoleAppender
log4j.appender.fichier=org.apache.log4j.FileAppender

# A MODIFIER
log4j.appender.fichier.file=ORI_HOME/tomcat-indexing/logs/ori-oai-indexing.log

log4j.appender.fichier.layout=org.apache.log4j.PatternLayout
log4j.appender.fichier.layout.ConversionPattern=%5p %d{MMM/dd HH:mm:ss} %c :: %m%n
```

Enregistrez le fichier

iii. Déploiement

Placez vous dans le répertoire **ORI_HOME/src/ori-oai-indexing-svn**

Tapez :

```
ant all
```

Cette commande permet de compiler les sources et de les déployer dans le répertoire **webapps** de **tomcat-indexing**.

Si le module est correctement installé lancez le tomcat :

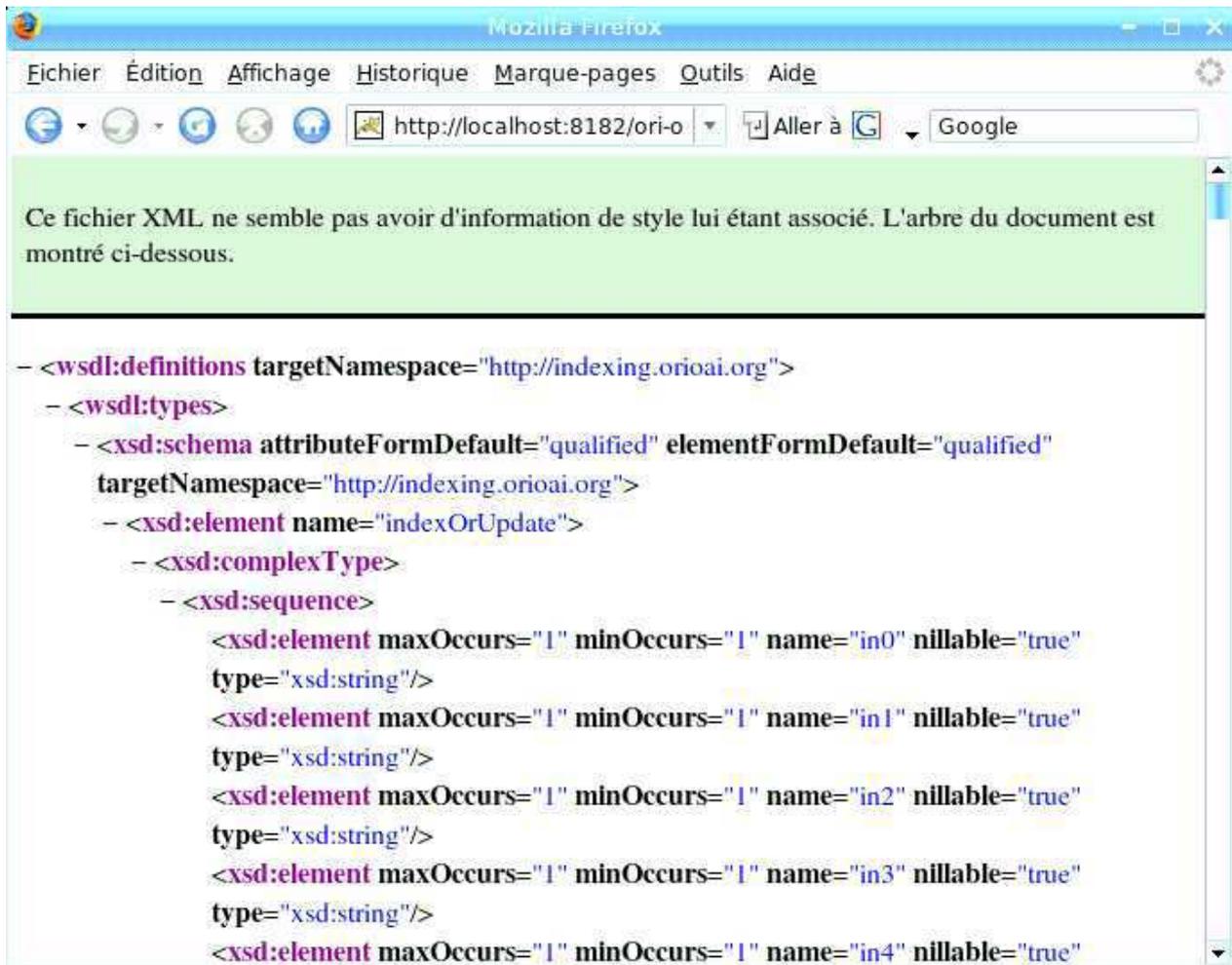
```
ORI_HOME/tomcat-indexing/bin/startup.sh
```

iv. Test

Ouvrez un navigateur et tapez l'url :

« **http://NOM_SERVEUR_ORI:8182/ori-oai-indexing/xfire/IndexingService?wsdl** ».

Vous devriez obtenir l'affichage suivant :



Pour vérifier si le module fonctionne correctement, placez-vous dans le répertoire **ORI_HOME/src/ori-oai-indexing-svn** puis tapez :

```
ant testIndex
```

Deux documents vont alors être indexés. Si l'indexation s'est bien passée vous devriez avoir :

```

[java] Notice : Dublin_Core_example.xml ,Identifiant : id1 ,Repository : UVHC
[java] The notice is correctly indexed
[java] Notice : LOM_example.xml ,Identifiant : id2 ,Repository : Lille1
[java] The notice is correctly indexed

BUILD SUCCESSFUL
Total time: 8 seconds
[or:@localhost indexing]$

```

Il reste à vérifier si la recherche fonctionne correctement. Pour cela tapez :

```
ant testSearch
```

Vous devriez alors obtenir le résultat suivant :

```

ori@localhost: /usr/local/ori/download/indexing
[java] Identifier : id2
[java] Repository : Lille1
[java] Metadata Format : http://ltsc.ieee.org/xsd/LOM
[java] Date Stamp :20070101
[java] End of the notice
[java]
[java] xpath : //lom:general/lom:title/lom:string[@language='fr'], Value : java
[java] xpath : //lom:general/lom:description/lom:string[@language='fr'], Value : exe
mple
[java] xpath : //lom:general/lom:keyword/lom:string[@language='fr'], Value : prog
[java] xpath : //lom:lifeCycle/lom:contribute[lom:role/lom:value='author']/lom:entit
y(name), Value : brochet
[java] xpath : //lom:lifeCycle/lom:contribute[lom:role/lom:value='author']/lom:date/
lom:dateTime, Value : 20070717
[java] Identifier : unit-ori-wf-1-5
[java] Repository : ori-oai-workflow
[java] Metadata Format : http://ltsc.ieee.org/xsd/LOM
[java] Date Stamp :20070703
[java] End of the notice
[java] ***** END OF THE LOM SEARCH *****

BUILD SUCCESSFUL
Total time: 4 seconds
[ori@localhost indexing]$

```

Si tout s'est correctement déroulé, il vous faut supprimer cet index de test en supprimant le contenu du dossier nommé **index** dans **ORI_HOME/data/indexes** afin que les fiches de test n'apparaissent pas dans votre index de production. Pour cela, arrêtez votre tomcat, supprimez le contenu de **index** et redémarrez le tomcat.

NB : Luke (<http://www.getopt.org/luke/>) est un outil graphique qui vous permettra d'explorer votre index. Facile d'utilisation, il vous suffira simplement d'indiquer le répertoire contenant votre index pour consulter vos données.

Important : Il est fortement recommandé de sauvegarder régulièrement votre index en copiant le dossier **index**. Si votre index devenait inutilisable, il vous suffirait alors de supprimer le dossier index et de le remplacer par votre copie la plus récente. Un redémarrage de votre serveur Tomcat hébergeant le module d'indexation sera nécessaire.

Il existe également des procédures de restauration de l'index depuis les modules ori-oai-workflow et ori-oai-harvester.



2. ori-oai-vocabulary

1. Installation

Depuis un répertoire **ORI_HOME/src**, lancez la commande suivante pour récupérer les sources de ce module:

```
svn checkout http://subversion.cru.fr/ori-vocabulary/ori-oai-vocabulary-spring/tags/ori-oai-vocabulary-1.0.0 ori-oai-vocabulary-svn
```

i. Configuration

Ouvrez le fichier **build.properties** dans le dossier **ORI_HOME/src/ori-oai-vocabulary-svn** et modifiez :

```
deploy.home=ORI_HOME/tomcat-vocabulary/webapps/ori-oai-vocabulary
```

Enregistrez le fichier.

Ouvrez **main-config.properties**

```
vi ORI_HOME/src/ori-oai-vocabulary-svn/conf/properties/main-config.properties
```

et modifiez :

```
# ldap [ldap.xml]
# WARNING: you should modify ldapVocabulary.xml TOO (config vcard)
ldap.url=ldap://LDAP_ETABLISSEMENT:PORT_LDAP_ETABLISSEMENT
#ldap.username=cn=admin,dc=www,dc=unit-c,dc=fr
#ldap.password=pass
ldap.username=
ldap.password=
ldap.base=dc=xxx,dc=yyy
ldap.people.searchBase=ou=people
ldap.people.objectClassValue=Person
ldap.people.uid=uid
ldap.group.searchBase=ou=groups
ldap.group.objectClassValue=posixGroup
ldap.group.uid=cn

# exceptions [exceptionHandling.xml]
exceptions.recipientEmail=Mail_de_l_administrateur

# smtp [smtp.xml]
smtp.smtpFromAddress.address=Mail_de_l_administrateur
smtp.smtpFromAddress.personal=ORI
smtp.smtpInterceptAddress.address=Mail_de_l_administrateur
smtp.smtpInterceptAddress.personal=Nom Administrateur
smtp.smtpServer.host=SMTP_ETABLISSEMENT
smtp.smtpServer.port=25

# indexing [indexingVocabulary.xml]
indexing1.wsdlDocumentUrl=http://NOM_SERVEUR_ORI:8182/ori-oai-
indexing/xfire/IndexingService?WSDL
indexing1.lookupServiceOnStartup=false
```

Enregistrez le fichier.

Note : si vous utilisez le serveur LDAP installé localement,

```
ldap.url=ldap://localhost:389
ldap.base= dc=ori,dc=nom_etablissement,dc=fr
```

ii. Déploiement

Placez-vous dans le répertoire « **ORI_HOME/src/ori-oai-vocabulary-svn** ».

Tapez :

ant deploy.

Démarrez le Tomcat correspondant à l'éditeur :

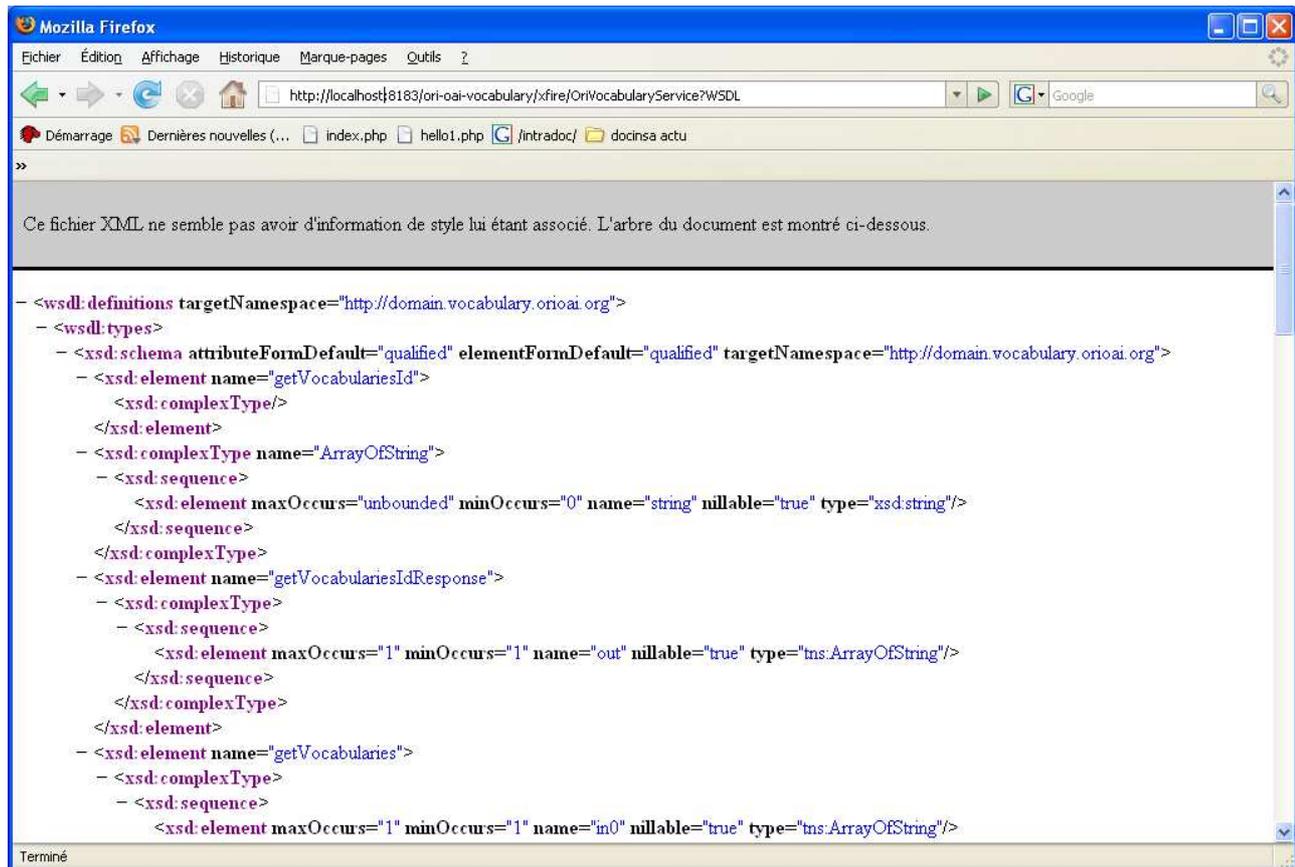
```
ORI_HOME/tomcat-vocabulary/bin/startup.sh
```

iii. Test

Pour tester le module ori-oai-vocabulary, ouvrez un navigateur et tapez l'url :

http://NOM_SERVEUR_ORI:8183/ori-oai-vocabulary/xfire/OriVocabularyService?WSDL

Vous devriez obtenir l'affichage suivant :



```

- <wsdl:definitions targetNamespace="http://domain.vocabulary.orioai.org">
- <wsdl:types>
- <xsd:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://domain.vocabulary.orioai.org">
- <xsd:element name="getVocabulariesId">
  <xsd:complexType>
  </xsd:complexType>
  </xsd:element>
- <xsd:complexType name="ArrayOfString">
- <xsd:sequence>
  <xsd:element maxOccurs="unbounded" minOccurs="0" name="string" nillable="true" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
- <xsd:element name="getVocabulariesIdResponse">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="1" name="out" nillable="true" type="tns:ArrayOfString"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
- <xsd:element name="getVocabularies">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="1" name="in0" nillable="true" type="tns:ArrayOfString"/>
  </xsd:sequence>
  </xsd:complexType>
  </xsd:element>

```

3. ori-oai-search

i. Installation

Depuis un répertoire **ORI_HOME/src**, lancez la commande:

```
svn checkout http://subversion.cru.fr/ori-oai-search/tags/1.0.1 ori-oai-search-svn
```

ii. Configuration

Configuration minimale

Ouvrez le fichier **build.properties** afin d'éditer les paramètres de l'installation :

```
vi ORI_HOME/src/ori-oai-search-svn/build.properties
```

et modifiez :

```

#Répertoire d'installation de Tomcat
tomcat.home = ORI_HOME/tomcat-search

#Répertoire de déploiement
#deploy.home = C:/esupdev/esupdev-2.5/webapps

```



Installation et configuration de ORI-OAI version 1.0

```
deploy.home = ORI_HOME/tomcat-search/webapps
```

```
#Nom de distribution de l'application  
#app.name.deploy=ori-oai-search  
app.name.deploy=ori-oai-search
```

Enregistrez le fichier.

Dans ori-oai-search, plusieurs configurations sont proposées par défaut :

- **config.example.unt.xml** : configuration centrée sur les UNT qui propose des interfaces de recherche sur les métadonnées LOM et Dublin core sans distinction. Cette configuration est notamment utilisée pour le portail de recherche de l'UNT UNIT (<http://www.unit.eu>) en version servlet
- **config.example.documentaire.xml** : configuration plus destinée aux établissements où on distingue d'une part les documents locaux de l'établissement et les documents moissonnés par OAI-PMH
- **config.example.formations.xml** : interfaces de recherche sur un format de métadonnées CDM. Cette configuration est notamment utilisée pour le portail de recherche de formations de l'UNR Nord-pas-de-Calais (<http://www.unr-npdc.org/formations>) en version portlet intégrée dans le CMS infoglué
- **config.example.complet.xml** : configuration de démonstration complète où on distingue des recherches sur différents formats et une recherche croisée sur les formats LOM, Dublin core et CDM

Lors de l'installation de ce module, vous pouvez choisir une de ces configurations comme une base de départ que vous pourrez modifier selon vos besoins ou partir d'une toute nouvelle configuration.

Dans le cadre de notre exemple d'installation, dupliquez le fichier **config.example.complet.xml** et renommez-le en **config.xml** :

```
cd ORI_HOME/src/ori-oai-search-svn/properties  
cp config.example.complet.xml config.xml
```

Ouvrez le fichier **config.xml**. Ce fichier permet d'éditer tous les principaux paramètres de ce module :

```
vi ORI_HOME/src/ori-oai-search-svn/properties/config.xml
```

et modifiez :

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
  
<!--  
* ORI-OAI-search - Copyright (c) 2007 ORI-OAI  
* For any information please refer to http://www.ori-oai.org  
* You may obtain a copy of the licence at http://www.ori-oai.org/license/  
-->  
  
<config>  
  
<index_search_url>http://NOM_SERVEUR_ORI:8182/ori-oai-  
indexing/xfire/IndexingService</index_search_url>  
  
<vocabulary_url>http://NOM_SERVEUR_ORI:8183/ori-oai-  
vocabulary/xfire/OriVocabularyService</vocabulary_url>  
  
<default_docs_per_page>10</default_docs_per_page>  
<max_docs_per_page>40</max_docs_per_page>  
  
...  
</config>
```

Enregistrez le fichier.

Il existe dans ori-oai-search une configuration pour un plugin de recherche intégré à IE 7 ou à Firefox à partir de la version 2. Ce plugin respecte le format Open Search. Plusieurs fichiers sont pré-configurés dans la version de base de ori-oai-search. Ceux-ci se trouvent dans le dossier **ORI_HOME/src/ori-oai-search-svn/webapp/opensearch**. Il faut alors éditer le(s) fichier(s) cités dans le fichier config.xml parmi les suivants :

- **simple_cdm.xml**
- **simple_complet.xml**
- **simple_dc.xml**
- **simple_lom.xml**

Dans notre exemple, nous utiliserons le fichier **simple_complet.xml** :

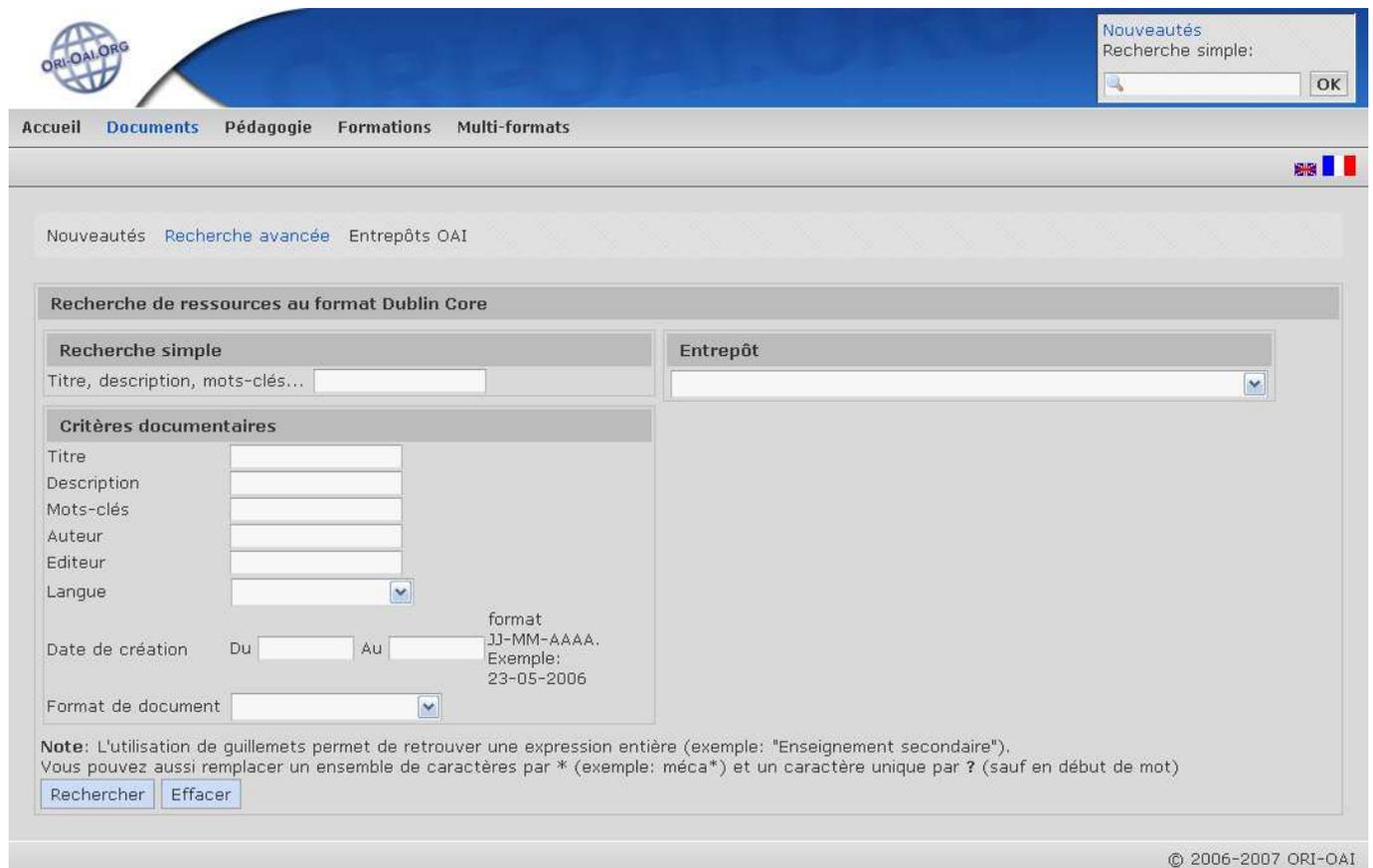
```
<Url type="text/html" template="http://NOM_SERVEUR_ORI:8184/ori-oai-search/.....
<moz:SearchForm>http://NOM_SERVEUR_ORI:8184</moz:SearchForm>
```

Enregistrez le fichier.

Feuille de style

Tout comme il existe plusieurs configurations, il existe plusieurs feuilles de style CSS proposées par défaut dans ori-oai-search. Celles-ci se trouvent dans le dossier **ORI_HOME/src/ori-oai-search-svn/webapp/css** :

- ori-oai-search.css : CSS par défaut :



The screenshot shows the ORI-OAI search interface. At the top left is the ORI-OAI.ORG logo. A search box with an 'OK' button is in the top right. The main navigation menu includes 'Accueil', 'Documents', 'Pédagogie', 'Formations', and 'Multi-formats'. Below the menu, there are links for 'Nouveautés', 'Recherche avancée', and 'Entrepôts OAI'. The main content area is titled 'Recherche de ressources au format Dublin Core' and contains a search form. The form has a 'Recherche simple' section with a text input for 'Titre, description, mots-clés...'. Below this is the 'Critères documentaires' section with fields for 'Titre', 'Description', 'Mots-clés', 'Auteur', 'Editeur', 'Langue', and 'Date de création'. The 'Date de création' field has 'Du' and 'Au' sub-fields and a 'format' label with the example 'JJ-MM-AAAA.' and 'Exemple: 23-05-2006'. There is also a 'Format de document' dropdown. At the bottom of the form are 'Rechercher' and 'Effacer' buttons. A note at the bottom of the form explains the use of quotation marks and wildcards. The footer of the page shows '© 2006-2007 ORI-OAI'.

- ori-oai-search_old-style.css: CSS avec un look approchant celui du site institutionnel du projet :



Installation et configuration de ORI-OAI version 1.0



Nouveautés
Recherche simple:

[Accueil](#) [Documents](#) [Pédagogie](#) [Formations](#) [Multi-formats](#)

Documents > Recherche avancée

[Nouveautés](#) [Recherche avancée](#) [Entrepôts OAI](#)

Recherche de ressources au format Dublin Core

<p style="text-align: center;">Recherche simple</p> <p>Titre, description, mots-clés... <input type="text"/></p> <hr/> <p style="text-align: center;">Critères documentaires</p> <p>Titre <input type="text"/></p> <p>Description <input type="text"/></p> <p>Mots-clés <input type="text"/></p> <p>Auteur <input type="text"/></p> <p>Editeur <input type="text"/></p> <p>Langue <input type="text"/></p> <p>Date de création Du <input type="text"/> Au <input type="text"/> format JJ-MM-AAAA. Exemple: 23-05-2006</p> <p>Format de document <input type="text"/></p>	<p style="text-align: center;">Entrepôt</p> <p><input type="text"/></p>
--	--

Note: L'utilisation de guillemets permet de retrouver une expression entière (exemple: "Enseignement secondaire").
Vous pouvez aussi remplacer un ensemble de caractères par * (exemple: méca*) et un caractère unique par ? (sauf en début de mot)

© 2006-2007 ORI-OAI

- ori-oai-search_unr-npdc.css : CSS utilisée sur le site de l'UNR Nord-pas-de-Calais:

Nouveautés
Recherche simple:



Documents > Recherche avancée

- [Accueil](#)
- [Documents](#)
- [Pédagogie](#)
- [Formations](#)
- [Multi-formats](#)

[Nouveautés](#) [Recherche avancée](#) [Entrepôts OAI](#)

Recherche de ressources au format Dublin Core

<p style="text-align: center;">Recherche simple</p> <p>Titre, description, mots-clés... <input type="text"/></p> <hr/> <p style="text-align: center;">Critères documentaires</p> <p>Titre <input type="text"/></p> <p>Description <input type="text"/></p> <p>Mots-clés <input type="text"/></p> <p>Auteur <input type="text"/></p> <p>Editeur <input type="text"/></p> <p>Langue <input type="text"/></p> <p>Date de création Du <input type="text"/> Au <input type="text"/> format JJ-MM-AAAA. Exemple: 23-05-2006</p> <p>Format de document <input type="text"/></p>	<p style="text-align: center;">Entrepôt</p> <p><input type="text"/></p>
--	--

Note: L'utilisation de guillemets permet de retrouver une expression entière (exemple: "Enseignement secondaire").
Vous pouvez aussi remplacer un ensemble de caractères par * (exemple: méca*) et un caractère unique par ? (sauf en début de mot)

Pour utiliser une de ces feuilles de style, il suffit de renommer celle qui vous intéresse en **ori-oai-search.css** dans le même dossier.

iii. Déploiement

Placez-vous dans le répertoire **ORI_HOME/src/ori-oai-search-svn** et tapez la commande suivante :



```
cd ORI_HOME/src/ori-oai-search-svn
ant all-servlet
```

L'indication "**build successful**" indique que le déploiement s'est déroulé avec succès.

