

Space Details

| | |
|-----------------------------------|--|
| Key: | ORIOAImdeditor |
| Name: | ORI-OAI-md-editor |
| Description: | |
| Creator (Creation Date): | ycolmant@univ-valenciennes.fr (Jul 04, 2008) |
| Last Modifier (Mod. Date): | ycolmant@univ-valenciennes.fr (Aug 29, 2008) |

Available Pages

- Version 1.1 
 - Spécifications
 - Choix techniques
 - Implémentation
 - Spécifications du module
 - Installation
 - Changements version 1.1.*
 - Installation du module
 - Test
 - Utilisation
 - Aspects Pratiques
 - Encodage
 - ori-oai-md-editor et le module ori-oai-vocabulary
 - Performances
 - Personnalisation de formulaires
 - Ajout d'une métadonnée dans le formulaire

ORI-OAI-md-editor : Interface de saisie des métadonnées



Module optionnel

[Voir l'architecture du système](#)

Dans quels cas l'utiliser

Pour la saisie des fiches de métadonnées

Composants obligatoires

- **ORI-OAI-vocabulary** pour fournir les vocabulaires lors de la saisie des métadonnées
--> note: ce module peut aussi être utilisé seul sans la connexion au module ORI-OAI-vocabulary.
Ce cas peut se présenter dans le cadre de tests, de démonstration, ou pour utiliser un éditeur de métadonnées en dehors de ORI-OAI

Description

Ce composant est utilisé pour la saisie des métadonnées dans ORI-OAI. Les formulaires de saisie des métadonnées sont entièrement configurables. La technologie XForms associée au moteur de transformation Orbeon offre des formulaires de saisie dynamiques en fonction de fichiers XML XForms.

Le support des différents formats de métadonnées repose sur la possibilité d'enrichir le module ORI-OAI-md-editor de nouvelles configurations XForms.

Ce module permet une saisie des métadonnées très riche grâce notamment à de l'aide à la saisie, de l'auto complétion, de la recherche de personnes dans l'annuaire, etc.

Couplé au module ORI-OAI-workflow, il permet de proposer un formulaire de saisie des métadonnées à n'importe quelle étape du workflow. Utilisé seul, il permet l'édition de fiches de métadonnées en dehors du système. Il répond alors au besoin d'un éditeur simple et puissant de métadonnées au format XML.

[Voir la documentation technique](#)

Spécifications

This page last changed on Jan 05, 2009 by ycolmant@univ-valenciennes.fr.

- [Choix techniques](#)
- [Implémentation](#)
- [Spécifications du module](#)

Choix techniques

Formulaire d'édition des MD : XForms + Orbeon Forms

Les formulaires d'édition proposés à l'utilisateur dans ORI-OAI-Workflow correspondent à une partie très importante de ORI-OAI-Workflow dans le sens où c'est la partie la plus visible de ORI-OAI-Workflow pour l'utilisateur final. Ces formulaires peuvent se décliner selon les différents schémas (DC, LOM, LOM-FR, ...), selon les différents rôles des différents utilisateurs. Très souples, ils acceptent un certain nombre de paramètres qui diffèrent selon les usages. On pense notamment aux taxonomies/vocabulaires utilisés lors de la saisie via des widgets à choix multiple. Enfin ils sont ergonomiques et permettent une saisie efficace des fiches XML de métadonnées : auto-complétion, ajout/suppression instantané de champs, bref les formulaires sont des formulaires dynamiques (Ajax).

Le choix a été d'opter pour un standard W3C [XForms](#) qui permet de créer des formulaires ergonomiques, dynamiques, ... cela via un langage XML. Vu que XForms n'est pas supporté nativement par les navigateurs et que les implémentations de plugins sur les navigateurs ne sont pas encore tout à fait au point, et enfin pour des questions de souplesse, la solution proposée par Orbeon, [Orbeon Forms](#) a été retenue. Celle-ci permet entre autre de générer à partir de formulaires XForms une interface Html/Ajax interprétable par les principaux navigateurs clients du marché.