Notez que ce module est disponible en 2 versions :

- *Servlet* : installation autonome sur un serveur d'applications
- *Portlet* : intégration dans un moteur de portlets comme un ENT ou le CMS infoglué

Vous pouvez alors démarrer le serveur tomcat associé à ce module avec la commande :

```
ORI_HOME/tomcat-search/bin/startup.sh
```

iv. Test

Vous pouvez maintenant tester la bonne configuration du module et l'accès aux interfaces de recherche en allant sur :

« http://NOM_SERVEUR_ORI:8184/ori-oai-search »

Notez que l'index est vide à cette étape, vous ne trouverez donc aucune ressource lors de la recherche.

E. Agrégation de ressources

1. ori-oai-harvester

Le moissonneur est chargé de d'agréger les récoltes issues de plusieurs entrepôts OAI-PMH, afin de stocker et d'indexer les fiches des ressources qu'ils contiennent.

Il a donc accès à Internet pour « récolter » les entrepôts en amont, et en aval il doit pouvoir dialoguer avec le moteur d'indexation ori-oai-indexing.

i. Installation

Pour installer le moissonneur ori-oai-harvester, il faut commencer par rapatrier le projet via *SUBVERSION* à l'aide de la commande suivante depuis le dossier **ORI_HOME/src**:

```
svn checkout https://subversion.cru.fr/ori-oai-harvest/tags/1.0.0 ori-oai-harvester-svn
```

Ouvrez le fichier **build.properties** :

```
vi ORI_HOME/src/ori-oai-harvester-svn/build.properties
```

et modifiez :

```
#JDK 1.5 directory
java.home=JAVA_HOME

#Tomcat directory
tomcat.home=ORI_HOME/tomcat-harvester

#Deployment directory
deploy.home=ORI_HOME/tomcat-harvester/webapps

#Log Directory (optionnal - default : ${tomcat.home}/logs )
#log.dir=E:/logs

#Log mode : prod or debug
log.mode=prod
```

Enregistrez le fichier.

Dupliquez le fichier **ori.example.properties** et renommez le **ori.properties**. C'est dans ce fichier que l'on définit les URLs de la base eXist et du moteur d'indexation.

```
cd ORI_HOME/src/ori-oai-harvester-svn/properties
cp ori.example.properties ori.properties
```



Installation et configuration de ORI-OAI version 1.0

Ouvrez ce fichier

```
vi ORI_HOME/src/ori-oai-harvester-svn/properties/ori.properties
```

et modifiez :

```
# propertie file for ORI Harvester
#harvester global options

# set to true each time you want to update database with config file
harvester.reloadconfig=true
harvester.initconfig=ORI_HOME/tomcat-harvester/webapps/ori-oai-harvester/WEB-INF/config/harvesterConfig.xml

#indexing
indexing.ws.wsdlDocumentUrl=http://NOM_SERVEUR_ORI:8182/ori-oai-indexing/xfire/IndexingService?WSDL
indexing.ws.lookupServiceOnStartup=false

# set to true ONLY if you want to harvest in spite of unreachable indexing Web Service
indexing.ignore.error=false

#xml database (must be previously created)
exist.url=xmlldb:exist://URL_EXIST/exist/xmlrpc
exist.login=ori
exist.password=ori
exist.collection=/db/COLLECTION_EXIST/harvester

# if true, persist deleted OAI records in xml database
storage.persistDeleted=true
```

Enregistrez le fichier **ori.properties** après avoir créer le fichier **harvesterConfig.xml** le cas échéant.

Note : la collection /db/**COLLECTION_EXIST/harvester** doit être créée au préalable dans la base de données exist.

Optionnel : **harvesterConfig.xml** peut être utilisé pour amorcer l'application avec des moissons prédéfinies. Si on met `harvester.reloadconfig` à true, il ne faut pas oublier de le créer à partir de l'exemple :

```
cd ORI_HOME/src/ori-oai-harvester-svn /WebContent/WEB-INF/config/harvesterConfig.example.xml
cp harvesterConfig.example.harvesterConfig.xml
```

Placez-vous dans le répertoire **ORI_HOME/src/ori-oai-harvester-svn**

```
cd ORI_HOME/src/ori-oai-harvester-svn
```

Tapez :

```
ant deploy
```

ii. Test

Une fois le module déployé, il faut lancer Tomcat pour l'exécuter :

```
ORI_HOME/tomcat-harvester/bin/startup.sh
```

L'accès à l'interface du moissonneur se fait par l'URL :

« **http://NOM_SERVEUR_ORI:8181/ori-oai-harvester** »

Liste des définitions de moisson

Rafraîchir

Identifiant	Collection	Préfixe OAI	Ensemble	Dernière moisson :
theses_inpt   	dc/inpt	oai_dc		2008-01-14
thes_ups   	dc/ups	oai_dc		2008-01-14
these_insa   	dc/insa	oai_dc		2008-01-14
 Définir une nouvelle moisson				

Récupération

Ré-indexer toutes les fiches moissonnées

Attention : il faut bien s'assurer d'avoir supprimé l'index avant de lancer la restauration !

Tout ré-indexer

Pour lancer une moisson, cliquez sur la flèche verte correspondante, et vérifiez l'état de la récolte dans l'onglet « Récolte ». Dès qu'une date de « dernière moisson » apparaît, c'est que la récolte est terminée. Appuyez sur le bouton « rafraîchir » si ce n'est pas le cas, jusqu'à obtention de la date.

Quand la moisson est terminée, consultez le rapport dans l'onglet « rapports ». Cette page liste les rapports des moissons effectuées pour chaque définition.

Le menu « récoltes » liste les contenus des moissons déjà effectuées :

Liste des récoltes

Rafraîchir

Identifiant	Entrepôts		
inpt_theses 	Dernière moisson :	2007-10-05T13:30:01Z	Nombre de documents
	préfixe	oai_dc	
	Ensemble		
	ethesis.inp-toulouse.fr	301	
	Total :	301	
lom_unit 	Dernière moisson :	2007-10-02T15:44:46Z	Nombre de documents
	préfixe	lom	
	Ensemble		
	www.unit.eu	424	
	Total :	424	
mit_archi 	Dernière moisson :	2007-10-05T09:38:05Z	Nombre de documents
	préfixe	oai_dc	
	Ensemble	hdl_1721.1_7772	
	DSpace at MIT	436	
	Total :	436	



Installation et configuration de ORI-OAI version 1.0

Le menu « Tâche programmées » liste des informations sur toutes les programmations dont font l'objet les définitions de moissons :

The screenshot shows the 'Moissonneur ORI-OAI' application interface. The top navigation bar includes 'Définitions', 'Récoltes', 'Tâches programmées', 'Rapports', and 'A propos'. The 'Tâches programmées' menu is selected, displaying a table of scheduled harvest jobs.

JobName	NextFire	PreviousFire	TriggerName
injacs0	25 janv. 2007 23:00:00		harvests-trigger.injacs0
MIT0	26 janv. 2007 01:30:00		harvests-trigger.MIT0
docinsa0	26 janv. 2007 00:30:00		harvests-trigger.docinsa0
theselNPO	25 janv. 2007 23:30:00		harvests-trigger.theselNPO

Enfin, le menu « Rapports » permet de visualiser l'historique des moissons, et l'évolution des récoltes :

The screenshot shows the 'Moissonneur ORI-OAI' application interface with the 'Rapports' menu selected. It displays a list of harvest reports and a detailed view for the 'UNIT' report.

Identifiant	Dernière moisson :
injacs	2007-01-25T15:23:36Z
theselNP	2007-01-25T15:24:38Z
docinsa	2007-01-25T15:26:10Z
MIT	2007-01-25T15:27:56Z
UNIT	2007-01-25T15:32:45Z

Date :	Documents ajoutés/mis à jour	Documents supprimés	Rapport :								
2007-01-25T15:31-32Z	277	277	harvesting UNIT started harvesting UNIT finished : 277 documents added and 277 documents deleted in 1 m 13 s 641 ms <table border="1"> <thead> <tr> <th>Nom de cet entrepôt</th> <th>Documents ajoutés/mis à jour</th> <th>Documents supprimés</th> <th>Durée</th> </tr> </thead> <tbody> <tr> <td>UNIT</td> <td>277</td> <td>277</td> <td>1 m 13 s 641 ms</td> </tr> </tbody> </table>	Nom de cet entrepôt	Documents ajoutés/mis à jour	Documents supprimés	Durée	UNIT	277	277	1 m 13 s 641 ms
Nom de cet entrepôt	Documents ajoutés/mis à jour	Documents supprimés	Durée								
UNIT	277	277	1 m 13 s 641 ms								

Si tout s'est bien passé, vous pouvez rechercher les documents moissonnés dans l'interface de recherche (ori-oai-search) à l'URL :

« http://NOM_SERVEUR_ORI:8184/ori-oai-search »

Exemple « non contractuel » :



Recherche simple:



OK

Accueil



[Nouveautés](#) [Recherche avancée](#) [Entrepôts OAI](#)

Recherche par établissement

Les Établissements

- Thèses INP Toulouse [344]
- Thèses INSA Toulouse [130]
- Thèses UPS Toulouse [86]

© 2006-2007 ORI-OAI

2. ori-oai-repository

L'entrepôt ORI-OAI expose les fiches moissonnées (par le module ori-oai-harvester) ou publiées (par le module ori-oai-workflow). Pour ce faire ce module interroge le module d'indexation (ori-oai-indexing), afin de répondre aux requêtes OAI des moissonneurs.

La configuration de l'entrepôt OAI consiste à choisir le contenu de ce qui va être rendu disponible aux moissonneurs OAI-PMH, ainsi que la façon dont ce contenu va être exposé à travers des ensembles sélectifs (en fonction d'une taxonomie par exemple).

i. Installation

Pour installer l'entrepôt ori-oai-repository, il faut rapatrier le projet via SUBVERSION à l'aide de la commande suivante depuis le dossier **ORI_HOME/src** :

```
svn checkout https://subversion.cru.fr/ori-oai-repo/tags/1.0.0 ori-oai-repository-svn
```

ii. Configuration

Ouvrez le fichier **build.properties** :

```
vi ORI_HOME/src/ori-oai-repository-svn/build.properties
```

et modifiez :

```
java.home=JAVA_HOME

#Tomcat directory
tomcat.home=ORI_HOME/tomcat-repository

#Deployment directory
deploy.home=ORI_HOME/tomcat-repository/webapps
```

Enregistrez le fichier.

Modification d'un chemin dans web.xml :



```
vi ORI_HOME/src/ori-oai-repository-svn/WebContent/WEB-INF/web.xml
```

Modifier l'emplacement du fichier de propriétés :

```
<context-param>
  <!--
    The location of the oaicat.properties file containing
    datapairs used to initialize the OAI repository software.
  -->
  <param-name>properties</param-name>
  <param-value> ORI_HOME/src/ori-oai-repository-svn/properties/ori-
oaicat.properties</param-value>
</context-param>
```

Dupliquez depuis le dossier **properties** le fichier **ori-oaicat.example.properties** et renommez le **ori-oaicat.properties**.

```
cd ORI_HOME/src/ori-oai-repository-svn/properties
cp ori-oaicat.example.properties ori-oaicat.properties
```

Ouvrez le fichier **ori-oaicat.properties** :

```
vi ORI_HOME/src/ori-oai-repository-svn/properties/ori-oaicat.properties
```

et modifiez:

```
# ORI-OAI-repository Configuration
# classes used for main servlet
AbstractCatalog.oaiCatalogClassName=org.orioai.repository.domain.service.ORICatalog
AbstractCatalog.recordFactoryClassName=org.orioai.repository.domain.logic.ORIRecordFactory
AbstractCatalog.secondsToLive=3600

# datestamp granularity
#AbstractCatalog.granularity=YYYY-MM-DD
AbstractCatalog.granularity=YYYY-MM-DDThh:mm:ssZ

# Change to include properties peculiar to your implementation of AbstractCatalog
ORICatalog.maxListSize=40
ORICatalog.listSets.maxListSize=300
ORIRecordFactory.repositoryIdentifier=mon_etablissement.ori

# Custom Identify response values
Identify.repositoryName=ORI MON_ETABLISSEMENT
Identify.adminEmail=mailto:mail_de_l_administrateur
Identify.earliestDatestamp=2006-12-04T00:00:00Z
Identify.deletedRecord=no
Identify.description.1=<description><oai-identifier
xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier\"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance\"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai-identifier
http://www.openarchives.org/OAI/2.0/oai-
identifier.xsd\"><scheme>oai</scheme><repositoryIdentifier>
mon_etablissement.ori</repositoryIdentifier><delimiter>:</delimiter><sampleIdentifier>oai:
mon_etablissement.ori:ORI-0000012</sampleIdentifier></oai-identifier></description>

# main Spring context for servlet
Spring.contextFile=oaiHandlerContext.xml

# xslt files for crosswalk formats
xslt.dc2lom=ORI_HOME/tomcat-repository/webapps/ori-oai-repository/WEB-INF/xslt/lom2dc.xsl

# Web service for ori-oai-indexing (repository-ws.xml)
indexing.ws.wsdlDocumentUrl=http://NOM_SERVEUR_ORI:8182/ori-oai-
indexing/xfire/IndexingService?WSDL
indexing.ws.lookupServiceOnStartup=false

# Web service for ori-oai-vocabulary (repository-ws.xml)
vocabulary.taxonomyID=unit_taxonomie
```



```
vocabulary.ws.wsdlDocumentUrl=http://NOM_SERVEUR_ORI:8183/ori-oai-  
vocabulary/xfire/OriVocabularyService?WSDL  
vocabulary.ws.lookupServiceOnStartup=false  
  
# static MD names (repository-domain.xml)  
static.mdIdentifier=md-ori-oai-id  
static.mdFormat=md-ori-oai-namespace  
static.mdRepository=md-ori-oai-repository  
static.mdDatestamp=md-ori-oai-datestamp  
  
#repository name for workflow records  
workflow.repositoryName=ori-oai-workflow
```

Enregistrez le fichier.

Ouvrez depuis le dossier **properties** le fichier **repository-crosswalks.xml** qui permet de définir les formats exposés.

```
vi ORI_HOME/src/ori-oai-repository-svn/properties/repository-crosswalks.xml
```

et modifiez :

```
<bean id="LOMtoDCCrosswalk"  
class="org.orioai.repository.domain.logic.crosswalk.XSLTCrosswalk"  
init-method="init">  
  
<!-- property name="xsltFile" value="{xslt.dclom}" / -->  
<property name="xsltFile" value="ORI_HOME/tomcat-repository/webapps/ori-oai-repository/WEB-  
INF/xslt/lom2dc.xsl" />  
  
<property name="nameSpaceURL" value="http://www.openarchives.org/OAI/2.0/oai_dc/ " />  
<property name="schemaURL" value="http://www.openarchives.org/OAI/2.0/oai_dc.xsd" />  
<property name="inPrefix" value="lom" />  
<property name="outPrefix" value="oai_dc" />  
  
</bean>
```

Enregistrez le fichier.

Modifier la XSL où le nom du contexte est en dur ihm_main.xsl :

```
vi ORI_HOME/src/ori-oai-repository-svn/WebContent/xsl/ihm_main.xsl
```

Et adapter la valeur du contexte :

```
<xsl:param name="context" select="'ori-oai-repository'" />
```

Note : la valeur de ce paramètre **DOIT** correspondre avec le paramètre app.name.deploy du fichier build.properties, par défaut « **ori-oai-repository** »

Il est possible de filtrer les fiches que l'ont veut exposer à l'aide du fichier **repository-domain.xml** selon plusieurs critères :

- le nom des entrepôts d'où les fiches sont originaires (un nom spécial « **ori-oai-workflow** » désigne les fiches locales)
- le format des fiches (DC, LOM, TEF...)

(Se référer à la documentation du module pour des précisions).

Ouvrez le fichier **repository-domain.xml**

```
vi ORI_HOME/src/ori-oai-repository-svn/properties/repository-domain.xml
```

et modifiez :

```
<property name="exposedRepositories">  
<list>  
  <!-- <value>ori-oai-workflow</value> -->  
</list>
```

```
</property>
```

Important : En laissant en commentaire la liste des entrepôts à afficher, tous les entrepôts configurés seront exposés.

Enregistrez le fichier.

iii. Déploiement

Lancer la commande ANT deploy pour le déploiement :

```
cd ORI_HOME/src/ori-oai-repository-svn/  
ant deploy
```

Une fois le module déployé, il faut lancer Tomcat pour l'exécuter :

```
ORI_HOME/tomcat-repository/bin/startup.sh
```

iv. Test

Pour accéder à l'interface Web de l'entrepôt, utilisez l'URL :

« http://NOM_SERVEUR_ORI:8180/ori-oai-repository »

Vous devez accéder à cette interface :



Pour l'affichage en XML brut, vous pouvez également tester l'URL suivante dans un navigateur :

http://NOM_SERVEUR_ORI:8180/ori-oai-repository/OAIHandler?verb=Identify

F. Productions de données

1. ori-oai-md-editor

ori-oai-md-editor est l'éditeur de métadonnées XML ORI-OAI basé sur Orbeon Forms. Orbeon Forms est une application J2EE qui utilise XForms côté serveur pour proposer dans un navigateur des interfaces Web Ajax.



XForms, recommandation W3C, est une technologie qui permet de décrire en XML des formulaires manipulant des fichiers XML.

ori-oai-md-editor apporte un certain nombre d'outils permettant de mettre en oeuvre l'édition de fiches de métadonnées :

- des XForms pour un certain nombre de schémas de métadonnées et de cas d'utilisation,
- des "widgets" qui permettent de faciliter la création de formulaires, c'est à dire d'alléger les XForms notamment dans l'écriture de blocs souvent redondant,
- d'une procédure de sauvegarde/chargement de XForms (via l'IHM ou via Web Service : dans le cas du Web Service il est ainsi client avec un module ori-oai-workflow-spring),
- d'une procédure permettant le chargement (mise à jour régulière) de vocabulaires via Web Service : il est ainsi client d'un module ori-oai-vocabulary,
- d'une procédure permettant d'ajouter de nouveaux formulaires en copiant/collant simplement un formulaire existant.

i. Installation

Depuis le dossier **ORI_HOME/tomcat-editeur/webapps/**, récupérez la version du module via la commande :

```
svn checkout http://subversion.cru.fr/ori-workflow/ori-oai-md-editor/tags/ori-oai-md-editor-1.0.1 ori-oai-md-editor
```

ii. Configuration

Ouvrez le fichier **config.xml**

```
vi ORI_HOME/tomcat-editeur/webapps/ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor/config.xml
```

et modifiez :

```
<ori-oai>
  <workflow-spring>
    <location>http://NOM_SERVEUR_ORI:8185/ori-oai-workflow</location>
  </workflow-spring>

  <vocabulary>
    <location>http://NOM_SERVEUR_ORI:8183/ori-oai-vocabulary</location>
  </vocabulary>

  <md-editor>
    <location>http://NOM_SERVEUR_ORI:8186/ori-oai-md-editor</location>
    <vocab-cache-time>43200000</vocab-cache-time>
  </md-editor>
</ori-oai>
```

Enregistrez le fichier.

iii. Démarrage

Démarrez le Tomcat correspondant à l'éditeur :

```
ORI_HOME/tomcat-editeur/bin/startup.sh
```

iv. Tests

Accédez à la première page de l'éditeur depuis un navigateur web :

http://NOM_SERVEUR_ORI:8186/ori-oai-md-editor/

Vous devriez voir apparaître quelque chose de similaire à cette capture d'écran :

Bienvenue sur l'Editeur de Métadonnées ORI : ORI-OAI-MD-EDITOR



Ce module vous permet d'éditer différentes fiches de métadonnées selon différents formulaires.

Voici les différents formulaires disponibles :

- [oaidc-full](#)
- [lom-full-extend](#)
- [lom-full](#)
- [lom-author-light](#)

Ce module peut-être utilisé seul ou au travers du module de workflow.

Ce module est connecté à un module de vocabulaire Ori pour proposer un ensemble de vocabulaires dans les différents éditeurs.

Voici les configurations actuelles de votre éditeur ORI :

workflow:

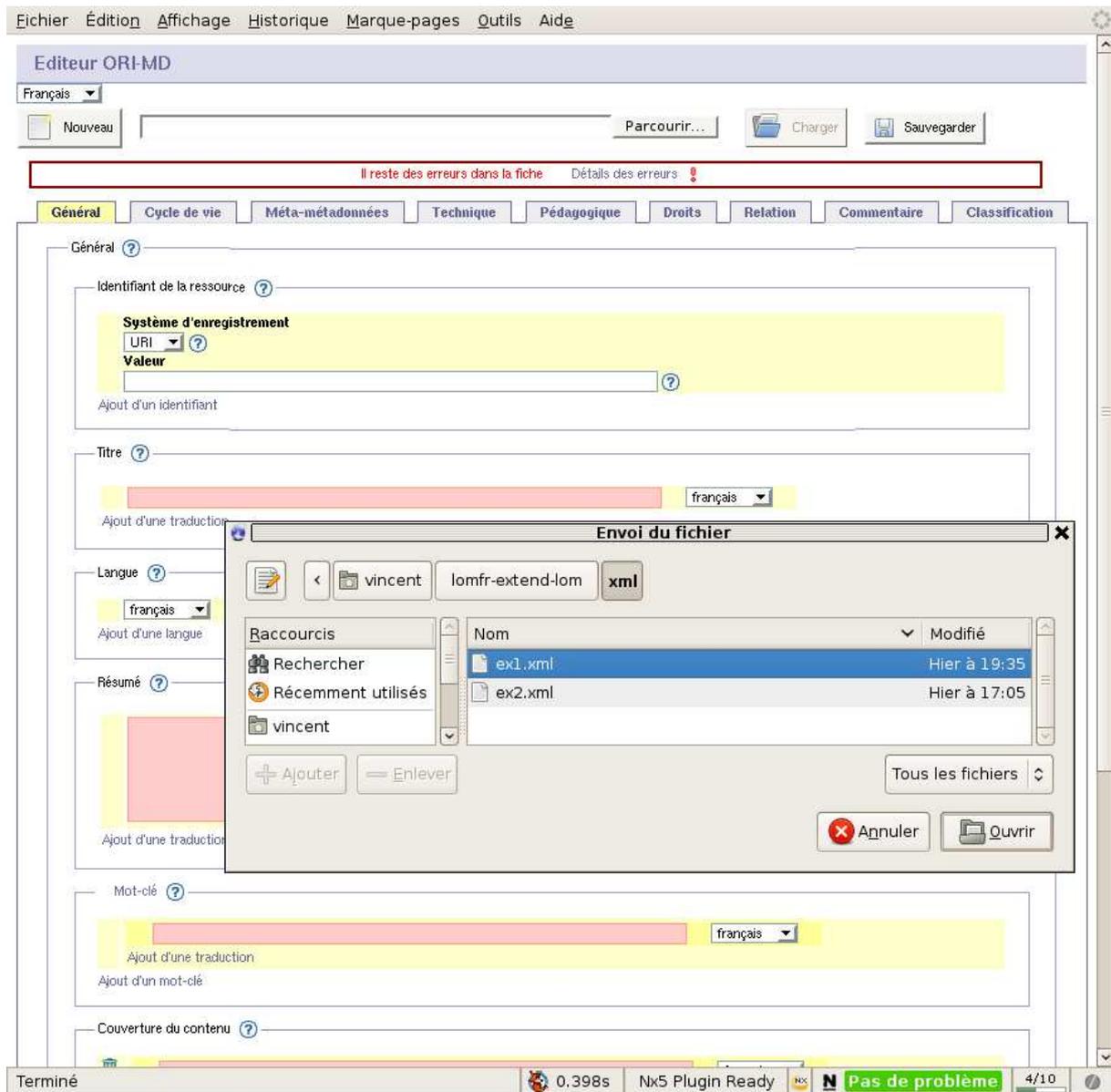
<http://ori-oai-workflow.univ-rennes1.fr/workflow>

vocabulary:

<http://ori-oai-vocabulary.univ-rennes1.fr/vocabulary>



Note : Si le module de vocabulaire est bien configuré, lancé, que les applications dont il dépend sont également bien configurées et lancées. Vous pouvez utiliser les formulaires de l'éditeur de métadonnées directement par cette interface : vous utilisez alors le module comme un éditeur WEB de fiches de métadonnées :



2. ori-oai-workflow

ori-oai-workflow est l'application qui permet de mettre en œuvre la procédure de référencement/indexation de son établissement. Techniquement, il permet de faire manipuler des fiches de métadonnées XML aux différents acteurs de l'établissement, cela via une IHM fonction du workflow mis en place dans l'établissement.

Il utilise OI-OAI-MD-EDITOR pour l'édition des fiches XML en proposant, en fonction des configurations, un formulaire choisi parmi ceux disponibles. Il permet une identification dans son système d'information (LDAP, CAS). Il peut interagir avec d'autres applicatifs au travers du workflow, notamment il peut interagir avec le module ori-oai-indexing pour indexer les fiches (que l'on souhaite « publier » ... ceci étant donc fonction de la configuration).

i. Installation

Depuis le dossier **ORI_HOME/src/**, récupérez la version du module via la commande :

```
svn checkout http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-1.0.1 ori-oai-workflow-svn
```

ii. Configuration

Ouvrez le fichier **build.properties**

```
vi ORI_HOME/src/ori-oai-workflow-svn/build.properties
```



Installation et configuration de ORI-OAI version 1.0

et modifiez :

```
deploy.home=ORI_HOME/tomcat-workflow/webapps/ori-oai-workflow
```

Enregistrez le fichier.

Créer une collection **workflow** dans la collection **COLLECTION_EXIST** de la base de données eXist

Ouvrez le fichier **main-config.properties**

```
vi ORI_HOME/src/ori-oai-workflow-svn/conf/properties/main-config.properties
```

Dans ce fichier, les variables **identifieur.localApplicationPrefix** et **identifieur.urlApplicationPrefix** doivent être adaptées en fonction de l'établissement.

Exemple :

```
identifieur.localApplicationPrefix=insa-lyon-ori-wf-1
identifieur.urlApplicationPrefix=http://ori.insa-lyon.fr/uid/
```

Modifiez également l'adresse du LDAP :

```
ldap.providerUrl=ldap://LDAP_ETABLISSEMENT/dc=xxxx,dc=yyyyyy
```

Modifiez:

```
# eXist [spring-dao.xml spring-exist.xml]
exist.collectionPath=/db/COLLECTION_EXIST/workflow
exist.url=xmldb:exist://URL_EXIST/exist/xmlrpc
exist.username=ori
exist.password=ori

# hibernate [dao.xml -> advanced config can be made in hibernate.cfg.xml]
hibernate.connection.driver_class=com.mysql.jdbc.Driver
hibernate.connection.url=jdbc:mysql://URL_MYSQL/BASE_MYSQL
hibernate.connection.username=mysql_ori
hibernate.connection.password=mysql_ori
#hibernate.properties.dialect=org.hibernate.dialect.MySQLDialect
# in fact with MySQL, innodb is required - so verify the storage engine of your MySQL DataBase
hibernate.properties.dialect=org.hibernate.dialect.MySQLInnoDBDialect

# identifieur [spring-services.xml]
identifieur.localApplicationPrefix=mon_etablissement-ori-wf-1
identifieur.urlApplicationPrefix=http://mon_etablissement.fr/uid/

# ldap [acegi-authentication-ldap.xml see it if you want to connect not anonymously]
ldap.providerUrl=ldap://LDAP_ETABLISSEMENT/dc=xxxx,dc=yyyyyy
#ldap.managerDn=cn=admin,dc=www,dc=unit-c,dc=fr
#ldap.managerPassword=pass
# ldap authentication/groups [acegi-authentication.xml]
ldap.people.userDnPattern=uid={0},ou=people
ldap.people.peopleRetrieveFilter=(objectClass=Person)
ldap.people.peopleSearchBase=ou=people
ldap.people.uidAttribute=uid
ldap.people.mailAttribute=mail
#ldap.group.groupSearchBase=ou=groups
ldap.group.groupSearchBase=
ldap.group.groupSearchFilter=(memberUid={0})
ldap.group.groupSearchUserUid=memberUid
ldap.group.groupUidSearchFilter=cn={0}
ldap.group.groupRetrieveFilter=(objectClass=posixGroup)
ldap.group.groupRoleAttribute=cn
# see acegi-authentication-additional-groups.xml to add "virtual group" via people attributes

# authentication -> for cas authentication, you should configure web.xml
```



```
# moreover users that can log with cas must be in ldap too
authentication.casAuthenticationEnabled=true
authentication.ldapAuthenticationEnabled=true

# exceptions [exceptionHandling.xml]
exceptions.recipientEmail=mail_de_l_administrateur

# smtp [smtp.xml]
smtp.smtpFromAddress.address= mail_de_l_administrateur
smtp.smtpFromAddress.personal=ORI
smtp.smtpInterceptAddress.address= mail_de_l_administrateur
smtp.smtpInterceptAddress.personal=Admin
smtp.smtpServer.host=SMTP_ETABLISSEMENT
smtp.smtpServer.port=25

# indexing [spring-webservices-client.xml (in fact you can use more than one indexing service
- edit directly spring-webservices-client.xml)]
indexing1.namespaceUri=http://NOM_SERVEUR_ORI:8182/ori-oai-indexing/xfire/IndexingService?WSDL
indexing1.lookupServiceOnStartup=false

vocabulary.wsdlDocumentUrl=http://NOM_SERVEUR_ORI:8183/ori-oai-
vocabulary/xfire/OriVocabularyService?WSDL
vocabulary.lookupServiceOnStartup=false

# md-editor [spring-metadata-types.xml]
mdeditor.url=http://NOM_SERVEUR_ORI:8186/ori-oai-md-editor
```

Enregistrez le fichier.

Pour affecter des rôles à des utilisateurs, nous allons mettre les utilisateurs dans des groupes virtuels.

Ouvrez le fichier **acegi-authentication-additional-groups.xml**

vi ORI_HOME/src/ori-oai-workflow-svn/conf/properties/spring/acegi/acegi-authentication-additional-groups.xml

et modifiez :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-
beans.dtd">

<beans>
  <bean name="additionalGroupsFiltersMapping"
class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="all">
          <value>(objectClass=Person)</value>
        </entry>
        <entry key="moderators">
          <value>( (& (objectClass=Person) (uid=login_moderateur)) (& (objectClass=Person) (ui
d=gadmin)) (& (objectClass=Person) (uid=smagadur)) (& (objectClass=Person) (uid=vbonamy)) )<
/value>
        </entry>
        <entry key="admins">
          <value>(& (objectClass=Person) (uid=login_administrateur))</value>
        </entry>
      </map>
    </property>
  </bean>
</beans>
```

Enregistrez le fichier.



iii. Déploiement

Placez-vous dans le répertoire **ORI_HOME/src/ori-oai-workflow-svn**

```
cd ORI_HOME/src/ori-oai-workflow-svn
```

Afin de compiler et déployer l'application dans le répertoire de déploiement cible, tapez

```
ant deploy
```

Si le déploiement s'est bien terminé, vous devriez avoir le message « **build successful** ».

Ensuite, afin de créer les tables de la base de données SQL, c'est-à-dire afin d'initialiser celle-ci, tapez

```
ant init
```

A la question voulez vous initialiser la base SQL, répondez

```
Y
```

Afin de répercuter les configurations des permissions et rôles globaux dans la base de données SQL, tapez :

```
ant update-acls
```

Démarrez le Tomcat correspondant au module de workflow :

```
ORI_HOME/tomcat-workflow/bin/startup.sh
```

iv. Test

Pour tester la connexion au module ori-oai-vocabulary, à la base eXist, Mysql, ...

Placez-vous dans le répertoire « **ORI_HOME/src/ori-oai-workflow-svn** ».

Tapez :

```
ant remoteconfigtest
```

Voici ce que vous devriez obtenir à peu de choses près :

```
ori@localhost: /usr/local/ori/download/ori-oai-workflow-spring
saction (1): transaction manager [org.springframework.orm.hibernate3.HibernateTransaction
Manager@17e60a1]; default rollback = true
[junit] oriwf: 1736 INFO org.springframework.test.AbstractTransactionalSpringContext
Tests.endTransaction(AbstractTransactionalSpringContextTests.java:281) - Rolled back tra
nsaction after test execution
[junit] oriwf: 1767 INFO org.springframework.test.AbstractTransactionalSpringContext
Tests.startNewTransaction(AbstractTransactionalSpringContextTests.java:314) - Began tran
saction (1): transaction manager [org.springframework.orm.hibernate3.HibernateTransaction
Manager@17e60a1]; default rollback = true
[junit] oriwf: 1770 INFO org.springframework.test.AbstractTransactionalSpringContext
Tests.endTransaction(AbstractTransactionalSpringContextTests.java:281) - Rolled back tra
nsaction after test execution
[junit] -----
[junit]
[junit] Testcase: testGetVocabulariesId took 8,422 sec
[junit] Testcase: testWFgetExistIdsFromQuery took 0,206 sec
[junit] Testcase: testSearchIndexingServices took 0,347 sec
[junit] Testcase: testInitialDirContextFactory took 0,043 sec
[junit] Testcase: testConnectUrls took 0,036 sec
[junit] Testcase: testSmtip took 0,035 sec