Orbeon est à l'image de Cocoon un framework complet de développement qui tire parti, de manière extrêmement ingénieuse, de la technique de transformation XML pour concevoir des applications complexes de manière efficace. On a choisi de retenir Orbeon Forms pour la saisie des métadonnées et donc pour sa qualité première et spécifique de technologie de formulaires.

Conclusion

Le module ORI-OAI-MD-Editor correspond à un XForms interprété par Orbeon Forms pour générer une interface Ajax à l'utilisateur. Il ne contient pas de logique applicative. Il reçoit un XML et en renvoie un. Pour ce faire un protocole de communication spécifique (s'appuyant sur des Web Services SOAP) est cependant mis au point pour permettre à Orbeon et ori-oai-workflow (spring) d'interagir ensemble.

Implémentation

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

Implémentation

ori-oai-md-editor

On distingue 2 parties dans l'implémentation de ori-oai-md-editor : * les XForms, c'est à dire les Editeurs de Métadonnées effectifs,

- l'interaction en SOAP avec les autres modules : ori-oai-workflow-spring et ori-oai-vocabulary.

XForms

Un certain nombre d'XForms sont proposés en standard dans ORI-OAI-Workflow.

Pour le LOM, plusieurs XForms sont donnés (pour l'instant 2), ils permettent à des niveaux différents d'informer un fichier de métadonnées en LOM. Ces niveaux sont fonctions du degré de précision de la saisie.

Exemple : les personnes indexeurs auront accès à toutes les métadonnées du LOM (dont la dewey), tandis que le dépositaire de la ressource n'en informera que certaines.

Interaction en SOAP avec ori-oai-workflow-spring / ori-oai-vocabulary

ori-oai-md-editor communique avec ori-oai-workflow-spring en SOAP pour récupérer et sauver le XML présentant l'instance à éditer. ori-oai-md-editor communique avec ori-oai-vocabulary pour récupérer les XMLs correspondant aux différents thésaurus/vocabulaires utilisés dans le formulaire XForms.

Mise à jour de la distribution Orbeon Forms **[s'adresse uniquement aux développeurs/packageurs]**

Pour mettre à jour la distribution de orbeon forms : * récupérez directement une distribution du WAR du type ops.war

- écrasez dans la distribution de ori-oai-md-editor : *** le fichier WEB-INF/web.xml (modifiez le toutefois en supprimant les services/servlets inusités dans ori-oai-md-editor.
 - - les librairies de WEB-INF/lib/
 - le répertoire WEB-INF/resources/apps/context en modifiant toute fois le fichier servlet-initialized.xpl (en prenant exemple sur le diff entre l'ancien servlet-initialized.xpl_init et servlet-initialized.xpl : l'idée est d'appeler init-vocab-scheduler.xpl lorsque la servlet xforms-server est chargée)
 - le répertoire WEB-INF/resources/config en modifiant toute fois epilogue-servlet.xpl (en prenant exemple sur le diff entre l'ancien epilogue-servlet.xpl_init et epilogue-servlet.xpl) ainsi que xforms-widgets.xsl, properties.xml. Recopiez également le fichier theme-ori.xsl (utilisé dans epilogue-servlet.xpl).

Suivant bien sûr les modifications apportées par la nouvelle version Orbeon Forms, ori-oai-md-editor devrait marcher directement tel quel.

Spécifications du module

This page last changed on Jan 05, 2009 by ycolmant@univ-valenciennes.fr.

Spécifications

ORI-OAI-MD-Editor (XForms+Orbeon)

Formulaires par défaut.

Un certain nombre de formulaires sont donnés par défaut : édition de fiche DC, LOM (complet et simple pour l'auteur). Libre aux utilisateurs d'en concevoir et d'en ajouter de nouveaux (et des les partager

avec la communauté 😊). L'application étant prévu pour cela des formulaires pour d'autres formats de métadonnées sont à l'étude. L'idée est effectivement de partager avec la communauté ces possibles contributions et les intégrer dans les formulaires proposés par ORI par défaut.