BUILD SUCCESSFUL
Total time: 10 seconds
[ori@localhost ori-oai-workflow-spring]$
```

v. Utilisation – Référencer un document

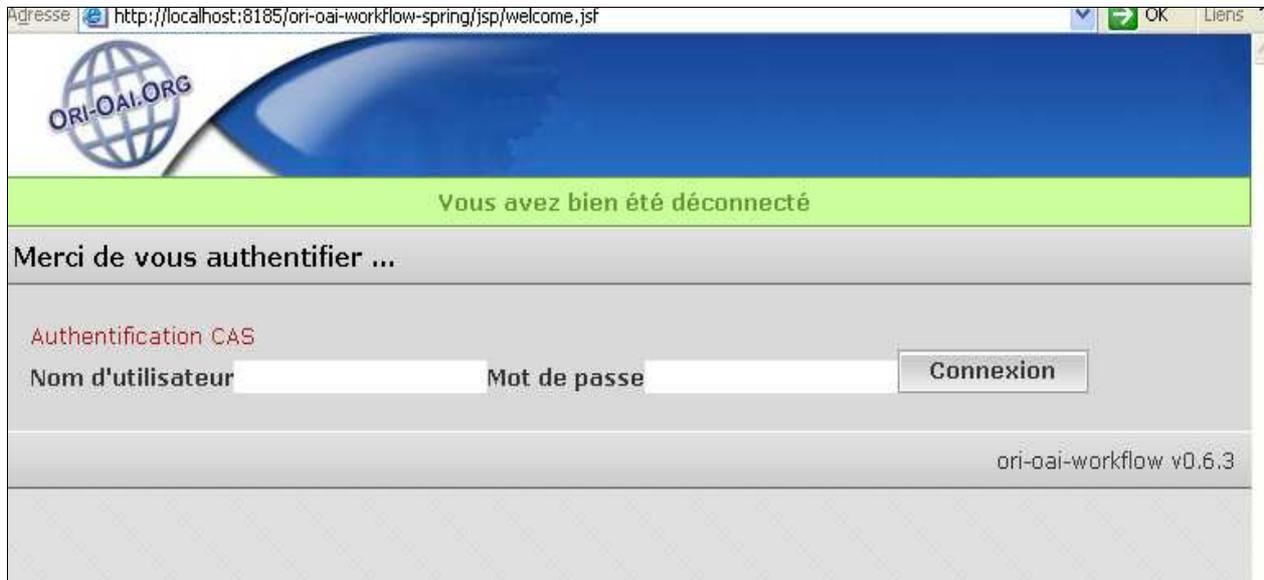


Installation et configuration de ORI-OAI version 1.0

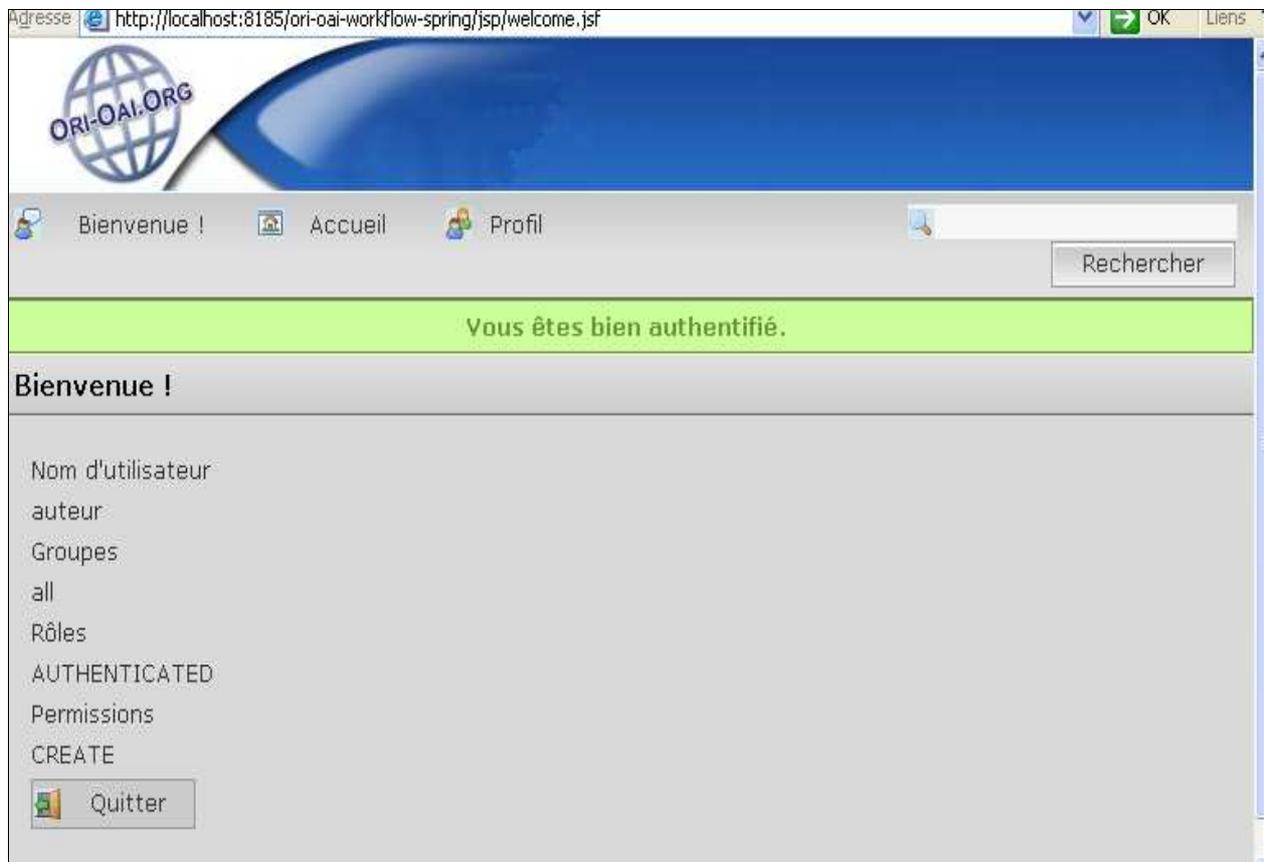
Avec votre navigateur, vous pouvez vous connecter à **http://NOM_SERVEUR_ORI:8185/ori-oai-workflow** et tester l'application :

Le workflow proposé par défaut pour les fiches LOM comporte trois états :

- privé,
- en attente de publication,
- publié.



Identifiez-vous avec un login ayant le rôle **auteur. (login_auteur)**



Cliquez sur « **Accueil** ».



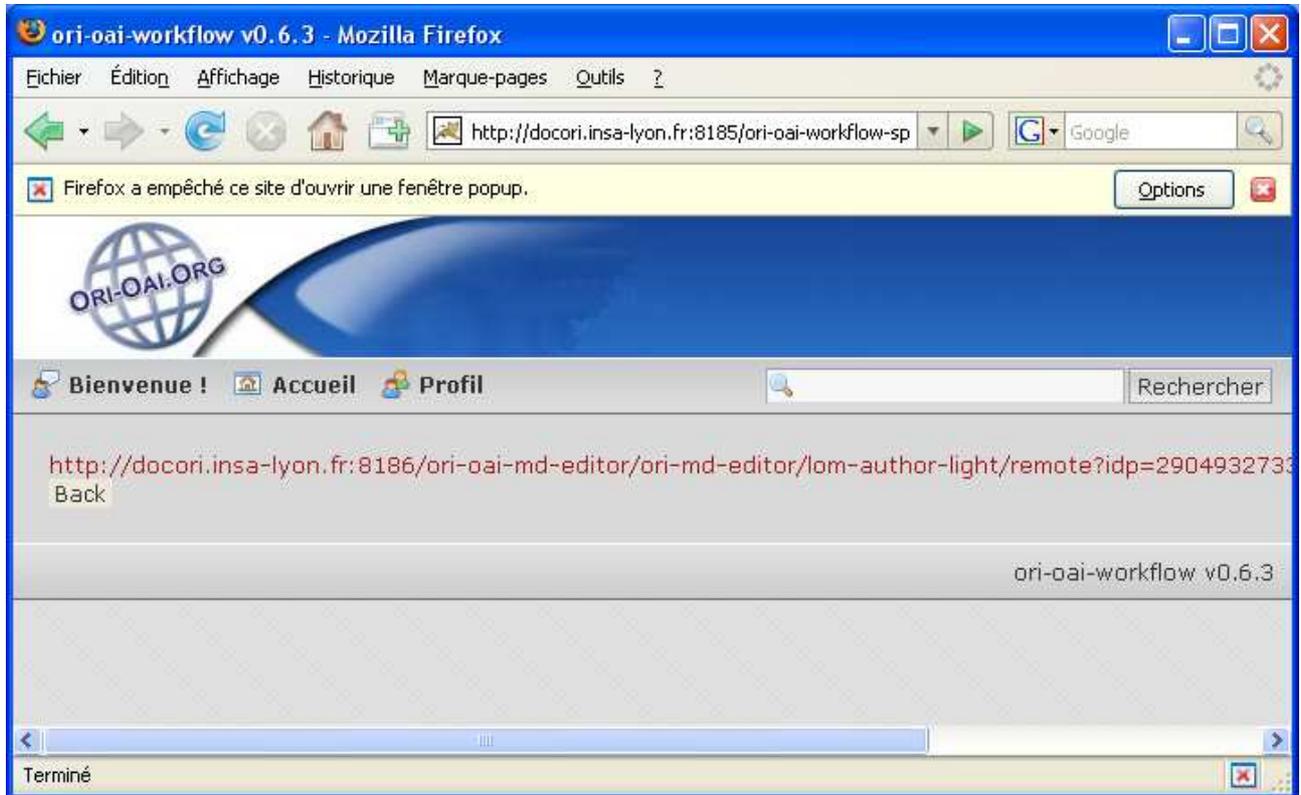
Installation et configuration de ORI-OAI version 1.0

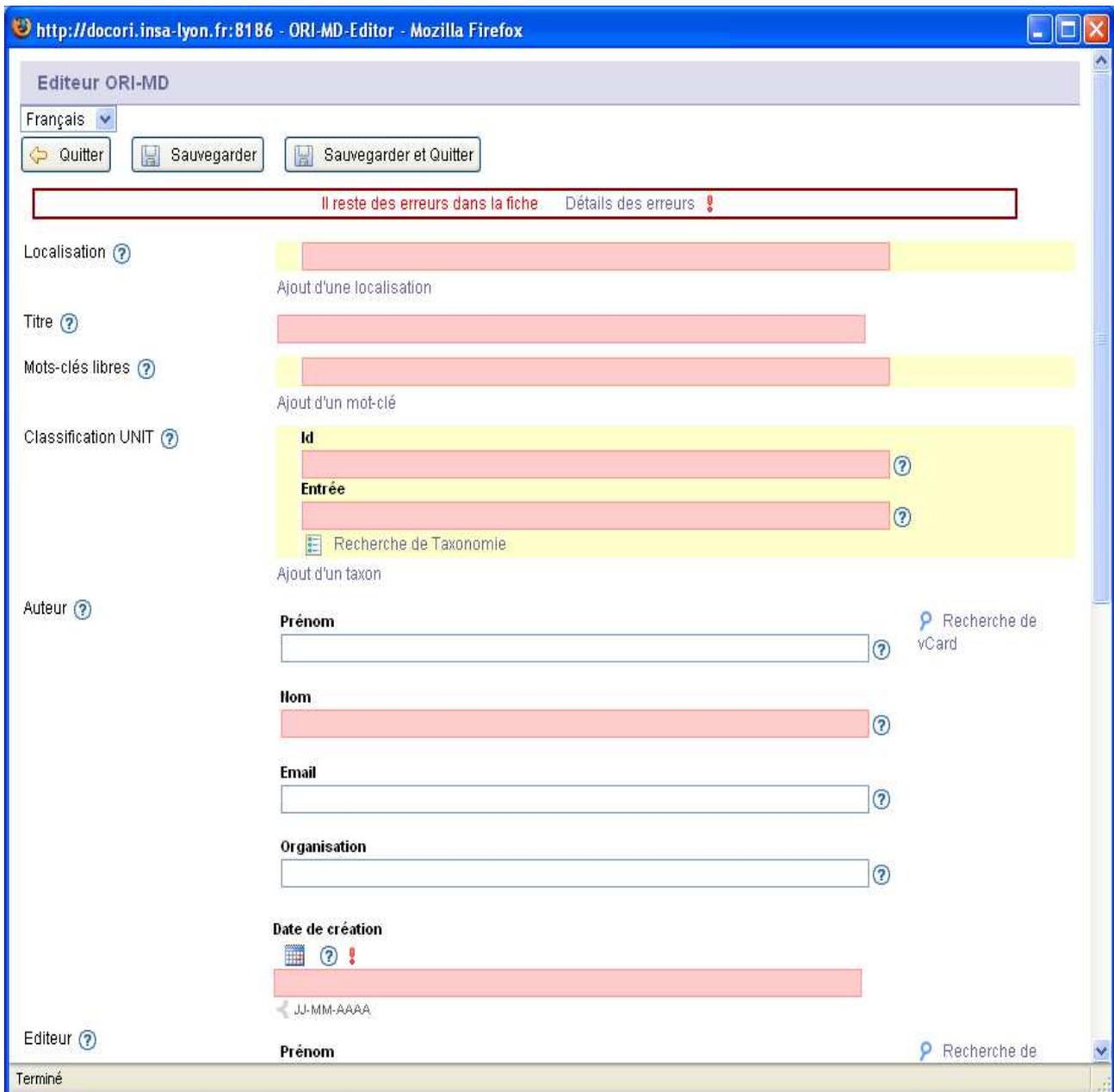
Dans la partie « Référencer une Ressource », choisissez le type de métadonnées que vous souhaitez créer (par exemple « **Ressource Pédagogique [LOM]** »).

Vous aurez peut-être une alerte indiquant que le navigateur a refusé d'ouvrir une fenêtre.

Sous Firefox : **Firefox a empêché ce site d'ouvrir une fenêtre popup**

Cliquez sur **Options** (dans le petit bandeau jaune), puis sur **Autoriser les popup pour NOM_SERVEUR_ORI**





Remplissez le formulaire, lorsque celui-ci **est rempli et validé** (le message d'erreur en haut du formulaire a disparu), cliquez sur « **Sauvegarder** ».

Les champs obligatoires sont en rose.

http://docori.insa-lyon.fr:8186 - ORI-MD-Editor - Mozilla Firefox

Editeur ORI-MD

Français

Quitter Sauvegarder Sauvegarder et Quitter

Localisation ? bibliothèque

Ajout d'une localisation

Titre ? le langage PHP

Mots-clés libres ? programmation

Ajout d'un mot-clé

Classification UNIT ?

Id f3 ?

Entrée informatique ?

Recherche de Taxonomie

Ajout d'un taxon

Auteur ?

Prénom Recherche de vCard

Nom Doc'INSA ?

Email ?

Organisation ?

Date de création Wednesday June 27, 2007 ?

2007-06-27 JJ-MM-AAAA

Editeur ?

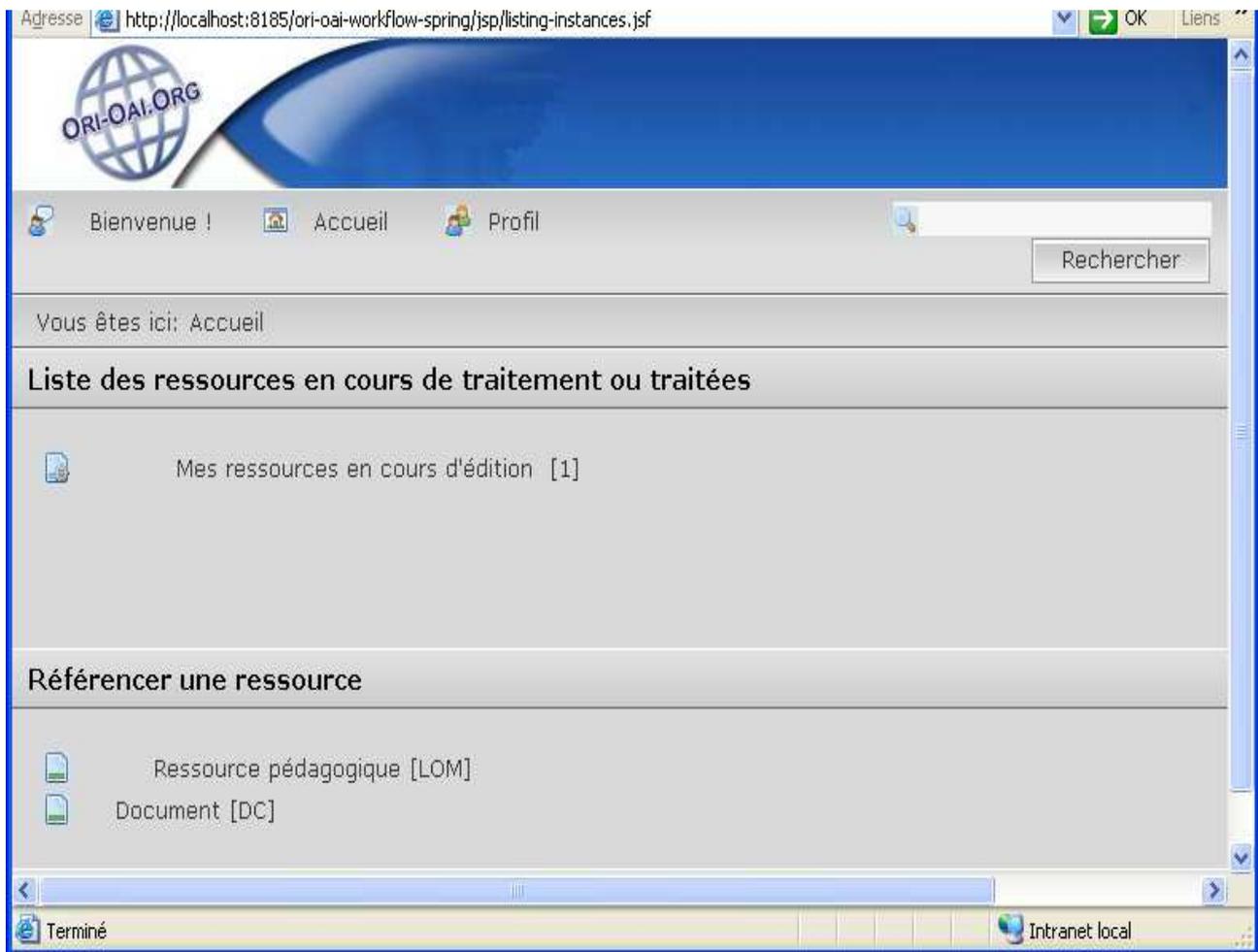
Prénom Recherche de vCard

Nom

Terminé

Vous pourrez revenir sur ce formulaire ultérieurement si vous le souhaitez.

Pour Demander la publication des métadonnées que vous avez saisies, cliquez sur **Mes ressources en cours d'édition** [1].



Adresse | http://localhost:8185/ori-oai-workflow-spring/jsp/listing-instances.jsf

BIENVENUE ! Accueil Profil

Rechercher

Vous êtes ici: Accueil

Liste des ressources en cours de traitement ou traitées

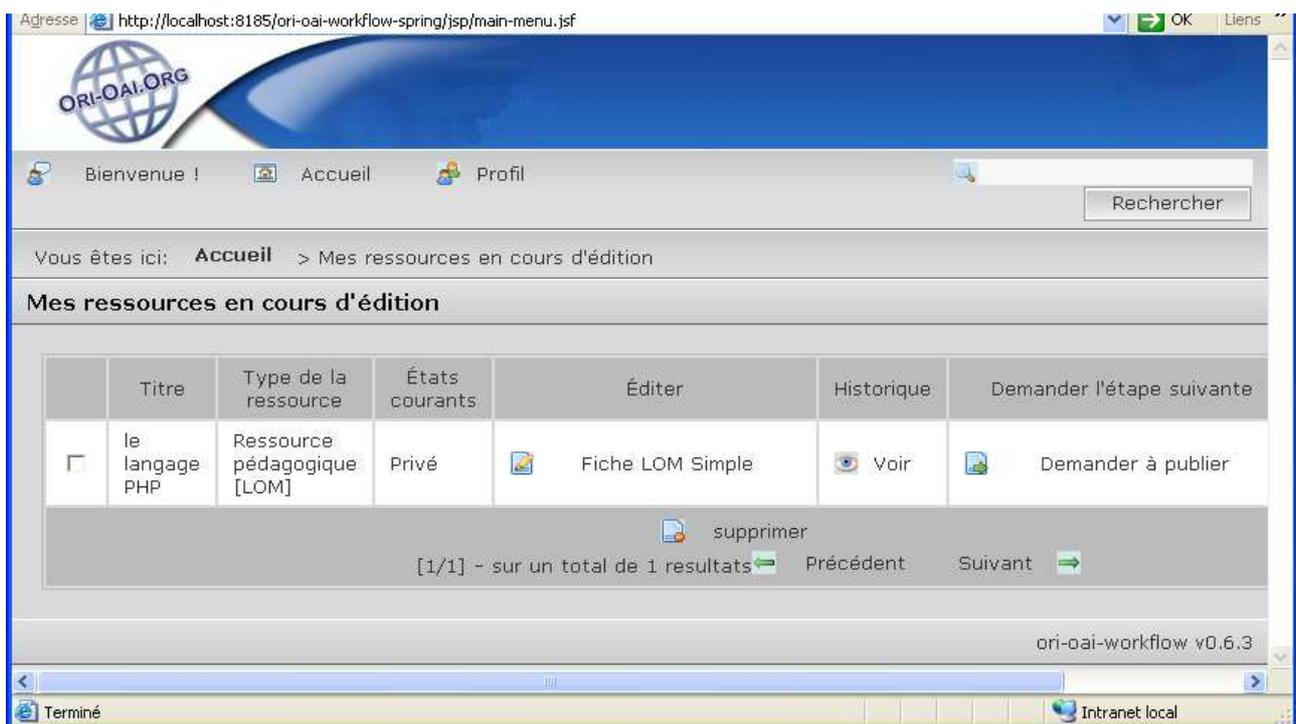
Mes ressources en cours d'édition [1]

Référencer une ressource

Ressource pédagogique [LOM]
Document [DC]

Terminé Intranet local

Le document est dans l'état « **privé** ».



Adresse | http://localhost:8185/ori-oai-workflow-spring/jsp/main-menu.jsf

BIENVENUE ! Accueil Profil

Rechercher

Vous êtes ici: Accueil > Mes ressources en cours d'édition

Mes ressources en cours d'édition

	Titre	Type de la ressource	États courants	Éditer	Historique	Demander l'étape suivante
<input type="checkbox"/>	le langage PHP	Ressource pédagogique [LOM]	Privé	Fiche LOM Simple	Voir	Demander à publier

supprimer

[1/1] - sur un total de 1 résultats ← Précédent Suivant →

ori-oai-workflow v0.6.3

Terminé Intranet local



Installation et configuration de ORI-OAI version 1.0

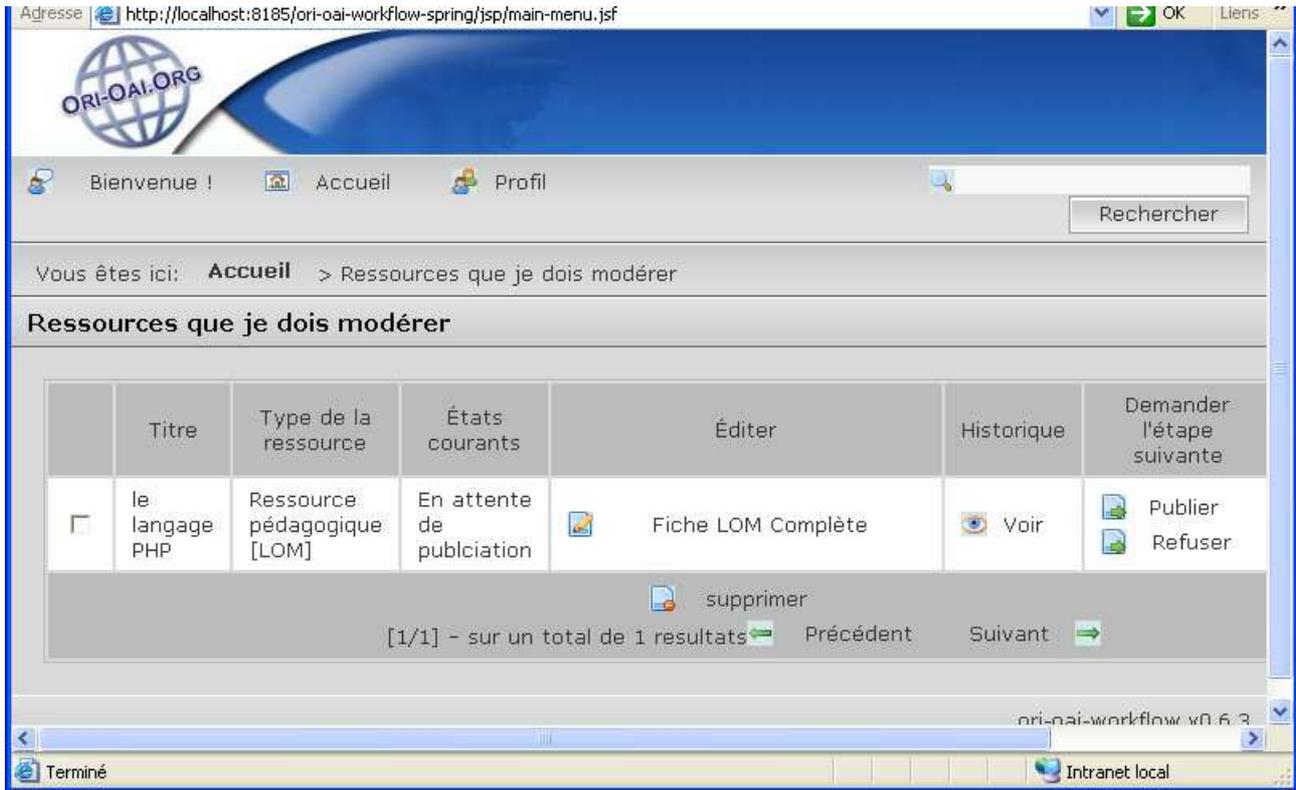
Pour passer à l'état « **en attente de publication** », cliquez sur « **demander à publier** ». Vous pouvez rédiger une observation, puis cliquez sur « **ok** ».

Déconnectez-vous de l'utilisateur ayant le rôle Auteur (clic sur Bienvenue, puis Quitté) et connectez-vous avec un utilisateur ayant le rôle **Modérateur (login_moderateur)**.

Pour publier le document, cliquez sur « **Accueil** » puis sur « **Ressources que je dois modérer [1]** ».

The screenshot shows a web browser window with the address bar displaying `http://localhost:8185/ori-oai-workflow-spring/jsp/welcome.jsf`. The page header includes the ORI-OAI.ORG logo and navigation links for 'Bienvenue !', 'Accueil', and 'Profil'. A search bar with a 'Rechercher' button is also present. The main content area is titled 'Liste des ressources en cours de traitement ou traitées' and contains a single entry: 'Ressources que je dois modérer [1]'. Below this, there is a section titled 'Référencer une ressource' with two entries: 'Ressource pédagogique [LOM]' and 'Document [DC]'. The browser's status bar at the bottom shows 'Terminé' and 'Intranet local'.

Cliquez sur « **Publier** », vous pouvez rédiger une observation, puis cliquez sur « **ok** ».



Vos données sont maintenant publiées.

3. Solution de stockage ESUP Portail et dépôt des ressources

i. Généralité

Il est possible dans ORI-OAI de référencer des ressources qui sont physiquement stockées sur un serveur quelconque. Dans ce cas on renseigne simplement dans le formulaire auteur la métadonnée permettant de localiser cette ressource.

Métadonnée **Identifier** dans le cas du Dublin-Core :

Identifier

Ajout d'un identifiant

Métadonnée **Localisation** dans le cas du LOM :

Localisation ?

Ajout d'une localisation

Néanmoins, le projet ORI-OAI propose aussi un composant de stockage pour assurer un stockage pérenne des documents et aussi offrir des mécanismes sophistiqués de contrôle d'accès aux ressources, notamment le support de la fédération d'identités permettant des autorisations entre établissements.

De plus, les futures versions du projet ORI-OAI vont offrir plus d'ergonomie en couplant plus fortement le dépôt du document et la saisie des métadonnées sur ce document. Il sera toujours possible d'utiliser un autre stockage (notamment parce que l'on peut toujours avoir besoin de référencer des ressources externes au système) mais l'utilisation du stockage proposé par ORI-OAI sera « bien intégrée ».



Note : Dans la version 1.0 de ORI-OAI, qui fait l'objet de ce document, cette intégration entre le stockage et la saisie des métadonnées n'est pas encore présente.

Dans la version 1.0 de ORI-OAI la solution de stockage préconisée est le serveur WevDAV distribué dans le cadre du projet Open Source ESUP-Portail.org.

Note : Ce serveur est utilisable indépendamment d'un environnement portail ESUP-Portail.

ii. Le serveur WebDAV ESUP-Portail

Le serveur de stockage proposé aujourd'hui par le consortium ESUP Portail répond à un certain nombre de standards et, à ce titre, peut interagir avec des outils compatibles avec ces standards :

- WebDAV
 - Le serveur est compatible WebDAV. Il est donc possible d'utiliser un client WebDAV pour déposer, renommer ou supprimer des ressources ou créer des répertoires.
 - Les clients webDAV utilisables sont par exemple : des outils intégrés au système d'exploitation (Les Dossiers Web de windows¹, DavFS pour Linux, etc.), des outils en ligne de commandes (cadaver par ex.), des outils graphiques en java (DAV Explorer), etc.
 - Pour accéder en lecture à une ressource ou naviguer dans la structure de répertoires un simple navigateur web est suffisant.
- ACP
 - En ce qui concerne le positionnement des droits le serveur supporte le protocole ACP (Access Control Protocol) qui est une extension de WebDAV. Là encore, il est possible d'utiliser des produits compatibles avec ACP pour manipuler les droits sur le serveur.
 - Parmi les outils utilisables on retrouve DAV Explorer, ou le plugin eclipse WebDAVPilot²

Une des particularités du serveur est aussi d'offrir plusieurs modes d'identification. De plus, comme WebDAV est une extension de http il est classique de pouvoir trouver un mode d'accès anonyme aux ressources publiques où aucune information d'identification n'est demandée à l'utilisateur.

Les modes d'identification supportés sont les suivants :

- unauthenticated : unauthenticated est le terme utilisé en protocole WebDAV pour désigner un utilisateur faisant une requête sans s'être identifié au préalable. C'est typiquement le cas lors l'un accès anonyme sur une ressource publique. Le mode d'identification unauthenticated du serveur WebDAV ESUP-Portail est un mode d'identification un peu spécial dans la mesure où il ne fait rien.
- Trusted : Ce mode permet d'être reconnu en tant que n'importe quel utilisateur à partir du moment où l'on utilise un mot de passe spécifique connu par le serveur (secret partagé entre le serveur et le client). Typiquement ce mode d'identification est utilisé dans le dialogue entre le portail (tiers de confiance) et le serveur WebDAV. Il est possible de limiter ce mode d'identification à quelques adresses IP bien spécifiques (celles des machines portail typiquement).
- LDAP : Ce mode permet de s'identifier en utilisant un nom d'utilisateur et un mot de passe stocké dans un annuaire LDAP.
- CAS : Ce mode permet de s'identifier en utilisant un service de SSO CAS.
- Shibboleth : Ce mode permet de s'identifier dans le cadre d'une fédération d'identités Shibboleth.

On voit qu'il est possible d'accéder à une même ressource de différentes façons et que suivant le cas d'utilisation on préférera utiliser tel ou tel mode d'identification.

¹ Les Dossiers Web installés par défaut dans Windows ne fonctionnent pas correctement et sont limités fonctionnellement (pas de support de https par ex.). Suivant que MS Office est installé ou pas ce composant peut avoir été mis à jour. Pour être sûr d'avoir une version à jour se référer à <http://www.microsoft.com/downloads/details.aspx?familyid=17c36612-632e-4c04-9382-987622ed1d64&displaylang=fr>

² Cf. http://www.s-und-n.de/index.php?option=com_content&task=view&lang=en&id=72&Itemid=117



Exemples :

- LDAP pour un accès en https via un dossier Web Windows sur une ressource non publique
- CAS pour un accès via un navigateur web sur une ressource non publique
- Shibboleth pour un accès via un navigateur sur une ressource non publique dans le cas où l'utilisateur qui tente d'accéder à la ressource n'est pas connu dans le SI de l'établissement qui héberge la ressource.
- Trusted pour les besoins du canal Stockage, par exemple, si ce dernier est installé dans un portail
- unauthenticated pour accéder via un navigateur Web à une ressource publique.

Pour permettre ce choix de mode d'identification on peut utiliser différentes solutions (détection du navigateur utilisé, paramètre passé dans la requête, etc.). Néanmoins, la solution préconisée est d'utiliser différents noms IP pour le même service WebDAV. Exemple :

- serveur.univ.fr (pour unauthenticated), serveur-ldap.univ.fr (pour LDAP), serveur-cas.univ.fr (pour CAS), etc.

Même si ce mécanisme est intéressant fonctionnellement on voit aussi qu'il pose un problème de lisibilité, notamment quand on veut référencer l'adresse d'une ressource. Il faudrait donner une adresse en serveur.univ.fr/mon_document pour une ressource publique que l'on voudrait rendre accessible au plus grand nombre et donner une autre adresse si cette ressource n'était pas en accès public. En plus, dans ce dernier cas, il faudrait donner une adresse en serveur-cas.univ.fr/mon_document si l'on a une infrastructure de SSO ou serveur-shib.univ.fr/mon_document si l'on a aussi une infrastructure shibboleth et que l'on veut autoriser l'accès à des personnes extérieures à l'établissement.

Pour pallier cette difficulté le serveur dispose d'un mécanisme de redirection automatique, détectant automatiquement si une ressource n'est pas en accès public, lui permettant de basculer dans un mode nécessitant identification. De plus, il est possible de forcer, suivant les besoins ou l'infrastructure du site, le mode d'identification à utiliser par défaut dans ce cas. Pour l'utilisateur cela se traduit par l'utilisation systématique d'une seule forme d'adresse en serveur.univ.fr/mon_document

Ce document ne couvre pas la partie installation du serveur WebDAV ESUP-Portail et son paramétrage. Pour plus d'information se référer à la documentation d'installation³ et la documentation de référence sur les filtres d'identification⁴.

iii. Le canal stockage ESUP

Le Canal stockage ESUP est un outil permettant de manipuler des ressources présentes sur des serveurs compatibles CIFS (serveur samba ou Windows NT et XP par exemple), FTP et WebDAV. Il est aussi compatible ACP.

Le canal peut accéder simultanément à plusieurs serveurs ou portions de serveurs. Ces différents serveurs ou portions de serveurs sont présentés de façon similaire à l'utilisateur en utilisant la notion d'espace.

Le canal a été conçu comme l'outil de référence pour positionner les droits sur un serveur WebDAV ESUP-Portail. Il est notamment pensé pour simplifier le positionnement des droits et garantir de ne pas supprimer de façon abusive des droits. En effet avec ACP un administrateur peut très bien positionner un droit disant qu'il ne pourra plus écrire sur un répertoire ! Dans ce cas, il faut intervenir sur la machine physique et ensuite redémarrer le serveur WebDAV pour corriger le problème.

Malheureusement le canal Stockage ESUP-Portail ne peut, à l'heure actuelle, fonctionner que dans un environnement portail ESUP-Portail (Ce point évoluera prochainement). Ceci étant, il nous semble intéressant d'en parler dans ce document pour les personnes qui disposent d'un environnement ESUP-Portail.

Le positionnement des droits par le canal n'est pas traité ici. Seule la configuration du canal pour offrir de l'ergonomie aux utilisateurs est présentée. Pour plus d'informations sur l'utilisation du canal ce référer à la documentation de référence⁵ et notamment le paragraphe 1.2.9 traitant du partage d'un dossier.

³ http://sourcesup.cru.fr/esup-webdav-srv/current/ESUPWDServer_installation.html

⁴ <http://sourcesup.cru.fr/esup-webdav-srv/current/authenticationLayer.html>

⁵ http://www.esup-portail.org/consortium/espace/Stockage_2F/CStockage/current/utilisation.html

iv. Obtenir l'URL à référencer

Comme nous l'avons vu dans les généralités nous avons besoin de saisir dans les métadonnées l'adresse URL de la ressource.

Si vous voulez référencer une ressource –généralement en accès public– qui n'est pas sur un serveur WebDAV ESUP il vous suffit, si vous ne connaissez pas l'adresse par cœur, d'utiliser un navigateur web pour naviguer jusqu'à la ressource, de copier son adresse dans le presse papier puis de revenir sur le formulaire pour la coller.

Nous voyons ci-après ce qu'il est possible de faire quand on utilise un serveur WebDAV ESUP-Portail et suivant que l'on utilise ou non le canal stockage ESUP-Portail. On traite ici le cas général d'une ressource en accès public ou non et où l'on utilise le mécanisme de redirection automatique du serveur.

Avec le canal stockage

Il est possible de configurer dans le canal, via le fichier **properties/CStockage.xml**, un espace (notion de serveur ou portion de serveur) de type WebDAV.

Prenons un exemple de fichier de configuration :

```
<SERVER url="http://ori-oai-webdav-trusted.univ.fr:80"
  serverType="webdav"
  authenticationMode="trusted"
  login="{uid}"
  password="secret_partage">
  <PARAMETER name="browserAccessUrl" value="http://ori-oai-webdav.univ.fr:80"/>
  <PARAMETER name="aclNamespace" value="DAV:"/>
  <PARAMETER name="aclUserPrefix" value="/users/"/>
  <PARAMETER name="aclGroupPrefix" value="/roles/"/>
  <PARAMETER name="aclHiddenGroupsInTree" value="/roles/local"/>
  <PARAMETER name="aclSecurity" value="/roles/local/root"/>
  <SPACE key="webdav-oril"
    label="Serveur ORI-OAI"
    path="/files/referencement"
    sharing="complex"
    actionType="classic"
    showQuota="true"
    allowPublicSharing="true"
    browserAccessUrlVisibility="menu"
    manageShibUsers="true">
  </SPACE>
</SERVER>
```

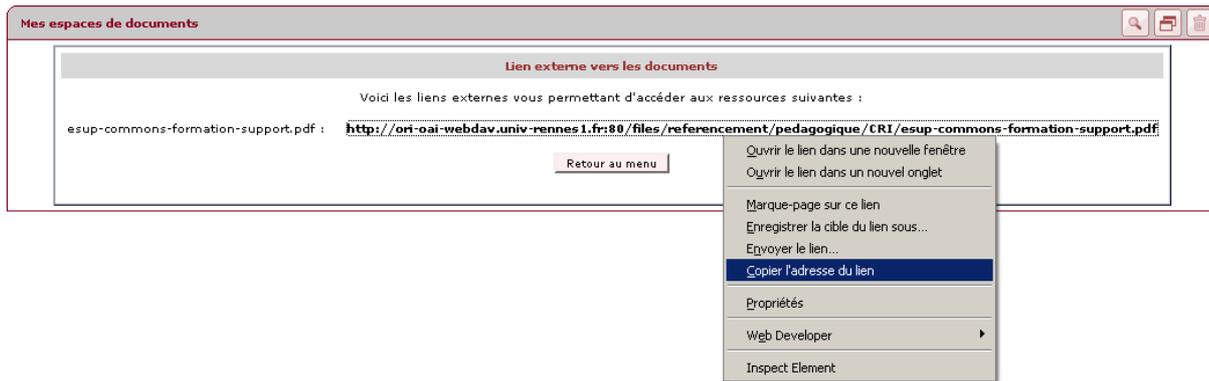
On voit ici que l'on est sur un serveur de `serverType="webdav"`, que l'on référence une adresse URL de type `serveur-trusted.univ-fr` afin d'utiliser un mode d'identification `trusted` pour la communication entre le canal contenu dans le portail et le serveur.

Un deuxième paramètre important est `<PARAMETER name="browserAccessUrl" value="http://ori-oai-webdav.univ.fr:80"/>`. Ce paramètre permet de donner l'URL de base qui servira à construire l'URL de la ressource à référencer. Typiquement on donne ici une adresse URL de type `serveur.univ-fr` afin d'utiliser le mode d'identification `unauthenticated` et éventuellement utiliser le mécanisme de redirection automatique du serveur WebDAV si la ressource n'est pas en accès publique.

Enfin le paramètre `browserAccessUrlVisibility` permet de préciser où et comment sera visible l'URL de la ressource à référencer. Les valeurs possibles sont :

- `public_only` (valeur par défaut si non le paramètre est absent) : L'URL n'est montrée que dans l'interface de partage lorsque l'on rend un dossier public
- `acl_only` : L'URL est toujours montrée dans l'interface de partage d'un dossier quels que soient les droits
- `menu` : On peut visualiser les URL d'un fichier, d'un ensemble de fichiers ou de dossiers depuis le menu principal du canal en cochant les ressources et en cliquant sur le bouton "Lien externe"

Dans notre cas d'utilisation (le référencement de ressources) la valeur à préconiser est "menu". Il sera alors possible de sélectionner une ou des ressources puis de copier les URL pour les coller ensuite dans le formulaire ORI-OAI de saisie des métadonnées. Exemple :



Note : Le canal stockage permet aussi de créer des dossiers, de positionner des droits, d'uploader des ressources. Ces différentes possibilités peuvent être de la responsabilité de différents acteurs dans l'établissement suivant les choix fonctionnels d'organisation. La description de ces fonctionnalités sort du cadre de ce document. Pour plus d'information se reporter à la documentation d'utilisation du canal (cf. page 54).

Sans le canal stockage

Sans le canal il faut faire le travail de construction de l'URL à référencer manuellement. Typiquement on va utiliser un navigateur et accéder au serveur en utilisant une adresse de type serveur-cas.univ-fr afin d'utiliser un mode d'identification nous donnant les droits suffisants pour naviguer à travers les différentes ressources. Une fois la ressource identifiée, il va falloir copier son URL (ex : <http://serveur-cas.univ-fr/rep1/rep2/res1.pdf>) ; coller cette URL dans le formulaire ORI-OAI de saisie des métadonnées ; NE PAS oublier de modifier cette URL de façon à la transformer afin d'utiliser le mode d'identification unauthenticated (ex : <http://serveur.univ-fr/rep1/rep2/res1.pdf>)

v. Avenir

Plusieurs des éléments logiciels décrits dans ce chapitre vont évoluer avec les versions futures de ORI-OAI :

- Le canal Stockage ESUP-Portail va évoluer pour pouvoir fonctionner dans un autre portail compatible avec la norme portlet (JSR 168) mais aussi indépendamment de tout portail.
- Le serveur WebDAV ESUP-Portail proposé aujourd'hui va évoluer. Il restera compatible WebDAV et deviendra, en plus, compatible JCR (Java Common Repository ou JSR 170) mais peut-être que les droits d'accès ne seront plus positionnables en protocole ACP. Néanmoins, il y aura de nouvelles interfaces pour positionner les droits. De plus, il devrait disposer de fonctionnalités complémentaires (gestion des versions, des liens, etc.). Avec ce nouveau serveur, seront livrés des outils de migration des contenus présents dans l'ancien serveur WebDAV ESUP-Portail.
- Il faut aujourd'hui faire du copier coller entre le canal stockage (ou un navigateur web) et le formulaire de saisie des métadonnées ORI-OAI. Il est prévu, dans la version 1.5 de ORI-OAI de supprimer cette opération par une meilleure intégration des deux outils (stockage d'une part et référencement d'autre part). Il sera par exemple possible d'uploader une ressource et d'initialiser son référencement lors d'une même opération dans un seul et même outil.

V. Configurations avancées

Nous allons maintenant aborder la configuration avancée de chaque module. Pour cela, nous distinguons plusieurs cas d'utilisation concrets.

A. Ajout d'une métadonnée

Dans ce cas d'utilisation, nous voulons configurer une métadonnée du format LOM qui n'a pas été configurée dans les différents modules. Cette métadonnée correspond à « semantic density (pédagogie : 5.4) ». Le but de ce chapitre sera donc d'ajouter les configurations nécessaires pour prendre en compte cette métadonnée dans tout le système.

La métadonnée «semantic density», que l'on traduira par « densité sémantique », est une métadonnée LOM de la catégorie « educational » (pédagogie). C'est une métadonnée qui ne peut admettre qu'un certain nombre de valeurs définies par le standard LOM :

- very low
- low
- medium
- high
- very high

1. Ajout d'un vocabulaire statique dans ori-oai-vocabulary

Les valeurs citées ci-dessus peuvent être potentiellement utilisées par plusieurs des modules ORI-OAI. On souhaite donc définir un nouveau vocabulaire dans le module ori-oai-vocabulary.

On donnera comme identifiant à ce nouveau vocabulaire **common_lom_educ_semantic_density**. Ce nouveau vocabulaire sera donné dans un fichier XML statique.

On crée un nouveau répertoire dans **ORI_HOME/src/ori-oai-vocabulary-svn/conf/properties** du module ori-oai-vocabulary que l'on va nommer **custom_vocabularies**.

Dans ce nouveau répertoire, on édite un fichier que l'on va nommer **common_lom_educ_semantic_density.xml** :