Ces formulaires sont en XForms et interprétés par Orbeon. Certains peuvent appeler des fichiers XML distants qui sont des taxonomies/vocabulaires qui peuvent être partagés par une communauté donnée (via le module Ori-Oai-Vocabulary). Les URL de ces vocabulaires distants sont à modifier selon le besoin.

Interactions ORI-OAI-MD-Editor / ORI-OAI-Workflow-Spring

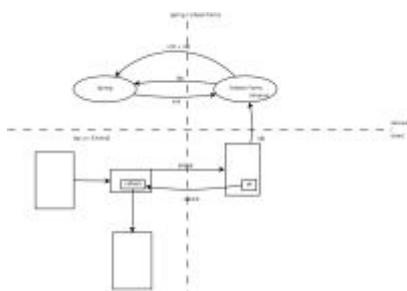
Lors d'un besoin d'édition de XML, ORI-OAI-MD-Editor est appelé depuis l'interface de ORI-OAI-Workflow-Spring avec comme paramètre un identifiant généré par Spring (idp), fonction de l'identifiant de la fiche à éditer.

Une popup (ou quelque chose de similaire) s'ouvre dans laquelle est affichée le formulaire de saisie produit par Orbeon Forms.

Pour afficher ce formulaire, Orbeon a demandé via Webservice, le XML correspondant à l'identifiant

Le bouton "sauver" de la popup orbeon provoque : * l'envoi du XML et de l'identifiant idp à ORI-OAI-Workflow-Spring via Webservice pour sauvegarde du XML,

- l'envoi d'une commande javascript (~ rafraichissement, submit d'un bouton, ...) sur la fenêtre principale Spring,
- la fermeture de la popup



Installation

This page last changed on Jan 05, 2009 by ycolmant@univ-valenciennes.fr.

- [Changements version 1.1.*](#)
- [Installation du module](#)
- [Test](#)

Changements version 1.1.*

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

Changements version 1.1.*

Les valeurs des propriétés principales par défaut ont été modifiées : elles apparaissent entre crochet et en majuscule.

Elles peuvent soit être éditées manuellement, soit être modifiées via la *configuration centrale du quick-install* voir plus bas.

ori-oai-md-editor

- L'installation de ce module peut se faire comme dans la version précédente en le disposant directement en tant qu'application Tomcat.
Un build.xml est également maintenant disponible, il permet de déployer ce module de la même façon que tous les autres modules, via notamment le module quick-install.
- ori-oai-md-editor peut s'utiliser interconnecté avec un module ori-oai-vocabulary (ou non) et/ou avec un module de ori-oai-workflow (ou non).
=> il peut donc être utilisé sans aucune interconnection avec d'autres modules ori-oai (la version précédente imposait une connection avec un module ori-oai-vocabulary distant)..
- Les vocabulaires utilisés sont des vocabulaires VDEX.
- Un formulaire LOMFR est à disposition pour tests, tout commentaire est le bienvenu.
- La version d'Orbeon Forms utilisée a été mise à jour : est utilisé [Orbeon Forms 3.6](#) (sorti en Décembre 2007).
ori-oai-md-editor bénéficie ainsi de facto de toutes les améliorations amenées avec cette version notamment au niveau des performances, consommation RAM,
- ... enfin un certain nombre d'améliorations ont été apportées sur des points de détails au niveau des formulaires ...

Installation du module

This page last changed on Jan 05, 2009 by ycolmant@univ-valenciennes.fr.

Installation

Vous noterez que le répertoire "source" est déposé tel quel en tant qu'application J2EE (dans webapps par exemple sur un Tomcat).

Pour faciliter les choses, on vous conseille (cf le [manuel de formation](#)) d'utiliser subversion pour télécharger le tag ou version stable de l'éditeur.