```
<orioaivocab xmlns="http://www.ori-oai.org/xsd/orioaivocab">
  <category id="semantic_density">
    <label lang="fr">Densité sémantique</label>
    <label lang="fr">Semantic density</label>
  </category>
  <category id="very_low">
    <label lang="fr">très bas</label>
    <label lang="en">very low</label>
    <value>very low</value>
  </category>
  <category id="low">
    <label lang="fr">bas</label>
    <label lang="en">low</label>
    <value>low</value>
  </category>
  <category id="medium">
    <label lang="fr">moyen</label>
    <label lang="en">medium</label>
    <value>medium</value>
  </category>
  <category id="high">
    <label lang="fr">élevé</label>
    <label lang="en">high</label>
    <value>high</value>
  </category>
</orioaivocab>
```



```
<category id="very_high">
  <label lang="fr">très élevé</label>
  <label lang="en">very high</label>
  <value>very high</value>
</category>

</category>
</orioaivocab>
```

Le vocabulaire via ce fichier XML est correctement défini. Il reste encore à le déclarer dans l'application.

Le point d'entrée des déclarations des différents vocabulaires correspond au fichier **ORI_HOME/src/ori-oai-vocabulary-svn/conf/properties/domain/domain.xml**. Les configurations du module ori-oai-vocabulary sont basées sur Spring et s'appuient donc sur le langage XML qui permet de définir les beans mis en place par Spring.

domain.xml importe en fait d'autres fichiers XML de configurations, cela afin d'ordonner et de différencier plus facilement les vocabulaires entre eux suivant leur type : vocabulaire statique, s'appuyant sur ldap, sur une BD SQL, etc.

On se propose ici d'ajouter un nouveau fichier de configuration que l'on nommera **customVocabulary.xml**. Aussi dans **domain.xml**, après le commentaire concernant **oriVocabulary.xml** par exemple on ajoute la ligne suivante:

```
<import resource="customVocabulary.xml" />
```

On crée ensuite **customVocabulary.xml** dans ce même répertoire **ORI_HOME/src/ori-oai-vocabulary-svn/conf/properties/domain**. Pour ce faire on peut bien sûr, selon le cas d'utilisation, s'inspirer des autres fichiers de configurations des vocabulaires : **indexingVocabulary.xml**, **alphabetVocabulary.xml**, etc. (ici, pour des vocabulaires statiques, on regarde plutôt **oriVocabularies.xml**).

Voici ce que doit donner simplement ce nouveau fichier **customVocabulary.xml** :

```
<?xml version="1.0" encoding="UTF-8" ?>

<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-
aop-2.0.xsd">

  <bean id="common_lom_educ_semantic_density"
    class="org.orioai.vocabulary.domain.providers.staticvoc.XmlStaticVocabularyProvider">

    <property name="xmlFileLocation"
      value="classpath:/properties/custom_vocabularies/common_lom_educ_semantic_density
.xml" />
  </bean>
</beans>
```

Pour prendre en compte ce nouveau vocabulaire, il faut maintenant déployer le module de vocabulaires :

```
ant deploy
```

et relancer le Tomcat correspondant au module de vocabulaires (histoire de relancer l'application de vocabulaires)

Le plus simple pour tester la récupération de ce vocabulaire (et donc vérifier qu'il a bien été configuré, bien pris en compte par le module) est de demander à l'éditeur de métadonnée de récupérer ce nouveau vocabulaire. Pour ce faire, vous demandez à votre navigateur l'URL suivante :

http://NOM_SERVEUR_ORI:8186/ori-oai-md-editor/ori-md-editor/vocab/common_lom_educ_semantic_density

Cela devrait vous renvoyer le XML correspondant.

Note : Le module de vocabulaires utilise un système de cache évolué. Aussi lorsque vous modifiez un vocabulaire, si vous souhaitez que celui-ci soit pris en compte directement, le plus simple est de stopper Tomcat, supprimer les fichiers **vocabularyServiceCache.data** et **vocabularyServiceCache.index** qui se trouvent dans le répertoire temporaire de Java/Tomcat (normalement le répertoire temp du Tomcat (répertoire tmp par défaut de Tomcat), cela peut aussi être sous /tmp (répertoire tmp par défaut de Java) ...

2. Saisie de la métadonnée dans ori-oai-md-editor

L'éditeur LOM auteur (Fiche LOM Simple) ne propose pas l'édition de cette métadonnée « semantic density ». On va ici ajouter cette possibilité. Qui plus est on va en faire une métadonnée obligatoire (cela aura à la fois des conséquences dans l'éditeur de métadonnées et dans la partie workflow comme nous allons le constater).

Le module ori-oai-md-editor est basé sur l'application Orbeon Forms, les « sources » de l'éditeur correspondent à des fichiers XML utilisés directement par l'application Orbeon Forms. Aussi ori-oai-md-editor a ceci de particulier (par rapport aux autres modules ORI-OAI) que les sources correspondent directement à l'application déployée. On modifie donc directement les fichiers dans l'application ori-oai-md-editor, c'est à dire ici le répertoire **ORI_HOME/tomcat-editeur/webapps/ori-oai-md-editor**.

Dans ce répertoire et ses sous-répertoires un certain nombre d'éléments concernent l'ensemble de l'application Orbeon Forms. Ce qui est spécifique à ori-oai-md-editor se trouve dans le répertoire **ORI_HOME/tomcat-editeur/webapps/ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor**/que nous nommerons dans la suite **MD_EDITOR_HOME**, c'est dans ce répertoire que nous allons travailler.

Les différents éditeurs de métadonnées se trouvent dans le répertoire xforms. Ici nous pourrions en créer un nouveau en copiant collant simplement le répertoire lom-author-light. On va se contenter ici de modifier ce formulaire. Le point d'entrée du formulaire a proprement parlé est xforms/lom-author-light/main-form.xhtml. Ce formulaire est accessible directement via l'url **http://NOM_SERVEUR_ORI:8186/ori-oai-md-editor/ori-md-editor/lom-author-light/blank**

Les modifications que l'on va effectuer seront aussitôt prises en compte dans l'interface (sans redémarrage nécessaire).

Dans **MD_EDITOR_HOME/xforms/lom-author-light/main-form.xhtml**, dans le bloc commenté « **Vocabulary instances** », on ajoute une nouvelle instance de vocabulaire :

```
<xforms:instance
  id="common_lom_educ_semantic_density"
  src="/ori-md-editor/vocab/common_lom_educ_semantic_density"
  xxforms:readonly="true"
  xxforms:shared="application"/>
```

main-form.xhtml inclut entre autre le fichier **content-xforms.xml**, dans lequel est défini le contenu du formulaire. On le modifie comme suit :

On propose d'ajouter le nouveau bloc correspondant à « **semantic density** » entre le bloc de « **Learning Resource Type** » et celui de « **Context** » (en prenant exemple sur ces derniers qui sont de même nature que « **semantic density** » :

```
<tr>
  <td class="table-legend">
    Densité Sémantique
  </td>

  <td>
    <widget:ori-block element="lom:semanticDensity" minOccurs="1" maxOccurs="1" parent-
element="lom:educational" preceding-elements="lom:interactivityType | lom:learningResourceType
| lom:interactivityLevel">
      <xforms:select1 ref="lom:value">
        <xforms:item>
          <xforms:label>[Select]</xforms:label>
          <xforms:value/>
        </xforms:item>

        <xforms:itemset
nodeset="xxforms:instance('common_lom_educ_semantic_density')/vocab:category/vocab:category">
          <xforms:label ref="vocab:label"/>
          <xforms:value ref="vocab:value"/>
        </xforms:itemset>
      </xforms:select1>
    </widget:ori-block>
  </td>
</tr>
```

```

</xforms:itemset>
</xforms:select1>
</widget:ori-block>
</td>
</tr>

```

On pourrait améliorer cela en ne mettant pas « **Densité Sémantique** » en dur mais en utilisant un vocabulaire approprié, en ajoutant de l'aide ... (comme pour les autres blocs déjà présents).

Ceci fait, et parce que l'on utilise ici la facilité des « **widget:ori-block** » amené par ori-oai-md-editor dans Orbeon Forms, il faudrait normalement vérifier (et ajouter si nécessaire) que l'on a bien dans les fichiers **MD_EDITOR_HOME/i18n/fr_FR.xml** et **MD_EDITOR_HOME/i18n/en_EN.xml** les balises du type :

```

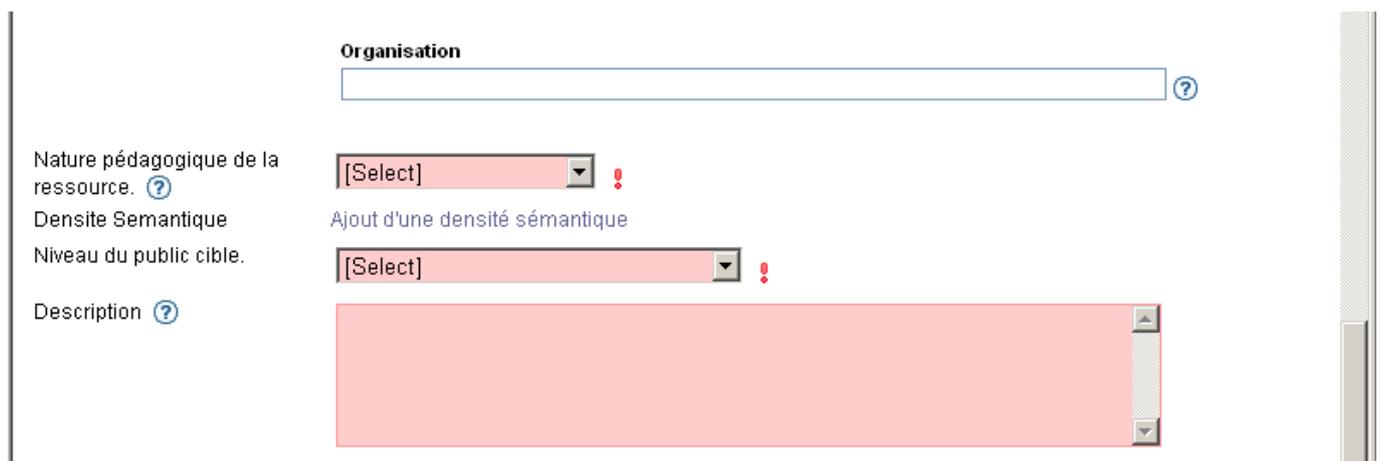
<add-semanticDensity>Ajout d'une densité sémantique</add-semanticDensity>
<del-semanticDensity>Suppression de la densité sémantique</del-semanticDensity>

```

Même si ici :

- elles existent déjà (car utilisées pour l'éditeur LOM complet)
- elles ne serviront pas, car on ne permet pas ici de supprimer la balise (minOccurs = 1) et on va faire en sorte que celle-ci soit présente par défaut (et donc que l'utilisateur n'ait pas besoin de l'ajouter, remarquez également que si la balise est présente par défaut, l'utilisateur ne pourra pas en ajouter une nouvelle : maxOccurs = 1 [cela est imposé par le schéma LOM]).

=> A ce stade, le formulaire **http://NOM_SERVEUR_ORI:8186/ori-oai-md-editor/ori-md-editor/lom-author-light/blank** doit vous donner la possibilité d'ajouter cette nouvelle balise.



Comme dit ci-dessus pour que la balise soit présente par défaut, il faut modifier ici le fichier utilisé pour initialiser le XML LOM édité: **MD_EDITOR_HOME/xml-blank/lom-author-light-blank.xml**.

On y ajoute simplement le bloc

```

<lom:semanticDensity>
  <lom:source>LOMv1.0</lom:source>
  <lom:value/>
</lom:semanticDensity>

```

au bon endroit.

Attention, ce fichier n'est pas utilisé lorsqu'on passe par le module de workflow pour éditer une fiche du workflow : dans le module de workflow, la fiche qui initialise un référencement est celle qui correspond au patron (template) de fiche :

le patron par défaut dans le module ori-oai-workflow (défini dans le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/spring-metadata-types.xml** via la propriété « **defaultXmlFile** »)

ou le patron qu'a éventuellement personnalisé l'utilisateur.

Reste enfin une dernière subtilité qui concerne le chargement des vocabulaires. ORI-OAI-MD-EDITOR permet de précharger puis recharger régulièrement les vocabulaires qui sont issus d'appels Web Services à un module ori-oai-



vocabulary. Pour ce faire, il utilise la possibilité de partager une même instance xforms entre plusieurs clients et formulaires ainsi qu'une tâche de rechargement qui s'opère régulièrement : d'où le paramètre vocab-cache-time dans le fichier config.xml.

Afin que le nouveau vocabulaire **common_lom_educ_semantic_density** soit pris en compte dans ces rechargements réguliers, il faut le spécifier à la fois dans **reload-vocab/load-vocab.xhtml** et **reload-vocab/unload-vocab.xhtml**.

Vous ajouterez ainsi comme il se doit et au bon endroit la ligne :

```
<xforms:instance
  id="common_lom_educ_semantic_density"
  src="/ori-md-editor/vocab/common_lom_educ_semantic_density"
  xxforms:readonly="true"
  xxforms:shared="application"/>
```

dans **reload-vocab/load-vocab.xhtml** et **reload-vocab/unload-vocab.xhtml**,

ainsi que la ligne

```
<xforms:dispatch name="xxforms-instance-invalidate"
  target="common_lom_educ_semantic_density"/>
```

Dans **reload-vocab/unload-vocab.xhtml**.

Note : Pour un exemple plus poussé d'ajout de métadonnées dans l'éditeur, voyez en Annexe la partie « **Ajout d'une métadonnée dans un formulaire d'ORI-OAI-MD-EDITOR** »

3. Prise en compte du caractère obligatoire dans ori-oai-workflow

Ajouter une condition sur la possibilité de passer à l'étape suivante dans le workflow se fait en modifiant le workflow en lui-même du module ori-oai-workflow.

Le workflow est la pièce maîtresse du module. Modifier un workflow, c'est modifier le fichier de configuration qui implémente en quelque sorte le diagramme d'état/transitions associé à un workflow donné.

Le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/spring-metadata-types.xml** qui permet de définir les types de métadonnées pris en compte dans ORI-OAI associe par défaut le type **ressource_pedago** au workflow **easy**.

Le workflow nommé **easy** est quand à lui défini dans le fichier de config **osworkflow ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/osworkflow/workflows/workflow_easy.xml** (cf le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/osworkflow/workflows.xml** qui permet de déclarer les workflow).

=> C'est donc ce fichier **workflow_easy.xml** que l'on modifie ici.

Ici la modification est simple : une condition **verifyXPathes** sur la transition « **Ask to Publish** » existe déjà. Il suffit donc d'ajouter le xpath qui va bien en argument de cette condition :

```
<arg name="xpath7">/lom:lom/lom:educational/lom:semanticDensity/lom:value[normalize-space(.)
!= '']</arg>
```

Comme après chaque modification des configurations, il faut redéployer l'application (ant deploy) et la redémarrer (on peut redémarrer le tomcat pour ce faire).

4. Indexation et recherche dans ori-oai-indexing

Pour indexer cette métadonnée il suffit de remplir le fichier **liusConfig.xml** du dossier **ORI_HOME/src/ori-oai-indexing-svn/properties**. Ce fichier est décomposé en trois parties.

La première concerne l'indexation. Toutes les métadonnées à indexer sont représentées sous forme de Xpath. La seconde permet d'indiquer les métadonnées sur lesquelles on souhaite faire des recherches. La troisième indique quelles métadonnées on souhaite pouvoir récupérer pour l'affichage des résultats.

Pour que cette métadonnée soit correctement indexée et que l'on puisse effectuer des requêtes sur celle-ci, les trois parties doivent être renseignées.

i. Ajout de la métadonnée dans la partie indexation

Cette partie est délimitée par la balise **<index>**. A l'intérieur de celle-ci se trouvent divers formats de métadonnées indexables dont celui qui nous intéresse : le format LOM. On peut le reconnaître grâce au commentaire `<!-- LOM -->`. Il existe beaucoup de métadonnées à indexer dans ce format. Pour plus de clarté ces dernières ont été triées par type (général, cycle de vie, méta-métadonnées, technique ...). La métadonnée que nous souhaitons ajouter se trouve dans la partie « Pédagogie ». A l'intérieur de cette partie, ajoutez la ligne suivante :

```
<uceneField name="%2F%2F%3Aeducational%2F%3AsemanticDensity%2F%3Avalue"
xpathSelect="//lom:educational/lom:semanticDensity/lom:value" type="Text" />
```

En consultant le fichier, vous devriez donc obtenir l'affichage suivant :

```
<!-- 4. Technique -->
<uceneField name="%2F%2F%3Atechnical%2F%3Alocation" xpathSelect="//lom:technical/lom:location" type="Text"/>
<uceneField name="%2F%2F%3Atechnical%2F%3Aformat" xpathSelect="//lom:technical/lom:format" type="Text"/>
<uceneField name="%2F%2F%3Atechnical%2F%3Asize" xpathSelect="//lom:technical/lom:size" type="Text"/>

<!-- 5. Pédagogie -->
<uceneField name="%2F%2F%3Aeducational%2F%3AlearningResourceType%2F%3Avalue" xpathSelect="//lom:educational/lom:learningResourceType/lom:value" type="Text" />
<uceneField name="%2F%2F%3Aeducational%2F%3AintendedEndUserRole%2F%3Avalue" xpathSelect="//lom:educational/lom:intendedEndUserRole/lom:value" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3Acontext%2F%3Avalue" xpathSelect="//lom:educational/lom:context/lom:value" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3AtypicalAgeRange%2F%3Astring%5B%40language%3D%27fr%27%5D" xpathSelect="//lom:educational/lom:typicalAgeRange/lom:string%5B%40language%3D%27fr%27%5D" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3Adifficulty%2F%3Avalue" xpathSelect="//lom:educational/lom:difficulty/lom:value" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3AtypicalLearningTime%2F%3Aduration" xpathSelect="//lom:educational/lom:typicalLearningTime/lom:duration" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3Adescription%2F%3Astring%5B%40language%3D%27fr%27%5D" xpathSelect="//lom:educational/lom:description/lom:string%5B%40language%3D%27fr%27%5D" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3Alanguage" xpathSelect="//lom:educational/lom:language" type="Text"/>
<uceneField name="%2F%2F%3Aeducational%2F%3AsemanticDensity%2F%3Avalue" xpathSelect="//lom:educational/lom:semanticDensity/lom:value" type="Text"/>

<!-- 6. Droits -->
<uceneField name="%2F%2F%3Arights%2F%3Adescription%2F%3Astring%5B%40language%3D%27fr%27%5D" xpathSelect="//lom:rights/lom:description/lom:string%5B%40language%3D%27fr%27%5D" type="Text"/>
<uceneField name="%2F%2F%3Arights%2F%3AcopyrightAndOtherRestrictions%2F%3Avalue" xpathSelect="//lom:rights/lom:copyrightAndOtherRestrictions/lom:value" type="Text"/>
```

- L'attribut « **xpathSelect** » correspond au xpath permettant de retrouver la métadonnée à indexer.
- L'attribut « **name** » donne un nom à cette métadonnée. Ce nom est en réalité l'encodage UTF-8 du xpath. Pour plus de facilité d'administration, il aurait été plus pratique d'utiliser le xpath tel quel comme nom de métadonnée. Mais ceci s'avère impossible car le xpath contient le caractère ":", caractère utilisé dans une requête Lucene. Par convention le groupe ORI-OAI a donc choisi d'encoder ce xpath en UTF-8, qui évite l'utilisation du caractère ":" tout en conservant dans le nom le chemin vers la métadonnée.
- L'attribut « **type** » définit le type de métadonnée. Trois types sont possibles : Text, ConcatDate et Vcard.
 - **ConcatDate** sera utilisé pour indexer une date.
 - **Vcard** sera utilisé si le champ à indexer contient une vCard.
 - **Text** sera choisi dans tous les autres cas.

ii. Ajout de la métadonnée dans la partie recherche

Cette partie est délimitée par la balise **<search>**. A l'intérieur de celle-ci se trouve la balise **<searchFields>** qui contient tous les noms de métadonnées séparés par une virgule. Pour permettre d'effectuer des requêtes sur cette métadonnée, ajoutez la ligne suivante en n'oubliant pas la virgule au début :

```
,%2F%2F%3Aeducational%2F%3AsemanticDensity%2F%3Avalue
```

Il s'agit du nom de métadonnée défini dans la partie indexation. Vous devriez obtenir l'affichage suivant en consultant le fichier **liusConfig.xml** :

```

<search>
  <!--
    org.apache.lucene.search.TermQuery
    org.apache.lucene.search.PrefixQuery
    org.apache.lucene.search.FuzzyQuery
  <queryTerm class="org.apache.lucene.search.WildcardQuery">
    <term field="creator"/>
  </queryTerm>
  <rangeQuery>
    <term field="creator"/>
    <booleanInclusive value="true"/>
  </rangeQuery>
  <queryParser>
    <defaultSearchField value="title"/>
  </queryParser>
  -->
  <!--If you want to search the field add it in the element searchFields -->

  <multiFieldQueryParser>
    <searchFields sep=",">%2F%2Fdc%3Atitle,%2F%2Fdc%3Asubject,%2F%2Fdc%3Acreator,%2F%2Fdc%3Adescription,%2F%2Fdc%3Apublisher,%2F%2Fdc%3Acontributor,%2F%2Fdc%
    %2F%2Fdc%3Aorgunit%2F%40id,%2F%2Fdc%3Aprogramname%2F%2Fdc%3Atext%5B%40language%3D%27fr-FR%27%5D,%2F%2Fdc%3Aprogramdescription%2F%2F,%2
    %2F%2Fdc%3Ageneral%2F%2Fdc%3Aidentifier%2F%2Fdc%3Aentry,%2F%2Fdc%3Ageneral%2F%2Fdc%3Atitle%2F%2Fdc%3Astring%5B%40language%3D%27fr-FR%27%5D,%2F%2Fdc%3
    %2F%2Fdc%3AlifeCycle%2F%2Fdc%3Acontribute%5Blom%3Arole%2F%2Fdc%3Avalue%3D%27author%27%5D%2F%2Fdc%3Aentity%2F%2Fdc%3AlifeCycle%2F%2Fdc%3Acontribute
    %2F%2Fdc%3AlifeCycle%2F%2Fdc%3Acontribute%5Blom%3Arole%2F%2Fdc%3Avalue%3D%27author%27%5D%2F%2Fdc%3Aentity%28name%29,%2F%2Fdc%3AlifeCycle%2F%2Fdc%3
    %2F%2Fdc%3AmetaMetadata%2F%2Fdc%3Acontribute%5Blom%3Arole%2F%2Fdc%3Avalue%3D%27creator%27%5D%2F%2Fdc%3Aentity%2F%2Fdc%3AmetaMetadata%2F%2Fdc%3A
    %2F%2Fdc%3Aclassification%2F%2Fdc%3AtaxonPath%5Blom%3Asource%2F%2Fdc%3Astring%3D%27Classification+UNIT%27%5D%2F%2Fdc%3Ataxon%2F%2Fdc%3Aentry%2F%2Fdc%3A
    %2F%2Fdc%3Aeducational%2F%2Fdc%3AsemanticDensity%2F%2Fdc%3Avalue
    </searchFields>
  </multiFieldQueryParser>
</search>

```

iii. Ajout de la métadonnée dans la partie affichage des résultats

Cette partie est délimitée par la balise **<searchResults>**. A l'intérieur de celle-ci se trouve la balise **<fieldsToDisplay>** qui contient les métadonnées à afficher. Ajoutez la ligne suivante :

```

<.luceneField name=" %2F%2Fdc%3Aeducational%2F%2Fdc%3AsemanticDensity%2F%2Fdc%3Avalue "
label=" //lom:educational/lom:semanticDensity/lom: value "/>

```

Vous devriez obtenir l'affichage suivant :