Une fois lancée, pour vérifier que l'application tourne comme il faut, pointez votre navigateur sur l'url de votre application tomcat

ORIOAIworkflow:HOST_MD_EDITOR:ORIOAIworkflow:PORT_MD_EDITOR/

ORIOAIworkflow:CONTEXT_MD_EDITOR

c'est à dire par exemple

<http://localhost:8080/ori-oai-md-editor>

Vous devriez voir apparaître les différents formulaires disponibles dans ori-oai-md-editor, c'est à dire pour ori-oai-md-editor- : * oaidc-full : correspond au formulaire LOM dans sa version intégrale

- lom-full : correspond au formulaire LOM dans sa version intégrale
- lomfr-full : correspond au formulaire LOMFR dans sa version intégrale (ORIOAIworkflow:n'est pas utilisé par défaut dans la version courante de ori-oai-workflow, le sera dans une prochaine version après vérification qu'il répond bien à la norme LOMFR)
- lom-author-light : correspond au formulaire LOM dans sa version simplifiée (à destination par exemple d'un auteur lors de la première étape dans un workflow donné).

Les fichiers propres à ORI se trouvent ici : ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor/

La configuration de base est de donner les urls des modules de workflow, de vocabulary ainsi que l'url du module md-editor lui-même. Cela se fait ici : ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor/config.xml

Si vous souhaitez utiliser l'éditeur seul, c'est à dire sans qu'il interagisse avec un module ori-oai-vocabulary et ori-oai-workflow, vous pouvez simplement supprimer l'url référençant le module ori-oai-vocabulary : les vocabulaires embarqués (localement) avec l'éditeur seront alors utilisés. Le module ori-oai-workflow n'est utilisé que si on accède à l'éditeur depuis ce dernier.

Test

This page last changed on Jan 05, 2009 by ycolmant@univ-valenciennes.fr.

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

```
http://[HOST_INSTALL]:8186/ori-oai-md-editor
```

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



Eichier Édition Affichage Historique Marque-pages Outils Aide

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 :

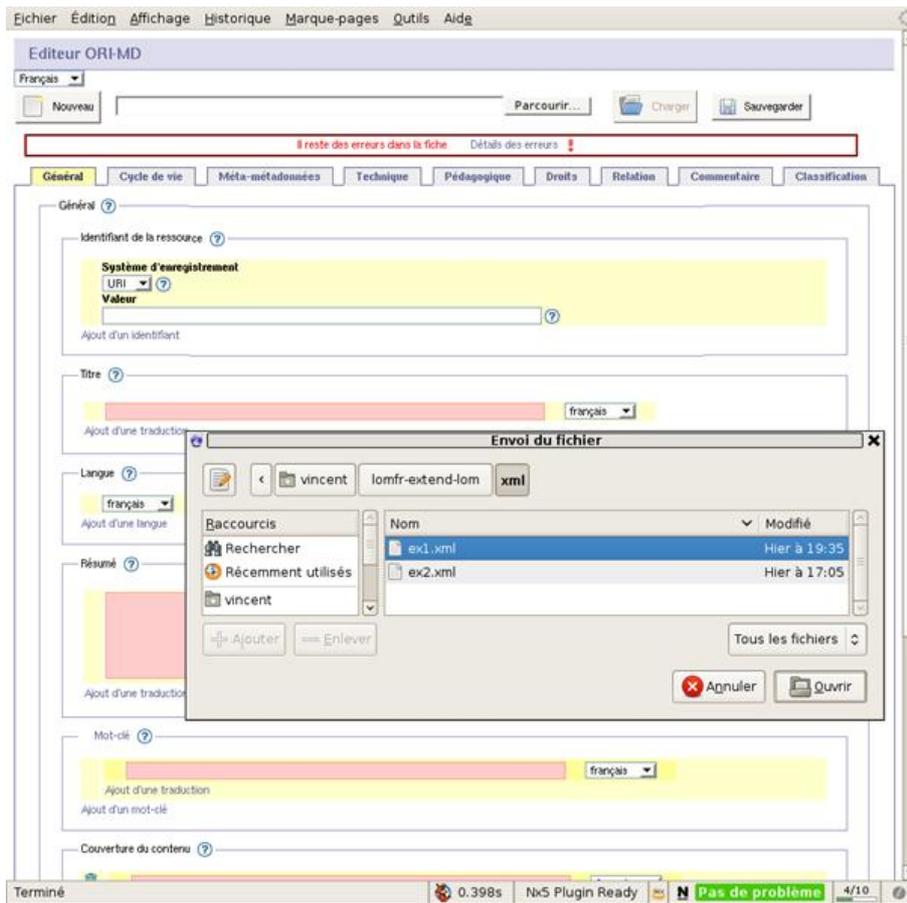
workflow:
<http://ori-oai-workflow.univ-rennes1.fr/workflow>