```
<!-- LOM -->
<search_menu key="lom">

  <!-- Recherche avancée -->
  <advanced_search key="advanced" file="lom_advanced.xml">
    ...
  </advanced_search>
</search_menu>
```

Ceci nous indique que la définition du formulaire de recherche avancée de l'onglet « Pédagogie » (LOM) se trouve dans le fichier **ORI_HOME/src/ori-oai-search-svn/properties/advanced/lom_advanced.xml**. En effet, tous les fichiers de configuration de recherche avancée de type **advanced_search** se trouvent dans le dossier **ORI_HOME/src/ori-oai-search-svn/properties/advanced**.

Ouvrez en mode édition ce fichier. On peut y voir que tous les champs de recherche **<field>** sont regroupés dans différents groupes **<group>**. Ces groupes sont distingués par différentes zones dans l'interface de recherche.

Le champ de recherche que nous voulons ajouter correspond à un critère pédagogique. Il faut donc ajouter un bloc **<field>** dans le groupe « peda » :

```
<field id="peda_semanticDensity" format="text"
vocabularyId="common_lom_educ_semantic_density">
  <metadata>//lom:educational/lom:semanticDensity/lom:value</metadata>
</field>
```

- **id** est l'identifiant de ce champ de recherche
- **vocabularyId** permet de dire sur quel vocabulaire se base la construction de la liste déroulante dans l'interface de recherche
- **metadata** correspond ici au xpath de la métadonnée concernée. A noter qu'il peut y avoir plusieurs balises metadata afin qu'un champ de recherche corresponde à plusieurs métadonnées.

Une fois cette configuration ajoutée, il faut définir le libellé du champ de recherche dans les fichiers **ORI_HOME/src/ori-oai-search-svn/properties/messages/forms_XX.properties** où XX correspond aux différents codes de langues supportés dans votre application.

Comme nous nous intéressons ici à la recherche avancée ayant comme clef « **advanced** » qui elle-même se trouve dans un menu de recherche plus global « **lom** », il faut ajouter les entrées suivantes dans les fichiers de définition des libellés :

```
menu.form.label.lom.advanced.peda_semanticDensity=Densité sémantique
menu.form.comment.lom.advanced.peda_semanticDensity=métadonnée ajoutée en formation
```

7. Affichage dans les résultats de recherche

On ajoutera dans cet exemple l'affichage de ce résultat uniquement depuis la **Recherche avancée** du menu de recherche **Multi-formats**. La même procédure devra être appliquée pour afficher cette métadonnée dans tous les autres modes de recherches.

Dans le fichier **ORI_HOME/src/ori-oai-search-svn/properties/config.xml**, il faut ajouter le bloc suivant :

```
<search_menu key="all">

  <!-- Recherche avancée -->
  <advanced_search key="advanced" file="all_advanced.xml">

    <result_fields>
      ...
      <result_field key="semanticDensity"
format="vocabulary:common_lom_educ_semantic_density">
        <metadata>//lom:educational/lom:semanticDensity/lom:value</metadata>
      </result_field>

    </result_fields>
    ...
  </advanced_search>
```



```
</search_menu>
```

Il faut alors indiquer le libellé de ce champ dans l'interface des résultats dans les fichiers **ORI_HOME/src/ori-oai-search-svn/properties/messages/menus_XX.properties**

Comme nous avons ajouté la métadonnée **semanticDensity** dans les résultats de la recherche avancée **advanced** du menu de recherche **all**, il faut ajouter l'élément suivant :

```
menu.result.all.advanced.semanticDensity=Densité sémantique
```

8. Affichage dans la notice

L'affichage et le rendu des fiches de métadonnées s'effectuent par une transformation XSLT commandée depuis la JSP **ORI_HOME/src/ori-oai-search-svn/webapp/WEB-INF/stylesheets/notice/notice.jsp**. Cette JSP fait le lien entre la fiche de métadonnée au format XML et une feuille de transformation XSL. Aussi, les fichiers de message multilingues étant gérés uniquement au niveau des JSP, **notice.jsp** fournit en paramètre à la XSL tous les libellés à afficher dans le rendu.

La transformation XSLT nécessite une XSL spécifique à chaque format de métadonnée. Cette XSL est spécifiée pour chaque format dans l'attribut **xsl** de la balise **config/notice_formats/format** du fichier de configuration **config.xml**.

Dans le cas du LOM pour notre cas, nous avons la configuration suivante :

```
<notice_formats>
  <format formatMetadataValue="..." prefix="lom" xsl="lom1.xsl" headXsl="lom.xsl">
    ...
  </format>
</notice_formats>
```

Cela signifie que nous utilisons la XSL **lom1.xsl** qui se trouve dans le dossier **ORI_HOME/src/ori-oai-search-svn/webapp/WEB-INF/xsl**.

Editez ce fichier.

La première étape consiste à ajouter en paramètre d'entrée du fichier l'appel à un paramètre fourni par la JSP au moment de la transformation XSLT. Ce paramètre correspond au libellé de la métadonnée qui sera affiché dans l'interface. A la suite des autres paramètres, ajoutez :

```
<xsl:param name="xsl.lom.semanticDensity"/>
```

Il faut maintenant faire appel à ce libellé et afficher la valeur de la métadonnée. Ajoutez le bloc suivant juste après le bloc similaire qui concerne la variable **peda_interactivityLevel** :

```
<xsl:variable name="peda_semanticDensity"
  select="//lom:educational/lom:semanticDensity/lom:value"/>
<xsl:if test="$peda_semanticDensity!=''">
  <xsl:value-of select="$xsl.lom.semanticDensity" /> :
  <xsl:value-of select="$peda_semanticDensity" />
</xsl:if>
```

Enregistrez ce fichier.

Il faut alors indiquer le libellé de ce champ dans la notice dans les fichiers **ORI_HOME/src/ori-oai-search-svn/properties/messages/xsl_XX.properties** :

```
xsl.lom.semanticDensity=Densité sémantique
```

Il faut maintenant éditer le fichier **notice.jsp** afin de donner le paramètre lié aux libellés à la XSL comme ceci à la suite des autres paramètres:

```
<x:param name="xsl.lom.semanticDensity">
  <fmt:message key="xsl.lom.semanticDensity"/>
</x:param>
```



Enregistrez ce fichier.

A cette étape, la valeur qui sera affichée dans la notice correspond à la valeur réelle du champ dans la fiche de métadonnées LOM. Afin de « traduire » cette valeur dans un terme compréhensible par tous, il faut paramétrer une traduction de cette valeur au moment de l'affichage de la notice en utilisant le vocabulaire défini au préalable **common_lom_educ_semantic_density**. Pour cela, il faut éditer le fichier **config.xml** et intervenir dans le bloc **config/notice_formats/format** et ajouter une balise **metadata** comme suit :

```
<notice_formats>
  <format formatMetadataValue="..." prefix="lom" xsl="lom1.xsl" headXsl="lom.xsl">
    ...
    <metadata format="vocabulary:common_lom_educ_semantic_density">
      //lom:educational/lom:semanticDensity/lom:value
    </metadata>
  </format>
</notice_formats>
```

Enregistrez ce fichier.

9. Mise à jour des modifications

Arrêtez le serveur tomcat :

```
ORI_HOME/tomcat-search/bin/shutdown.sh
```

Redéployez l'application :

```
cd ORI_HOME/src/ori-oai-search-svn
ant all-servlet
```

Relancez le serveur tomcat :

```
ORI_HOME/tomcat-search/bin/startup.sh
```

Et accédez aux interfaces de recherche pour visualiser les modifications.

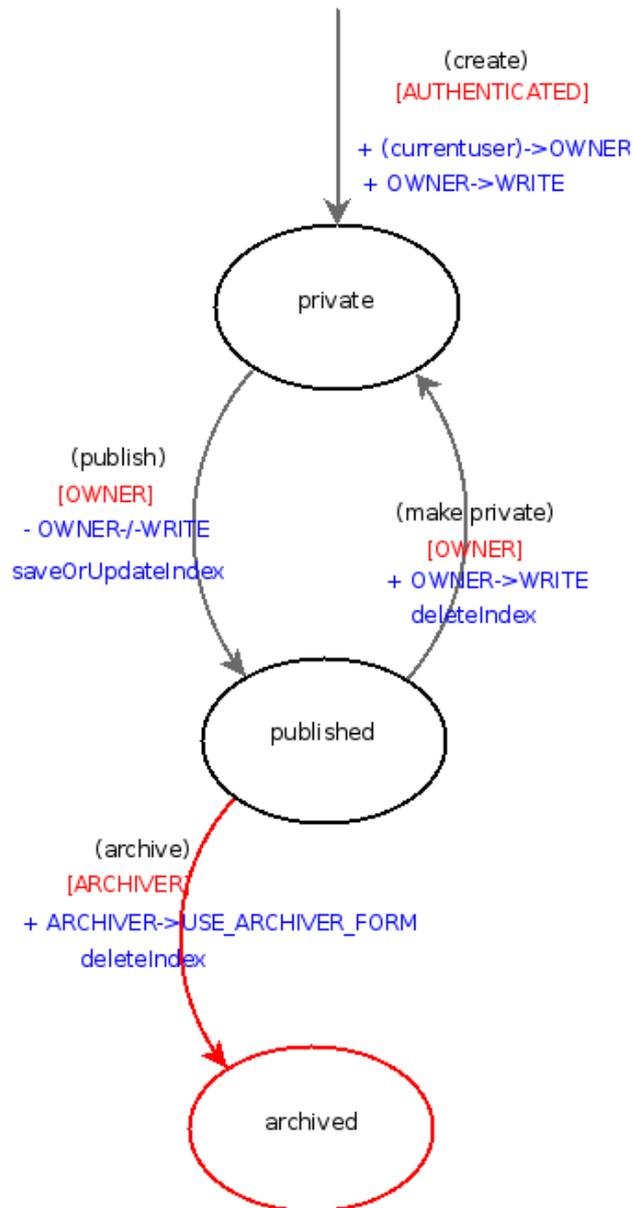
B. Modification du Workflow

On propose ici de modifier le workflow des types de métadonnées notés « document » (voir le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/spring-metadata-types.xml**).

D'un workflow à 2 états, on va en faire un workflow 3 états. Aux états « **privé** » et « **publié** », on y ajoute un nouvel état que l'on nomme « **archivé** ».

LEGEND

CONDITIONS	
hasPermission	permission form permission
hasRole	role
verifyXPathes	annotation xpath1 xpath2 ...
FUNCTIONS	
addPermission	+ role->permission
addRole	+ (currentuser)->role + user/group->role
deletePermission	- role/-/permission form - role/-/permission
deleteRole	- (currentuser)-/role - user/group-/role
saveXmlHistory	saveXmlHistory historyName
sendEmail	sendEmail roleDest
saveOrUpdateIndex	
deleteIndex	
(action)	état



Il faut donc rajouter ici l'état « **archived** » et la transition « **archive** ».

On voit également que l'on a conditionné la transition « **archive** » sur un nouveau rôle qui est celui d'« **ARCHIVER** ». Celui-ci n'existe pas par défaut il faut donc déclarer l'existence de ce nouveau rôle dans les fichiers de configuration Spring d'ori-oai-workflow.

Également on constate qu'on souhaite ajouter la permission de USE_ARCHIVER_FORM au rôle ARCHIVER lors de la transition « archive ». Il faut donc également déclarer cette nouvelle permission.

Enfin on note que l'idée de la permission **USE_ARCHIVER_FORM** est de donner accès à un nouveau formulaire pour les ressources de type « **document** ». Cela implique qu'il faut déclarer ce nouveau formulaire comme formulaire disponible pour le type « **document** », conditionner son utilisation par la permission **USE_ARCHIVER_FORM** fraîchement déclarée ... et enfin construire véritablement ce nouveau formulaire dans ori-oai-md-editor !



1. Ajout de permissions/rôles dans ori-oai-workflow

La documentation aidant, on doit ajouter les nouvelles possibilités de permissions/rôles que l'on souhaite pouvoir attribuer dans le module via le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/acegi/acegi-permissions.xml** (cf. <http://sourcesup.cru.fr/ori-workflow/installation.html#5.2.6>)

Ces permissions/rôles ont pour identifiant un nombre appartenant aux puissances de 2. Un certain nombre de permissions/rôles sont déjà prédéfinies dans l'application. On propose pour l'ajout de permissions/rôles personnelles à l'utilisateur de ne pas continuer directement à la puissance de 2 suivante mais de commencer à partir de 2 puissance 20 [$2^{20} = 1048576$] (afin de laisser au développeur la possibilité d'utiliser les 20 premières puissances de 2).

=> Donc on ajoute la permission

```
<prop key="1048576">USE_ARCHIVER_FORM</prop>
```

pour le bean **oriAclPermissionEntry**

et le rôle

```
<prop key="1048576">ARCHIVER</prop>
```

pour le bean **oriAclRoleEntry**.

2. On modifie le workflow en 3 étapes

Le workflow qui correspond au type « **document** » est le workflow **very_easy**: cf. **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/spring-metadata-types.xml**.

very_easy est défini par le fichier osworkflow: **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/osworkflow/workflows/workflow_very_easy.xml**

(cf. **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/osworkflow/workflows.xml**).

On modifie ainsi **workflow_very_easy.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE workflow PUBLIC "-//OpenSymphony Group//DTD OSWorkflow 2.6//EN"
"http://www.opensymphony.com/osworkflow/workflow_2_8.dtd">
<workflow>
  <initial-actions>
    <action id="100" name="Start Workflow">
      <restrict-to>
        <conditions type="AND">
          <condition type="spring">
            <arg name="bean.name">hasRole</arg>
            <arg name="mask">1</arg><!-- AUTHENTICATED -->
          </condition>
        </conditions>
      </restrict-to>
      <results>
        <unconditional-result old-status="Finished"
          status="Underway" step="1" />
      </results>
      <post-functions>
        <function type="spring">
          <arg name="bean.name">addRole</arg>
          <arg name="mask">2</arg><!-- Role 2 = OWNER -->
          <!-- no recipient -> current user -->
        </function>
        <function type="spring">
          <arg name="bean.name">addPermission</arg>
          <arg name="mask">20</arg><!-- Permission 20 = WRITE + DELETE
-->
          <arg name="recipient">2</arg><!-- Role 2 = OWNER -->
        </function>
      </post-functions>
    </action>
  </initial-actions>
</workflow>
```



```
        </post-functions>
    </action>
</initial-actions>
<steps>
    <step id="1" name="Private">
        <actions>
            <action id="1" name="Publish">
                <restrict-to>
                    <conditions type="AND">
                        <condition type="spring">
                            <arg name="bean.name">hasRole</arg>
                            <arg name="mask">2</arg><!-- Role 2 =
OWNER -->
                                </condition>
                            </conditions>
                        </restrict-to>
                    <results>
                        <unconditional-result status="Underway"
                            old-status="Finished" step="2" />
                    </results>
                    <post-functions>
                        <function type="spring">
                            <arg name="bean.name">deletePermission</arg>
                            <arg name="mask">20</arg><!-- Permission 20 =
WRITE + DELETE -->
                                <arg name="recipient">2</arg><!-- Role 2 = OWNER
-->
                                    </function>
                                <function type="spring">
                                    <arg name="bean.name">saveOrUpdateIndex</arg>
                                    <arg
name="idOriIndexing">indexingServicePublic</arg>
                                        </function>
                                    </post-functions>
                                </action>
                            </actions>
                        </step>
                    <step id="2" name="Public">
                        <actions>
                            <action id="2" name="Make Private">
                                <restrict-to>
                                    <conditions type="AND">
                                        <condition type="spring">
                                            <arg name="bean.name">hasRole</arg>
                                            <arg name="mask">2</arg><!-- Role 2 =
OWNER -->
                                                </condition>
                                            </conditions>
                                        </restrict-to>
                                    <results>
                                        <unconditional-result old-status="Finished"
                                            status="Underway" step="1" />
                                    </results>
                                    <post-functions>
                                        <function type="spring">
                                            <arg name="bean.name">addPermission</arg>
                                            <arg name="mask">20</arg><!-- Permission 20 =
WRITE + DELETE -->
                                                <arg name="recipient">2</arg><!-- Role 2 = OWNER
-->
                                                    </function>
                                                    <function type="spring">
                                                        <arg name="bean.name">deleteIndex</arg>
                                                        <arg
name="idOriIndexing">indexingServicePublic</arg>
                                                            </function>
                                                        </post-functions>
                                                    </action>
                                                        <action id="3" name="Archive">
                                                            <restrict-to>
```

```

                                <conditions type="AND">
                                    <condition type="spring">
                                        <arg name="bean.name">hasRole</arg>
                                        <arg name="mask">1048576</arg><!-- Role
1048576 = ARCHIVER -->
                                    </condition>
                                </conditions>
                            </restrict-to>
                            <results>
                                <unconditional-result old-status="Finished"
                                    status="Underway" step="3" />
                            </results>
                            <post-functions>
                                <function type="spring">
                                    <arg name="bean.name">addPermission</arg>
                                    <arg name="mask">1048576</arg><!-- Permission
1048576 = USE_ARCHIVER_FORM -->
                                </function>
                                <function type="spring">
                                    <arg name="recipient">1048576</arg><!-- Role
1048576= ARCHIVER -->
                                </function>
                                <function type="spring">
                                    <arg name="bean.name">deleteIndex</arg>
                                    <arg
name="idOriIndexing">indexingServicePublic</arg>
                                </function>
                            </post-functions>
                        </action>
                    </actions>
                </step>
                <step id="3" name="Archived">
                    <!-- Attention, pour OsWorkflow un step sans action "termine" le workflow
=> ici on propose une action fictive cad conditionnée par une condition
impossible à réaliser -->
                    <actions>
                        <action id="4" name="Dummy Action">
                            <restrict-to>
                                <conditions type="AND">
                                    <condition type="spring">
                                        <arg name="bean.name">hasRole</arg>
                                        <arg name="mask">-1</arg> <!-- Dummy Role -->
                                    </condition>
                                </conditions>
                            </restrict-to>
                            <results>
                                <unconditional-result old-status="Finished"
                                    status="Underway" step="3" />
                            </results>
                        </action>
                    </actions>
                </step>
            </steps>
        </workflow>

```

On a ici comme le schéma l'indiquait, ajouté une transition (une *action* en langage osworkflow) et un état (un *step*) [notez, comme dit en commentaire, que l'on a dû définir une action fictive depuis cet état Archived].

Dès lors, après redéploiement et redémarrage de ori-oai-workflow, les « documents » sont associés à ce workflow.

3. On déclare un nouveau formulaire

Pour déclarer un nouveau formulaire on édite le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/spring-metadata-types.xml** :

Pour le bean dont le **metadataTypeId** est « **document** » donc, on ajoute à la liste des forms un **dcarchiveform** :

```
<property name="forms">
```

```
<list>
  <ref bean="dcform"/>
  <ref bean="dcarchiveform"/>
</list>
</property>
```

Et on définit ce nouveau bean **dcarchiveform** juste après le bean **dcform** par exemple :

```
<bean class="org.orioai.workflow.beans.Form" id="dcarchiveform">
  <property name="title" value="Fiche Archive - OAI DC"/>
  <property name="permissionMask" value="1048576"/> <!-- USE_ARCHIVER_FORM -->
  <property name="url" value="{mdeditor.url}/ori-md-editor/oaidc-archive-full/remote"/>
</bean>
```

Ici on indique que ce nouveau formulaire a pour URL **{mdeditor.url}/ori-md-editor/oaidc-archive-full/remote** (**mdeditor.url** étant une propriété spring globale définie dans le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/main-config.properties**), faut-il encore le construire dans **ori-oai-md-editor** !

4. Création du formulaire oaidc-archive-full dans ori-oai-md-editor

Comme indiqué dans une section précédente, le travail s'effectue dans le répertoire **ORI_HOME/tomcat-editeur/webapps/ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor**.

On peut dans un premier temps le tester en mode « éditeur distant » (sans passer par le module **ori-oai-workflow** pour accéder à l'interface web donc).

- On copie/colle **xforms/oaidc-full** en **xforms/oaidc-archive-full**
- On copie/colle **xml-blank/oaidc-full-blank.xml** en **xml-blank/oaidc-archive-full-blank.xml**
- On procède aux modifications souhaitées dans **xforms/oaidc-archive-full/main-form.xhtml** : on vous suggère ici de simplement supprimer toutes les balises **fieldset/widget** (dans le `<div class="maincontent">`) permettant d'éditer chaque champ, sauf celui de la description par exemple. On obtient donc un formulaire d'édition du format oaidc qui ne contient que le champ « description ».

=> Votre nouveau formulaire est prêt à fonctionner en mode distant et via le workflow

... sauf que nous n'avons pour l'instant donné à personne le rôle de **ARCHIVER** dans le module **ori-oai-workflow** ... c'est ce que nous allons faire maintenant.

5. Affectation d'un rôle à un groupe, création d'un groupe virtuel

L'affectation des permissions/rôles peut

- soit se faire au travers d'un workflow, lors d'une transition, et donc pour une fiche donnée,
- soit se faire de manière globale via les fichiers de configurations spring.

On prendra ici la 2ème option.

On souhaite ici que l'utilisateur dont l'identifiant est **justin** soit « **ARCHIVER** ». On préfère plutôt que d'affecter directement le rôle d'archiver à **justin**, l'affecter à une groupe **groups_archivers**. Justin sera membre de ce groupe.

Pour ce faire, comme on l'a fait lors de la configuration du module après notre première installation, on édite le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/acegi/acegi-authentication-additional-groups.xml**.

On ajoute une entrée à la **sourceMap** de **additionalGroupsFiltersMapping**

```
<entry key="groups_archivers">
<value>(&amp;(objectClass=Person)(uid=justin))</value>
</entry>
```



On affecte alors par défaut le rôle de **ARCHIVER** au groupe **groups_archivers** en ajoutant, dans le fichier le fichier **ORI_HOME/ori-oai-workflow-svn/conf/properties/spring/acegi/acegi-acls-root.xml**, et à la liste des rôles du bean **entriesRoot**, le bean suivant :

```
<bean class="org.orioai.workflow.beans.acls.OriAclRole">
<property name="mask" value="1048576"/> <!-- ARCHIVER -->
<property name="recipient" value="groups_archivers"/> <!-- groups_archivers est un groupe
additionnel ldap -->
</bean>
```

Pour que ces modifications soient prises en compte, faites la commande suivante depuis le dossier **ORI_HOME/ori-oai-workflow-svn**:

```
ant update-acls
```

6. Ajout d'une catégorie « Ressources publiées »

Afin que l'archiviste puisse voir les ressources qu'il va pouvoir archiver si il le souhaite, on propose ici de créer une nouvelle catégorie destinée à l'archiviste donc qui est « Ressources publiées ».

Cette catégorie, visible seulement par un utilisateur qui a le rôle d'archiviste, va donc regrouper toutes les ressources qui correspondent au workflow « very_easy » et qui sont dans l'état public (id=2).

Pour ce faire, on édite le fichier **conf/properties/spring/spring-categories.xml** et on ajoute aux **categories** du bean « **roleStepCategoryConfig** » un nouveau RoleStepCategory défini comme ceci :

```
<bean class="org.orioai.workflow.beans.RoleStepCategory">
  <property name="id"
    value="docs_all_published" />
  <property name="workflowNameStepIds">
    <map>
      <entry key="very_easy">
        <list>
          <value>2</value> <!-- Public -->
        </list>
      </entry>
    </map>
  </property>
  <property name="roleMaskList">
    <list>
      <value>1048576</value> <!-- ARCHIVER -->
    </list>
  </property>
  <property name="description"
    value="Ressources publiées" />
</bean>
```

7. Ajout d'une catégorie « Ressources archivées »

Enfin, afin que le propriétaire de la fiche ainsi que l'archiviste puissent voir les ressources archivées s'afficher dans une catégorie spéciale, on crée de même une nouvelle catégorie : « Ressources archivées ».

```
<bean class="org.orioai.workflow.beans.RoleStepCategory">
  <property name="id"
    value="docs_archived " />
  <property name="workflowNameStepIds">
    <map>
      <entry key="very_easy">
        <list>
          <value>3</value> <!-- Archived -->
        </list>
      </entry>
    </map>
  </property>
  <property name="roleMaskList">
    <list>
      <value>2</value> <!-- OWNER -->
      <value>1048576</value> <!-- ARCHIVER -->
    </list>
  </property>
</bean>
```



```
</list>
</property>
<property name="description"
  value="Ressources archivées" />
</bean>
```

Nos modifications sont maintenant normalement correctement en place.

Il reste bien sur pour qu'elles soient prises en compte à redéployer et redémarrer l'application.

VI. Généralités

A. Utilisation d'un frontal http Apache

Les services ORI-OAI nécessitent un serveur d'application JAVA pour fonctionner. Tomcat est proposé comme solution par défaut.

Utiliser un serveur http Apache en frontal de ce ou ces serveurs Tomcat apporte de nombreux avantages :

- Sécurité avec Le serveur http Apache installé en DMZ et seul autorisé à dialoguer sur un port réseau particulier avec un serveur Tomcat situé dans un VLAN non accessible depuis internet
- Tolérance aux pannes et répartition de charge en utilisant un module Apache spécifique de répartition logicielle (mod_proxy_balancer)
- Contrôle d'accès à certaines URL sensibles afin d'en protéger l'accès
- Etc.

1. Configuration

Voici un exemple type de configuration d'un site virtuel Apache 2.2 en frontal d'un module ORI-OAI :

```
<VirtualHost 100.100.200.1:80>
  ServerName ori-oai-indexing.univ.fr
  VirtualDocumentRoot /data/webapps/ori-oai-indexing.univ.fr
  ProxyPass / ajp://ori-oaisrv.univ.fr:9526/ min=0 max=100 smax=50 ttl=10 route=ori-indexing
  # On protege tout ici
  <Location "/">
    order deny,allow
    deny from all
    # Autorise vlan serveurs
    allow from 100.100.300
  </Location>
</VirtualHost>
```

Ici le serveur répond à l'adresse ori-oai-indexing.univ.fr sur le port 80 et dialogue avec le serveur Tomcat grâce au protocole AJP sur le port 9526 de la machine ori-oaisrv.univ.fr

Ici le serveur apache n'établit aucune connexion avec le serveur Tomcat par défaut mais seulement au fur et à mesure des demandes des utilisateurs (jusqu'à 50). Si la charge augmente avec plus de 50 utilisations simultanées, alors le serveur peut créer jusqu'à 100 connexions pour revenir ensuite au seuil de 50 connexions.

Le paramètre `route` permet de reconnaître des cookies marqués par le serveur Tomcat afin de pouvoir orienter la requête suivante de l'utilisateur vers le même serveur Tomcat. Ce paramètre n'est pas indispensable ici car l'on n'utilise pas de mécanisme de répartition de charge où il faudrait garantir qu'un utilisateur soit toujours dirigé vers le même serveur d'applications.

2. Protection des URL sensibles

Dans l'exemple de configuration ci-dessous l'ensemble du site est protégé grâce aux directives `order`, `deny` et `allow` qui s'applique sur le chemin `"/`". Cette configuration permet de ne rendre accessible le module d'indexation que depuis une plage d'adresses IP où se retrouvent les différents serveurs. Le module n'est pas accessible depuis Internet. En effet, ce module n'a pas vocation à offrir un service direct aux utilisateurs mais est seulement utilisé par les autres modules ORI-OAI.

Il est particulièrement important de protéger certaines adresses afin d'éviter :

- Les dénis de service
- L'appel incontrôlé par des utilisateurs malveillants aux différents Web Services des modules ORI-OAI.

3. Quelles URL faut-il protéger ?

Voici la liste des URL à protéger dans le cadre de la mise en œuvre des modules ORI-OAI :

- ??? A valider en Groupe ???
- # ori-oai-workflow → /md-editor/ori-md-editor/vocab
- # ori-oai-vocabulary → "/"
- # ori-oai-indexing → "/"
- # ori-oai-harvester → "/harvester/admin"

B. Gestion de l'index

Il est fortement recommandé de sauvegarder régulièrement votre index en sauvegardant le dossier **ORI_HOME/data/indexes/index**. Si votre index devenait inutilisable, il vous suffirait alors de supprimer le dossier index et de le remplacer par votre copie la plus récente. Un redémarrage de votre serveur Tomcat hébergeant le module d'indexation sera nécessaire.

Toutefois, cette solution n'est envisageable que dans le cas où aucune fiche n'a été saisie dans le workflow et où aucune moisson n'a été effectuée depuis le moment où l'index est devenu inutilisable.

C'est pour cela qu'il existe des procédures de restauration de l'index à partir des modules ori-oai-workflow et ori-oai-harvester.

Avant de lancer ces procédures, vous devez supprimer le dossier **ORI_HOME/data/indexes/index** et redémarrer le serveur Tomcat correspondant au module ori-oai-indexing.

1. Restauration de l'index des fiches de ori-oai-workflow

Pour lancer la restauration de l'index du module ori-oai-indexing à partir des fiches stockées en base de données par ori-oai-workflow-spring, vous pouvez utiliser la target ant **reindexall**.

```
ant reindexall
```

Cela va appeler pour chaque fiche marquée comme devant normalement être indexée (cette information est gardée en base de données) le module d'indexation afin que celui-ci indexe chacune de ces fiches.

Note : A partir de la version ori-oai-workflow-spring-1.1.0, le module ori-oai-workflow intègre lui-même un index local (qui fonctionne de manière similaire au moteur d'indexation de ori-oai-indexing) aussi une nouvelle target apparait **local-reindexall** pour la réindexation de l'index local, qu'il ne faut donc pas confondre avec **reindexall**.

2. Restauration de l'index des fiches de ori-oai-harvester

Pour lancer la restauration de l'index à partir des fiches moissonnées, suivre les étapes suivantes :

- supprimer les fichiers de l'index du module d'indexation

```
rm -rf ORI_HOME/data/indexes/index/*
```

- lancer la ré-indexation depuis le menu « Définitions » en appuyant sur le bouton « Tout ré-indexer » :

Récupération

Ré-indexer toutes les fiches moissonnées

Attention : il faut bien s'assurer d'avoir supprimé l'index avant de lancer la restauration !

Tout ré-indexer

Le moissonneur va alors envoyer toutes les fiches moissonnées au moteur d'indexation pour ré-indexer.

VII. Evolutions en cours de réalisation

De nouvelles fonctionnalités sont actuellement en cours de conceptualisation et de développement. Elles aboutiront à la version 1.5 du projet ORI-OAI qui devrait être disponible en juin 2008.

ORI-OAI est paramétrable pour gérer tout type de ressources numériques. La première version proposait la gestion des ressources pédagogiques, la gestion d'autres types de ressources sera proposée avec des configurations adaptées dans la version 2 du projet prévue pour octobre 2008: les thèses et les publications scientifiques de la recherche.

A. Version 1.5 – Nouvelles fonctionnalités

1. Module ori-oai-workflow

Le dépôt des documents va être intégré au module ori-oai-workflow pour qu'un utilisateur puisse lier les métadonnées qu'il saisit à une ressource à travers une même interface.

Lors du dépôt des documents, l'utilisateur pourra définir des droits d'accès à sa ressource : il pourra ainsi la partager en lecture et/ou en écriture avec un ou des utilisateur(s) particulier(s), un ou des groupe(s) d'utilisateurs locaux ou distants, ou le diffuser de façon plus large sur un intranet ou sur internet.

Des protocoles de communication vont également être mis en place pour permettre à une application extérieure de démarrer un workflow.

Par exemple dans le cas d'utilisation d'un LMS avec saisie de métadonnées liée à la ressource dans le LMS et mise en place du workflow simple à trois états pour les ressources pédagogiques, si l'utilisateur souhaite intégrer sa ressource à l'archive institutionnelle de l'établissement de manière plus pérenne et donc à une instance d'ORI-OAI, le protocole de communication pourra être utilisé par le LMS pour engager la ressource dans le workflow en enlevant la première étape de création auteur (qui a déjà été effectuée dans le LMS), la ressource se trouvant directement en état « en cours de publication » avec une demande de validation à un modérateur.

2. Moteur d'indexation

Une indexation plein texte des documents sera proposée pour permettre la recherche sur ce champ.

La localisation de la ressource donnée dans les métadonnées ne correspond pas toujours à un lien direct vers la ressource, mais peut correspondre à une notice intermédiaire d'une autre application. La ressource peut également être proposée avec un accès protégé. Pour permettre l'indexation des ressources dans ces cas, un crawler web va être mis en place.

3. Moteur de recherche

Des flux RSS vont être proposés pour s'abonner à une recherche particulière.

La personnalisation possible des interfaces de recherche selon le profil de l'utilisateur va être améliorée.

De nouvelles fonctionnalités de recherche vont être proposées comme

- la recherche en croisant des champs à l'aide d'opérateurs booléens,
- une nouvelle interface pour la recherche thématique, elle proposera l'ensemble des champs d'une classification sur une même page et permettra d'en sélectionner plusieurs,
- le choix des valeurs d'un champ par cases à cocher ou boutons radios dans le formulaire de recherche avancée.

4. Gestionnaire de vocabulaires

Le module ori-oai-vocabulary va être modifié pour permettre la gestion de vocabulaires distants et centralisés. Certains vocabulaires utilisés dans l'application sont gérés par des UNT, UNR ou de manière nationale ; il semble intéressant que les gestionnaires de ces vocabulaires soient mieux à même de mettre à jour ces vocabulaires et que les changements faits soient diffusés aux utilisateurs de ces vocabulaires sans que ceux-ci aient à se soucier de mises à jour.

5. Evolutions générales du projet

Dans le cadre de la gestion de la production des ressources des établissements, ORI-OAI va mettre en place un module de gestion d'identifiants uniques et d'URL pérennes. Ceci permettra de diffuser les ressources produites en garantissant un accès sur le long terme aux utilisateurs et en identifiant de manière plus globale chacune de ces ressources.

Un module statistique va également être intégré à l'application, en particulier pour traiter les statistiques liées à la recherche et à l'accès aux documents.

B. Version 2 – Nouveaux types de ressources

Des configurations vont également être proposées pour permettre la gestion des thèses et des publications scientifiques.

Pour la gestion des thèses, un workflow va permettre leur gestion en suivant l'arrêté ministériel du 7 août 2006 relatif aux modalités de dépôt, de signalement, de reproduction, de diffusion et de conservation des thèses ou des travaux présentés en soutenance en vue du doctorat (Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche). Le workflow proposera des éditeurs pour le format TEF, format de description des thèses françaises, basés sur les formats internationaux METS et Dublin Core. Un échange des ressources et des métadonnées avec l'application STAR est également prévu pour permettre le dépôt légal et l'archivage des thèses au niveau national.

La gestion des publications scientifiques est également prévue dans le cadre de la mise en place de l'archive institutionnelle de l'établissement. Une configuration de l'application sera proposée pour la publication des ressources scientifiques produites au sein de l'établissement, le moissonnage de métadonnées en provenance d'autres archives et l'échange de ressources et de métadonnées avec la plateforme nationale française de gestion des archives ouvertes (HAL).



VIII. Liens

Tous les composants développés dans ORI-OAI reposent sur les mêmes technologies:

Java

Comme langage de programmation

<http://www.java.com>

Spring

Conteneur léger qui sert de framework général à l'application

<http://www.springframework.org>

XML

Comme norme d'échange

<http://www.w3.org/XML>

Subversion

Pour le téléchargement des sources des modules

<http://subversion.tigris.org/>

[http://fr.wikipedia.org/wiki/Subversion_\(logiciel\)](http://fr.wikipedia.org/wiki/Subversion_(logiciel))

Certains composants utilisent des technologies spécifiques:

ORI-OAI-workflow

OsWorkflow

Moteur de Workflow

<http://www.opensymphony.com/osworkflow>

XForms/Orbeon OPS

Utilisé pour générer des formulaires dynamiques et ergonomiques en Ajax depuis des fichiers XForms. Ces formulaires dynamiques correspondent à des éditeurs de fiches de métadonnées XML

<http://www.orbeon.com/>

http://fr.wikipedia.org/wiki/Asynchronous_JavaScript_And_XML

<http://www.w3.org/Markup/Forms/>

JSF

Framework MVC utilisé pour la "présentation" du workflow.

Les implémentations "Apache MyFaces" (dont Tomahawk) et jenia sont utilisées.

<http://java.sun.com/javaee/javaserverfaces/>

<http://myfaces.apache.org/>

<http://www.jenia.org/>

Spring XMLDB/eXist

Utilisé pour l'accès au stockage XML des fiches (base de données eXist)

<http://springxmldb.sourceforge.net/>

<http://exist.sourceforge.net/>

Hibernate

Framework de mapping objet-relationnel (MySql)

<http://www.hibernate.org/>

<http://www-fr.mysql.com/>

Acegi Security

Pour sécuriser l'application

<http://www.acegisecurity.org/>



XFire

Pour la communication entre Spring et Orbeon OPS, ainsi que la communication inter-modules
<http://xfire.codehaus.org>

ORI-OAI-harvester

OAI-PMH

Protocole d'échange des fiches de métadonnées
<http://www.openarchives.org/OAI/openarchivesprotocol.html>

OCLC harvester

Bibliothèque OAI pour Java
<http://www.oclc.org/research/software/oai/harvester2.htm>

Spring XMLDB/eXist

Utilisé pour l'accès au stockage XML des fiches (base de données eXist)
<http://springxmldb.sourceforge.net/>
<http://exist.sourceforge.net/>

Quartz

Pour la programmation des moissons
<http://www.opensymphony.com/quartz/>

Struts-Tiles

Pour les interfaces graphiques
<http://struts.apache.org/1.x/struts-tiles/>

ORI-OAI-repository

OAI-PMH

Protocole d'échange des fiches de métadonnées
<http://www.openarchives.org/OAI/openarchivesprotocol.html>

OCLC OAICAT

Pour l'exposition OAI des fiches de métadonnées
<http://www.oclc.org/research/software/oai/cat.htm>

ORI-OAI-indexing

Lucene

Moteur d'indexation et de recherche
<http://lucene.apache.org/java/docs/index.html>

LIUS

Framework d'indexation Java basé sur Lucene
<http://www.bibl.ulaval.ca/lius/>

Luke

Client de consultation d'un index Lucene
<http://www.getopt.org/luke/>

ORI-OAI-search

Spring MVC

Utilisé pour la présentation
<http://www.springframework.org>

Lucene

Pour le format de requêtes
<http://lucene.apache.org/java/docs/index.html>



XSLT

Pour la présentation des fiches de métadonnées

<http://www.w3.org/TR/xslt>

ESUP-serveur-WebDAV

WebDAV

Comme norme d'échange des données

<http://www.webdav.org/>

Jakarta/Slide

Comme socle

<http://jakarta.apache.org/slide/>

IX. Annexes

A. XML et les espaces de noms

Bien que fondamental en XML, le concept des espaces de noms n'est pas toujours bien compris et donc pas toujours bien respecté.

On tente ici d'expliquer par écrit ce concept, notamment au travers de l'exemple du LOM, LOMFR ...

1. Rappel

Les **espaces de noms** en **XML** sont une recommandation du W3C. Ils permettent d'identifier de manière unique les éléments et attributs des fiches XML. De fait, ils permettent de mettre en oeuvre le côté **eXtensible** de XML.

Aussi lorsque l'on définit un "langage XML" (on devrait d'ailleurs plutôt parler d' "**application XML**") dont les instances/fiches sont destinées à être des formats/vecteurs d'échange d'information entre plusieurs entités (logicielles ou non), il est important voir même impératif d'associer à ce nouveau langage un voir plusieurs espaces de noms XML (pré-existants et/ou nouvellement créés).

Un espace de noms s'applique aux **éléments** et **attributs** des fichiers XML. De manière pragmatique, chacun des éléments et attributs utilisés dans des fichiers XML d'échange devrait être déclaré dans un espace de noms (cela excepté les potentiels éléments/attributs **réservés** en XML).

Spécifier que des éléments et attributs sont dans un espace de noms donné, c'est garantir que ces éléments et attributs répondent au langage (implémentant un standard, une norme ...) **identifié par l'espace de noms**.

Un espace de noms est donné/représenté par un **identifiant** : une **URI** (Uniform Resource Identifier) qui est généralement une **URL** (il faut noter que cette URL ne pointe pas forcément sur un document existant : c'est bien un identifiant). Cet identifiant est utilisé via l'attribut réservé en XML `xmlns` qui permet dans une fiche XML de spécifier (via un **préfixe ou non** dans le cas de l'espace de noms par défaut) ensuite que l'on utilise tel élément ou tel attribut de tel espace de noms. Finalement, cela permet d'identifier de manière unique les éléments et attributs d'un "langage XML" : **c'est le couple espace de noms et nom de l'élément/attribut qui permet cette identification**.

L'exemple suivant :

```
<title xmlns="http://dummy.univ-rennes1.fr/xsd/dummy" />
```

est par exemple équivalent à :

```
<toto:title xmlns:toto="http://dummy.univ-rennes1.fr/xsd/dummy" />
```

mais non équivalent à :

```
<title xmlns="http://dummy.cndp.fr/xsd/dummy" />
```

L'élément **title** issu de **<http://dummy.univ-rennes1.fr/xsd/dummy>** et l'élément **title** issu de **<http://dummy.cndp.fr/xsd/dummy>** ne sont pas similaires, ils représentent 2 balises différentes qui répondent à 2 langages différents dans 2 espaces de noms différents.

Les différents logiciels qui manipulent des fichiers XML d'un langage XML donné doivent utiliser les espaces de noms au travers de parsers XML et autres outils pour assurer un réel support du langage concerné. En XML, il est possible de vérifier que les éléments et attributs d'un fichier XML sont techniquement bien formés par rapport aux espaces de noms auxquels ces éléments et attributs font référence. Pour ce faire, les technologies XML disposent de plusieurs outils : DTD, XML Schema, RELAX NG, Schematron ... Cette phase de validation est nécessaire dans le cadre de la validation d'une fiche XML par rapport à un langage donné.

2. LOM

Le standard LOM utilise naturellement un espace de noms spécifique LOM **<http://ltsc.ieee.org/xsd/LOM>** qui permet donc d'identifier les ressources LOM. Lorsqu'on utilise cet espace de noms LOM dans des fiches XML, on s'engage donc en quelque sorte à respecter le standard qu'est le LOM.

Pour pouvoir valider le format des fiches XML LOM, le LOM propose des schémas XML LOM disponibles en ligne.

Le schéma **<http://ltsc.ieee.org/xsd/lomv1.0/lom.xsd>** (qui correspond en fait à lomStrict.xsd) est la déclinaison « strict e » de LOM : ce schéma ne permet pas l'utilisation conjointe d'un autre espace de noms que l'espace de noms LOM dans une fiche dite LOM.

Le schéma lomLoose **<http://ltsc.ieee.org/xsd/lomv1.0/lomLoose.xsd>** autorise quand à lui une utilisation conjointe de l'espace de noms LOM et d'autres espaces de noms. Lorsque l'on décide de produire et manipuler du LOM ou une extension de celui-ci, la validation technique des fiches en lomLoose est nécessaire, elle n'est cependant pas suffisante.

Aussi dans le cadre d'un projet comme LOMFR qui vise à étendre le LOM, il est nécessaire de proposer des outils permettant de valider au mieux les fiches LOMFR.

3. LOMFR

Au niveau du LOMFR, l'espace de noms officiel associé à la norme est **<http://www.lom-fr.fr/xsd/LOMFR>**.

Le LOMFR étend le LOM en

- proposant de nouveaux éléments/attributs,
- proposant de nouveaux vocabulaires,
- rendant obligatoire/unique certains éléments/attributs,
- supprimant certains éléments/attributs,
- supprimant certains vocabulaires.

Dans une fiche LOMFR, l'espace de noms LOMFR est à utiliser conjointement à l'espace de noms LOM. Les éléments et attributs issus de LOM doivent naturellement être déclarés dans l'espace de noms LOM <http://ltsc.ieee.org/xsd/LOM>, les éléments et attributs issus du LOMFR doivent être donnés dans l'espace de noms LOMFR <http://www.lom-fr.fr/xsd/LOMFR>.

i. Outils

Afin d'aider au mieux les développements d'outils, la mise en place d'échange entre plusieurs entités etc. autour du LOMFR, le LOMFR proposera un certain nombre d'outils, de recommandations, de bonnes pratiques autour de la norme LOMFR.

Parmi eux, on trouvera des schémas et schématrons XML qui permettent de valider une fiche LOMFR : ils valident à la fois les éléments LOM (en prenant en compte les spécificités apportés par LOMFR : suppression/cardinalité de certains éléments/vocabulaires) ainsi que les éléments LOMFR.

4. Conclusion

Déclarer dans un XML que les éléments et attributs utilisés proviennent des espaces de noms LOM et LOMFR, c'est identifier le XML comme répondant à la norme LOMFR. La validation des fichiers XML est une condition nécessaire pour s'assurer que ces fichiers respectent bien ce contrat ainsi passé. La possible interopérabilité des fichiers XML LOMFR entre les plateformes d'échange (de type **OAI-PMH** par exemple), la compatibilité des outils d'éditeurs, etc. ne peuvent être réalités que si ces règles (et donc la prise en compte des espaces de noms) sont scrupuleusement respectées.

C'est également le respect de ces règles qui permet une interopérabilité des fiches LOM (dont les fiches LOMFR, et toutes autres fiches dont le langage étend LOM) au niveau des différents outils LOM.

B. Exploitation d'applications avec subversion - installation et mises à jour

Subversion (<http://subversion.tigris.org/>) est un outil de versioning puissant. Basé sur un modèle centralisé, il a l'avantage de rester relativement simple.

L'utilisation d'outil de versioning chez les développeurs s'est aujourd'hui réellement généralisé, au point que faire sans n'est maintenant plus envisageable.

Pour le métier de l'exploitant pourtant, l'utilisation de tels outils libres et open sources semble tarder à venir. Pourtant c'est une pratique qui peut faciliter grandement la vie de l'exploitant ... à l'utilisation l'exploitant devrait même être logiquement encore plus gourmand et exigeant quant aux possibilités de tels outils ...

Ce chapitre décrit l'installation d'un logiciel/application (ici ori-oai-workflow) en utilisant subversion (de sourcesup). Puis il montre comment subversion peut aider l'exploitant à mettre à jour son application tout en conservant les modifications qu'il aura apportées dans les configurations, voir dans le code de l'application !

Nous proposons ici d'installer ori-oai-workflow, de le configurer puis de le mettre à jour avec l'outil subversion.

Ce principe peut être bien sûr adapté à l'installation d'autres applications disponibles via subversion (comme les différents modules ORI-OAI, mais pas seulement) et aussi à d'autres outils que subversion comme CVS, mais aussi des outils plus puissants comme Mercurial (<http://www.selenic.com/mercurial/>), qui bien utilisé, amènerait d'ailleurs encore bien plus de possibilités que subversion pour l'exploitant ...

1. Checkout d'ori-oai-workflow 0.8.3

Plutôt que de télécharger une version d'ori-oai-workflow sous forme d'archive, on télécharge ici la version taguée dans le subversion de sourcesup.

Comprenez bien qu'ici on se place en tant qu'exploitant, on utilise donc le subversion en mode anonyme, donc sans avoir les droits d'écriture sur le repository.

De plus, on préfère l'utilisation de subversion en ligne de commande via la commande svn sur un serveur linux (a priori pour ce genre d'opération, vous serez connecté sur un serveur distant sans Interface Graphique, en SSH).

sourcesup propose 2 modes de navigation dans un repository subversion :

- via viewvc (<http://sourcesup.cru.fr/cgi/viewvc.cgi?root=ori-workflow>) : propose une navigation agréable dans le repository svn : coloration syntaxique du code, logs, etc
- directement via l'interface dav-svn (apache) de subversion avec l'url suivant <http://subversion.cru.fr/ori-workflow/> : c'est cette dernière url qui permet à l'outil client svn de venir synchroniser votre répertoire d'installation avec le repository svn distant.

```

Fichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn checkout http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.8.3
A   test
A   test/org
A   test/org/orioai
A   test/org/orioai/workflow
A   test/org/orioai/workflow/utils
A   test/org/orioai/workflow/utils/TestMathUtils.java
A   test/org/orioai/workflow/RemoteConfigTest.java
A   test/org/orioai/workflow/services
A   test/org/orioai/workflow/services/remote
A   test/org/orioai/workflow/services/remote/Testldap.java
A   test/org/orioai/workflow/services/remote/Testsmtp.java
A   test/org/orioai/workflow/services/remote/TestMDEditorFormsUrls.java
A   test/org/orioai/workflow/services/remote/TestVocabularyService.java
A   test/org/orioai/workflow/services/remote/TestIndexingServices.java
A   test/org/orioai/workflow/services/remote/TestExist.java

```

=> Pour l'exemple on récupère via svn ("**checkout**") la distribution ori-oai-workflow-0.8.3 grâce à l'url :

<http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.8.3/>

L'idée étant de mettre à jour ori-oai-workflow au fur et à mesure des nouvelles versions, dans un dossier nommé ori-oai-workflow-svn (en omettant donc le numéro de version).

Voici la commande **checkout** à faire depuis le dossier **ORI_HOME/src** :

```
svn checkout http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.8.3 ori-oai-workflow-svn
```

Copie d'écran :



```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn checkout http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.8.3 .
A   test
A   test/org
A   test/org/orioai
A   test/org/orioai/workflow
A   test/org/orioai/workflow/utils
A   test/org/orioai/workflow/utils/TestMathUtils.java
A   test/org/orioai/workflow/RemoteConfigTest.java
A   test/org/orioai/workflow/services
A   test/org/orioai/workflow/services/remote
A   test/org/orioai/workflow/services/remote/TestLdap.java
A   test/org/orioai/workflow/services/remote/TestSntp.java
A   test/org/orioai/workflow/services/remote/TestMdEditorFormsUrls.java
A   test/org/orioai/workflow/services/remote/TestVocabularyService.java
A   test/org/orioai/workflow/services/remote/TestIndexingServices.java
A   test/org/orioai/workflow/services/remote/TestExist.java

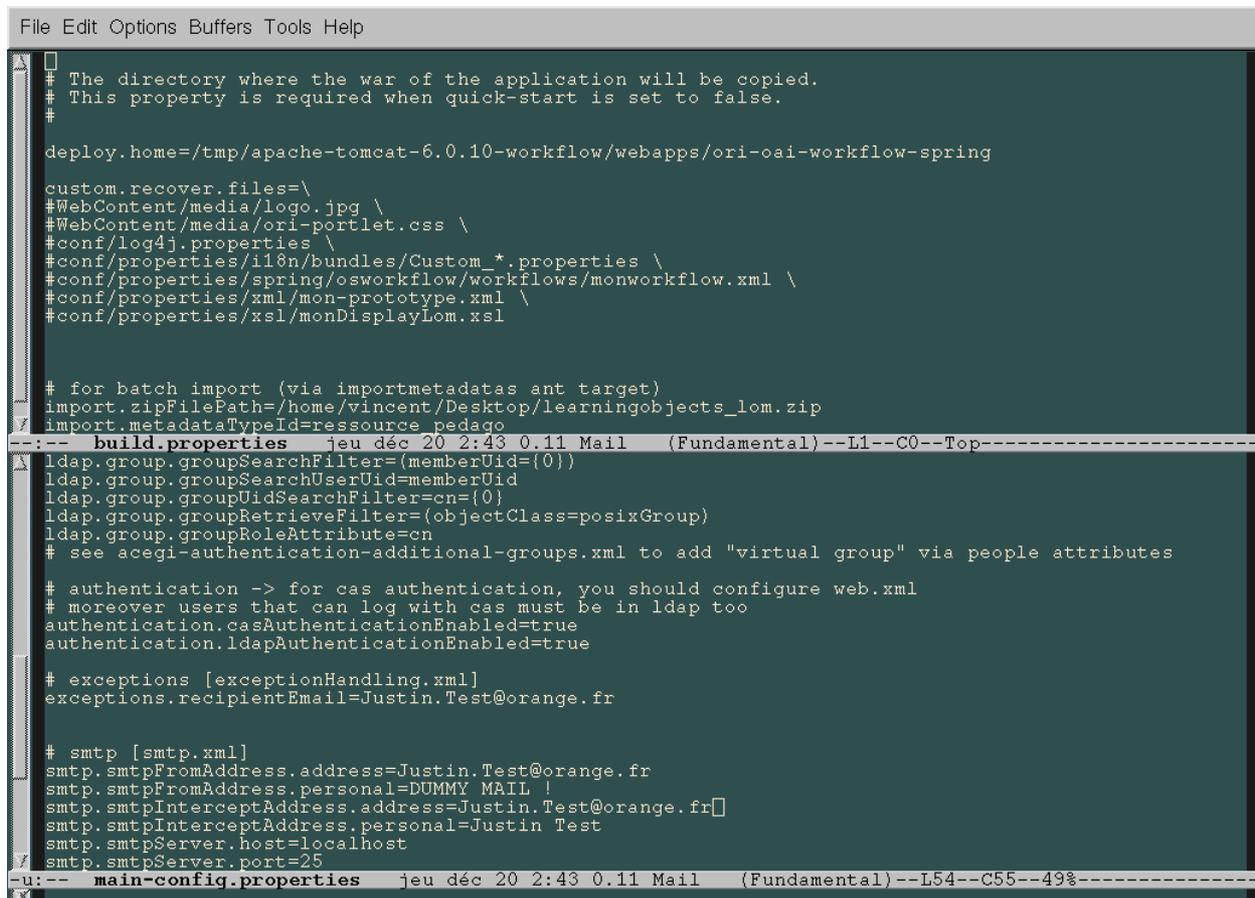
```

L'ensemble des fichiers est ainsi récupéré un à un et ajouté (le **A** signifiant **Add**) pour la première fois à votre répertoire d'installation.

2. Configurations d'ori-oai-workflow 0.8.3

Les documentations d'installation aidant, on modifie les fichiers **build.properties** et **conf/properties/main-config.properties** pour que cela corresponde à nos besoins propres.

Rapidement, une nouvelle capture d'écran montrant l'édition de ces fichiers :



```

File Edit Options Buffers Tools Help
# The directory where the war of the application will be copied.
# This property is required when quick-start is set to false.
#
deploy.home=/tmp/apache-tomcat-6.0.10-workflow/webapps/ori-oai-workflow-spring

custom.recover.files=\
#WebContent/media/logo.jpg \
#WebContent/media/ori-portlet.css \
#conf/log4j.properties \
#conf/properties/il8n/bundles/Custom_*.properties \
#conf/properties/spring/osworkflow/workflows/monworkflow.xml \
#conf/properties/xml/mon-prototype.xml \
#conf/properties/xsl/monDisplayLom.xsl

# for batch import (via importmetadatas ant target)
import.zipFilePath=/home/vincent/Desktop/learningobjects_lom.zip
import.metadataType=ressource_pedago
--- build.properties   jeu déc 20 2:43 0.11 Mail (Fundamental) --L1--C0--Top-----
ldap.group.groupSearchFilter=(memberUid={0})
ldap.group.groupSearchUserUid=memberUid
ldap.group.groupUidSearchFilter=cn={0}
ldap.group.groupRetrieveFilter=(objectClass=posixGroup)
ldap.group.groupRoleAttribute=cn
# see acegi-authentication-additional-groups.xml to add "virtual group" via people attributes

# authentication -> for cas authentication, you should configure web.xml
# moreover users that can log with cas must be in ldap too
authentication.casAuthenticationEnabled=true
authentication.ldapAuthenticationEnabled=true

# exceptions [exceptionHandling.xml]
exceptions.recipientEmail=Justin.Test@orange.fr

# smtp [smtp.xml]
smtp.smtpFromAddress.address=Justin.Test@orange.fr
smtp.smtpFromAddress.personal=DUMMY MAIL !
smtp.smtpInterceptAddress.address=Justin.Test@orange.fr
smtp.smtpInterceptAddress.personal=Justin Test
smtp.smtpServer.host=localhost
smtp.smtpServer.port=25
-u:-- main-config.properties   jeu déc 20 2:43 0.11 Mail (Fundamental) --L54--C55--49%-----

```

Une fois fait, svn est capable de nous résumer les modifications via la commande :

```
svn status
```

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn status
M      conf/properties/main-config.properties
M      build.properties
vincent@debian:/tmp/ori-oai-workflow-spring-svn$

```

Les fichiers listés avec **M** indique que ces fichiers ont été modifiés (**M** pour **Modify**).

Une commande `svn diff` permet de visualiser les différences apportées par rapport à la distribution originale. Exemple d'un **diff** sur **build.properties** :

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn diff build.properties
Index: build.properties
=====
--- build.properties      (révision 471)
+++ build.properties      (copie de travail)
@@ -3,7 +3,7 @@
 # This property is required when quick-start is set to false.
 #

-deploy.home=/opt/ori/tomcat/apache-tomcat-6.0.10-workflow/webapps/ori-oai-workflow-spring
+deploy.home=/tmp/apache-tomcat-6.0.10-workflow/webapps/ori-oai-workflow-spring

custom.recover.files=\
#WebContent/media/logo.jpg \
vincent@debian:/tmp/ori-oai-workflow-spring-svn$

```

Ce sont des fichiers de **diff** classiques qui en ressortent : les + et - indiquent si la ligne est ajoutée/modifiée par rapport à la distribution originale.

Bien sûr ici, et notamment pour le module ori-oai-workflow, l'exploitant va configurer un bon nombre de fichiers, modifier les css, les logos etc.

Ensuite il déploiera l'application dans un Tomcat (cf. le **build.properties**) ... et viendra le temps de la mise à jour ... !!

3. Mise à jour d'ori-oai-workflow : 0.8.3 -> 1.0.0

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn switch http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-1.0.0
A      LICENSE
G      conf/properties/main-config.properties
U      conf/properties/spring/common/misc/application.xml
U      CHANGELOG
A      README
U      build.xml
U      WebContent/media/interface/logo.jpg
U      WebContent/media/interface/background_logo.jpg
Actualisé à la révision 471.
vincent@debian:/tmp/ori-oai-workflow-spring-svn$

```

Un `svn info` nous permet de constater que nous sommes bien sur le tag 0.8.3, c'est à dire sur l'url: <http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.8.3>

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn info
Chemin : .
URL : http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.8.3
Racine du dépôt : http://subversion.cru.fr/ori-workflow
UUID du dépôt : 934bc96f-b41c-0410-b36d-ef89294fb87a
Révision : 471
Type de noeud : répertoire
Tâche programmée : normale
Auteur de la dernière modification : vbonyam
Révision de la dernière modification : 429
Date de la dernière modification: 2007-10-10 15:26:00 +0200 (mer, 10 oct 2007)
vincent@debian:/tmp/ori-oai-workflow-spring-svn$

```

Mettre à jour ori-oai-workflow, c'est ici le passer (on dit qu'on le **switch**) de 0.8.3 à 1.0.0.

Cette manipulation subversion peut avoir des effets non contrôlés, c'est une manipulation qui suivant les cas peut être délicat.

Pour plus de sûreté, on propose avant de faire un **switch** de sauvegarder l'ensemble du répertoire ori-oai-workflow-svn, on peut faire un tar-gz par exemple simplement :

```
tar czf ori-oai-workflow-svn-backup.tgz ori-oai-workflow-svn
```

On récupèrera la nouvelle url correspondant à la version 1.0.0 via un navigateur : ici, c'est <http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-1.0.0>

Donc depuis le répertoire **ori-oai-workflow-svn** on peut procéder au switch :

Le **U** nous indique ici les fichiers directement mis à jour,

Le **A** les fichiers ajoutés,

Le **G** les fichiers mis à jour qui ont nécessité une résolution de conflits car modifiés à la fois par l'exploitant et le développeur.

Les fichiers marqués en **G** sont donc usuellement à vérifier rapidement : on vérifie leur intégrité, on peut pour cela utiliser la commande `svn diff` pour voir les différences entre le fichier de l'exploitant et le fichier original de la version 1.0.0 d'ori-oai-workflow (la plupart du temps, on peut faire confiance à subversion et au développeur qui a conscience du procédé mis en œuvre et qui agit au niveau du développement en conséquence).

Ici il n'y a pas de conflit (marqué **C**) et donc tout s'est extrêmement bien passé. A priori il n'y a pas grand chose à faire de plus pour cette mise à jour (mis à part bien sûr suivre les indications de la documentation : faire pour le workflow un ant upgrade par exemple ici puis redéployer l'application).

L'avantage ici est que l'on n'a pas eu à recopier et répercuter soit même les modifications de configuration entre une ancienne et une nouvelle.

Cependant, et même si le développeur, en connaissance de cause, prend garde à faire en sorte que ces **switchs** se passent au mieux, il se peut que l'on rencontre parfois des "**conflits**".

4. Mise à jour d'ori-oai-workflow : conflits !

Comme dit ci-avant, usuellement lorsque le développeur y prend garde, il est normalement rare d'aboutir à un conflit. Mais cela arrive tout de même, notamment lorsque (contrairement au paragraphe ci-avant) à la fois

- un certain nombre de modifications entre 2 versions ont été faites côté développement
- un certain nombre de configurations/personnalisations ont été faites côté exploitation



```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn info
Chemin : .
URL : http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-0.6.4
Racine du dépôt : http://subversion.cru.fr/ori-workflow
UUID du dépôt : 934bc96f-b41c-0410-b36d-ef89294fb87a
Révision : 471
Type de noeud : répertoire
Tâche programmée : normale
Auteur de la dernière modification : vbonamy
Révision de la dernière modification : 408
Date de la dernière modification: 2007-09-26 12:01:45 +0200 (mer, 26 sep 2007)

```

Ici on se met dans une situation où on aura lors du switch un conflit.

Pour cela on se place comme un exploitant qui a chargé via svn le tag 0.6.4 de ori-oai-workflow et qui a modifié quelques fichiers de configurations et autre, notamment le fichier CSS **WebContent/media/ori-addon.css** :

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn status
M      conf/properties/spring/acegi/acegi-acls-root.xml
M      conf/properties/spring/osworkflow/workflows/workflow_easy.xml
M      conf/properties/main-config.properties
?      WebContent/media/ori-portlet.css_
M      WebContent/media/ori-addon.css
M      WebContent/media/ori-portlet.css
M      build.properties
  
```

On procède au switch vers le tag 1.0.0 directement :

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn switch http://subversion.cru.fr/ori-workflow/ori-oai-workflow-spring/tags/ori-oai-workflow-spring-1.0.0
U      test/org/orioai/workflow/services/remote/TestLdap.java
U      test/org/orioai/workflow/services/remote/TestSmtplib.java
U      test/org/orioai/workflow/services/remote/TestMdEditorFormsUrls.java
U      test/org/orioai/workflow/services/remote/TestVocabularyService.java
U      test/org/orioai/workflow/services/remote/TestIndexingServices.java
U      test/org/orioai/workflow/services/remote/TestExist.java
U      test/org/orioai/workflow/web/TestWIPerformAction.java
U      test/org/orioai/workflow/web/TestMainBeanWI.java
U      test/org/orioai/workflow/web/TestUserLoginBean.java
  
```

Dans la copie d'écran, on ne donne pas le listing complet, mais la plupart des fichiers sont notés **U**, d'autres **A**, quelques uns **G** et surtout le fichier **WebContent/media/ori-addon.css** est noté **C**.

On peut d'ailleurs retrouver les fichiers notés **C** via un `svn status` :

```

Eichier  Édition  Affichage  Terminal  Onglets  Aide
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ svn status
M      conf/properties/spring/acegi/acegi-acls-root.xml
M      conf/properties/spring/osworkflow/workflows/workflow_easy.xml
M      conf/properties/main-config.properties
M      build.properties
?      WebContent/media/ori-addon.css.mine
?      WebContent/media/ori-addon.css.2.r471
?      WebContent/media/ori-portlet.css_
?      WebContent/media/ori-addon.css.r471
C      WebContent/media/ori-addon.css
M      WebContent/media/ori-portlet.css
vincent@debian:/tmp/ori-oai-workflow-spring-svn$ █
  
```

L'édition de **ori-addon.css** pour une résolution manuelle du conflit est impératif.

Ici une modification a été faite en parallèle par le développeur et par l'exploitant. On peut retrouver les différentes versions de ce fichier dans les fichiers nommés **ori-addon.css.***

Dans le fichier **ori-addon.css** on retrouvera les caractères <<<<<<< , ===== et >>>>>>> nous permettant d'identifier les régions du ou des conflits.

```
File Edit Options Buffers Tools Help
background: transparent url( ../media/icons/group.png ) no-repeat left center;
border: none;
cursor: pointer;
text-align: left;
}
.ori-logout-button
{
padding : 2px 2px 2px 20px;
<<<<<<< .mine
background: red url( ../media/icons/door_out.png ) no-repeat left center;
border: solid 1px #999999;
=====
background: transparent url( ../media/icons/door_out.png ) no-repeat left center;
border: none;
>>>>>>> .r471
cursor: pointer;
text-align: left;
}
.ori-search-button
{
margin: 0px 1px 0px 1px;
padding : 1px 1px 1px 1px;
background: #f7f7f7 url( ../media/interface/background_button.png ) 0px 0px repeat-x;
border: solid 1px #999999;
-- ori-addon.css jeu déc 20 4:36 0.28 Mail (Fundamental)--L237--C13--77%
```

=> finalement on modifiera simplement ici

```
<<<<<<< .mine
background: red url( ../media/icons/door_out.png ) no-repeat left center;
border: solid 1px #999999;
=====
background: transparent url( ../media/icons/door_out.png ) no-repeat left center;
border: none;
>>>>>>> .r471
```

par

```
background: red url( ../media/icons/door_out.png ) no-repeat left center;
border: none;
```

par exemple.

Les conflits résolus, on supprimera alors les fichiers **ori-addon.css.***.

5. Conclusion

Bien utilisé à la fois au niveau du développeur que de l'exploitant, subversion peut devenir une arme redoutable pour l'exploitant également (et non plus pour le développeur uniquement).

Pour certaines applications comme ori-oai-workflow qui permet une configuration fine et poussée via la modification de fichiers de configuration, l'exploitant a tout intérêt à utiliser et exploiter au maximum subversion ... côté exploitation !

C. Ajout d'une métadonnée dans un formulaire d'ORI-OAI-md-editor

XML permet d'étendre voir de concevoir entièrement des langages (applications) XML. En respectant les espaces de noms, on peut ainsi arriver à rester complètement compatible avec un langage donné tout en ajoutant des éléments et attributs propres.

Pour plus d'informations la dessus, revoyez **l'Annexe A** : Xml et les Espaces de noms dans ce même document.

Ici, on montre comment on peut ajouter une métadonnée à un XML LOM dans un espace de noms propre et comment on peut ensuite proposer à l'utilisateur d'éditer ce XML dans le formulaire LOM de l'éditeur de métadonnées ORI-OAI : ORI-OAI-MD-EDITOR.

→ en bref, ce chapitre explique comment on peut facilement étendre/modifier (et en fait créer) un formulaire dans ORI-OAI-md-editor basé, on le rappelle, sur Orbeon Forms.

1. Espace de noms Rennes1 et exemple d'XML LOM Rennes1

Sur un ori-oai-md-editor d'installé et qui fonctionne, on va donc ici montrer comment créer un nouveau formulaire sur la base du formulaire LOM pour lui ajouter une nouvelle métadonnée.

Ici on se place dans la position de l'Université de Rennes1 par exemple qui veut rajouter une métadonnée interne pour "taguer" ses ressources : l'Université faisant partie de chacune des UNT, on souhaiterait en effet lors de l'indexation d'une ressource pédagogique préciser à quel UNT la ressource peut correspondre. Cela permet de cibler un certain publique ... pour rester souple, on utilisera plutôt un nom de balise comme **targetPublic** par exemple.

Voici un exemple de fichier XML LOM étendu via un espace de noms et des balises propres à Rennes1 que l'on souhaiterait éditer dans notre nouveau formulaire donc :

```
<?xml version="1.0" encoding="utf-8"?>
<lom xmlns="http://ltsc.ieee.org/xsd/LOM"
  xmlns:lomurl="http://www.univ-rennes1.fr/xsd/LOMUR1">
  <general>
    <identifiant>
      <catalog>URI</catalog>
      <entry>http://www.univ-rennes1.fr/12345</entry>
    </identifiant>
    <title>
      <string language="fr">Médecine Informatique et Écologie</string>
    </title>
    <!-- etc. -->
  </general>
  <!-- etc. -->
  <lomurl:lomurl>
    <lomurl:targetPublic>UMVF</lomurl:targetPublic>
    <lomurl:targetPublic>UNIT</lomurl:targetPublic>
    <lomurl:targetPublic>UVED</lomurl:targetPublic>
  </lomurl:lomurl>
</lom>
```

Ce fichier est valide *lomLoose*, et on n'a pas forcément besoin de créer un schéma spécifique "LOM-Rennes1" pour notre besoin : celui-ci reste interne, notre espace de noms n'a pas vocation à être utilisé en dehors de nos applications. Par contre, on distingue bien grâce aux espaces de noms le LOM de nos balises personnelles.

Grâce aux espaces de noms, on peut noter que les différents modules peuvent travailler avec cet XML d'exemple ci-dessus : l'éditeur ori-oai-md-editor peut éditer cet XML directement (aussi bien la version complète, que la version auteur). Cependant, il ne nous permet pas d'éditer la partie spécifique Rennes1 qu'il ne connaît pas.

C'est pourquoi on va réaliser un nouveau formulaire basé sur le formulaire LOM complet d'ori-oai-md-editor.

2. Formulaire LOM Rennes1

i. copier/coller

Confère la documentation du ori-oai-md-editor à <http://sourcesup.cru.fr/ori-workflow/> (qui partage le projet sourcesup avec ori-oai-workflow), le répertoire spécifique à ori-oai-md-editor dans l'application Orbeon Forms est : **WEB-INF/resources/apps/ori-md-editor**

On va dupliquer le formulaire lom-full, je vous liste les commandes :

```
[ori-md-editor]$ ls
close.xhtml config.xml i18n media page-flow.xml prototypes reload-vocab schemas welcome xforms xml-
blank xpl
[ori-md-editor]$ ls xforms/
common lom-author-light lom-full oaidc-full
[ori-md-editor]$ cp -rf xforms/lom-full xforms/lom-full-extend
[ori-md-editor]$ cp xml-blank/lom-full-blank.xml xml-blank/lom-full-extend-blank.
```

Les formulaires en tant que tels correspondent aux sous-répertoires du répertoire xforms (excepté le répertoire common).



Les fichiers d'initialisation des formulaires en mode "standalone" (non connecté au workflow) sont les fichiers du répertoire **xml-blank** dont les noms doivent correspondre avec les noms des répertoires des formulaires.

Dès maintenant et sans redémarrer votre serveur ori-oai-md-editor, vous devez pouvoir utiliser ce nouveau formulaire directement :

http://NOM_SERVEUR:PORT/ori-oai-md-editor/ori-md-editor/lom-full-extend/blank

ii. *svn update*

Pour ceux qui sont connectés via subversion (cf. **l'Annexe B** : « Exploitation d'applications avec subversion - installation et mises à jour », on vous encourage vivement à procéder de la sorte) vous pouvez garder le répertoire **.svn** copié au passage dans **xforms/lom-full-extend**, cela vous permet de continuer à vous synchroniser au répertoire **lom-full** du repository svn officiel de ori-oai-md-editor. Par contre pour ce faire vous devrez aller dans le répertoire **lom-full-extend** et faire un `svn update` indépendamment du svn update de l'ensemble :

```
cd xforms/lom-full-extend/  
svn info
```

Path: .

URL: <http://subversion.cru.fr/ori-workflow/ori-oai-md-editor/trunk/WEB-INF/resources/apps/ori-md-editor/xforms/lom-full>

Repository Root: <http://subversion.cru.fr/ori-workflow>

Revision: 471

Node Kind: directory

Schedule: normal

Last Changed Author: vbonamy

Last Changed Rev: 471

Last Changed Date: 2007-12-16 14:05:14 +0100 (Sun, 16 Dec 2007)

a

```
svn update
```

```
U case-educational.xml
```

```
U main-form.xhtml
```

```
Updated to revision 471.
```

3. Prototype

Dans XForms et Orbeon Forms, l'ajout de noeud via un bouton "**Ajouter un targetPublic**" correspond en fait à un copier coller d'un noeud existant "ailleurs". Le "ailleurs" peut correspondre à une autre instance XML que celle que l'on édite. On appelle cela dans ori-oai-md-editor un prototype.

Ici, on va ajouter un prototype pour le LOM Rennes1 :

```
cd prototypes  
cp lom-prototype.xml lom-rennes1-prototype.xml
```

On y ajoute les tags vides qui vont bien, **en n'oubliant pas de déclarer l'espace de noms** :

```
....  
xmlns:lomurl="http://www.univ-rennes1.fr/xsd/LOMUR1"  
...  
  <lomurl:lomurl>  
    <lomurl:targetPublic/>  
  </lomurl:lomurl>  
....
```

4. Modification du formulaire

On se place dans le formulaire nouvellement créé pour maintenant le modifier :

```
cd xforms/lom-full-extend/  
ls
```



```
case-annotation.xml case-educational.xml case-lifecycle.xml case-relation.xml case-technical.xml
dialog-taxonomy-search.xml entity.xml
case-classification.xml case-general.xml case-metametadata.xml case-rights.xml dialog-dewey-search.xml
dialog-vcard-search.xml main-form.xhtml
```

Ces différents fichiers sont des XForms. Les modifications que l'on va apporter correspondent à reprendre du code existant dans ces formulaires pour l'adapter à nos nouvelles balises, modifier les vocabulaires etc.

Le fichier d'entrée du formulaire est en fait **main-form.xhtml**. Dans ce fichier main-form.xhtml :

- on change le prototype utilisé (on recherche simplement **lom-prototype.xml** que l'on va changer par **lom-rennes1-prototype.xml**)

```
<!-- LOM prototype -->
<xforms:instance id="ori-prototype" src="oxf:/apps/ori-md-editor/prototypes/lom-rennes1-
prototype.xml" xxforms:readonly="true" xxforms:shared="application"/>
```

- on ajoute un onglet rennes1 en dessous de celui de classification (en copiant collant celui de classification):

```
<widget:tab id="rennes1">
  <widget:label>Rennes1</widget:label>
  <xi:include href="case-rennes1.xml"/>
</widget:tab>
```

On va ensuite éditer un **case-rennes1.xml** en prenant exemple sur les autres **case-*.xml** en n'oubliant pas de déclarer l'espace de noms rennes1 et de l'utiliser !

Voici ce que peut donner ici un tel **case-rennes1.xml** :

```
<xforms:group ref="." xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xforms="http://www.w3.org/2002/xforms"
  xmlns:ev="http://www.w3.org/2001/xml-events"
  xmlns:xxforms="http://orbeon.org/oxf/xml/xforms"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:f="http://orbeon.org/oxf/xml/formatting"
  xmlns:lom="http://ltsc.ieee.org/xsd/LOM"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:widget="http://orbeon.org/oxf/xml/widget"
  xmlns:vocab="http://www.ori-oai.org/xsd/orioaivocab"
  xmlns:lomur1="http://www.univ-rennes1.fr/xsd/LOMUR1">

  <!-- Rennes1 -->
  <fieldset>
    <legend>
      Rennes1 - Metadonnees specifiques
    </legend>

    <widget:ori-block element="lomur1:lomur1" minOccurs="0" maxOccurs="1" preceding-
elements="lom:general | lom:lifecycle | lom:metaMetadata | lom:technical | lom:educational |
lom:rights | lom:relation | lom:annotation | lom:classification">

      <!-- targetPublic -->
      <fieldset>

<legend>

  targetPublic
  ...
</legend>

<widget:ori-block element="lomur1:targetPublic" minOccurs="0" maxOccurs="unbounded" parent-
element="lomur1:lomur1">
  <xforms:select1 ref=".">
    <xforms:item>
```



```
<xforms:label>[Select]</xforms:label>
<xforms:value></xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>UMVF</xforms:label>
  <xforms:value>UMVF</xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>UNIT</xforms:label>
  <xforms:value>UNIT</xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>UVED</xforms:label>
  <xforms:value>UVED</xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>UNJF</xforms:label>
  <xforms:value>UNJF</xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>AUNEGE</xforms:label>
  <xforms:value>AUNEGE</xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>UOH</xforms:label>
  <xforms:value>UOH</xforms:value>
</xforms:item>
<xforms:item>
  <xforms:label>UNISCIEL</xforms:label>
  <xforms:value>UNISCIEL</xforms:value>
</xforms:item>
</xforms:select1>
</widget:ori-block>

</fieldset>

</widget:ori-block>

</fieldset>

</xforms:group>
```

5. Termes i18n ...

Même si on a mis des termes en dur dans le XForms ci-dessus, la facilité des widgets qu'on donne dans ori-oai-md-editor nécessite le positionnement de paramètres i18n pour ajouter/supprimer les tags que l'on à rajouter. On ajoutera donc dans **i18n/fr_FR.xml** la partie suivante (pensez à faire de même pour **i18n/en_EN.xml** :

```
<!-- LOM Rennes1 - add ... -->
<add-lomurl>Ajouter un bloc Lomurl</add-lomurl>
<del-lomurl>Supprimer le bloc Lomurl</del-lomurl>
<add-targetPublic>Ajouter un bloc TargetPublic</add-targetPublic>
<del-targetPublic>Supprimer le bloc TargetPublic</del-targetPublic>
```

Redémarrez votre serveur tomcat.

Vous devriez maintenant avoir un nouvel onglet dans votre éditeur de métadonnées !

Editeur ORI-MD

Français Nouveau Parcourir... Charger Sauvegarder

Il reste des erreurs dans la fiche Détails des erreurs !

Général Cycle de vie Méta-métadonnées Technique Pédagogique Droits Relation Commentaire Classification Rennes1

Rennes1 - Metadonnees spécifiques

targetPublic ...	
UMVF	!
[Select]	!
Ajou [Select]	
UMVF	
UNIT	
UVED	
UNJF	
AUNEGE	
UOH	
UNISCIEL	

Powered by OPS