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

Powered by ORS 

Terminé 0.255s Nx5 Plugin Ready N Pas de probleme 4/10

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 :



Utilisation

This page last changed on Jan 05, 2009 by ycolmant@univ-valenciennes.fr.

A faire ...

Aspects Pratiques

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

- [Encodage](#)
- [ori-oai-md-editor et le module ori-oai-vocabulary](#)
- [Performances](#)
- [Personnalisation de formulaires](#)
- [Ajout d'une métadonnée dans le formulaire](#)

Encodage

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

Encodage

De par leurs fonctionnalités et leurs objectifs d'interopérabilité, les modules ORI-OAI sont voués à fonctionner dans un encodage UTF-8. Il est préférable (pour éviter au mieux les problèmes d'encodage dans les données, le rendu etc.) de positionner l'encodage à UTF-8 pour tous les composants acteurs dans ORI-OAI (les bases de données, les serveurs d'application, etc). Cela peut se faire de différentes manières (depuis les variables d'environnement du système par exemple LANG). Le positionner en tant qu'option Tomcat est une bonne option également :

```
CATALINA_OPTS=-Dfile.encoding=UTF-8
```

ori-oai-md-editor et le module ori-oai-vocabulary

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

ori-oai-md-editor et le module ori-oai-vocabulary

ori-oai-md-editor est client de ori-oai-vocabulary : il récupère par Web Services les vocabulaires dont il a besoin. Ainsi par exemple, vous pouvez récupérer le vocabulaire des langues du module ori-oai-vocabulary d'[UNIT](#) via le module ori-oai-md-editor d'UNIT : <http://www.unit.eu/ori-oai-md-editor/ori-md-editor/vocab/languages>. Vous pouvez donc faire de même entre votre module ori-oai-md-editor et votre module ori-oai-vocabulary. Cela vous permet de tester la bonne configuration de ori-oai-md-editor->ori-oai-vocabulary et de tester vos vocabulaires.

Les formulaires Orbeon Forms de Ori-Oai-Md-Editor partagent les mêmes vocabulaires qui sont mis en cache au niveau d'Orbeon Forms même. Par défaut, ces caches sont initialisés lors du lancement de l'application (donc du Tomcat) puis ils sont rechargés automatiquement régulièrement : le paramètre vocab-cache-time dans le fichier config.xml stipule l'intervalle de rechargement (en millisecondes). L'appel à des urls particulières permettent également de forcer manuellement à charger/décharger/recharger les caches. <http://localhost:8280/ori-oai-md-editor/ori-md-editor/reload-vocab> permet de recharger les vocabulaires.

Performances

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

Performances

Cf [Orbeon Performance and Tuning](#), il est possible d'améliorer les performance de Orbeon Forms de différentes manières. La version utilisée d'Orbeon Forms embarque désormais une base eXist utilisé pour stocker les instances XML en cours d'usage. Aussi ORI-OAI-MD-Editor a normalement moins besoin de RAM qu'avant. Cependant il reste conseillé d'allouer un large volume de RAM au Tomcat faisant tourner Orbeon Forms. ainsi par exemple vous pouvez positionner votre variable d'environnement comme cela :

```
export JAVA_OPTS='-Xmx512m -Xms512m'
```

(voir plus encore que 512MO si vous avez la possibilité d'allouer plus de RAM)
L'utilisation du JDK6 est conseillé également afin d'améliorer les performances.

Il est fortement conseillé d'utiliser les possibilités de compression des pages css, javascript et html (notamment la page html représentant le formulaire qui peut être relativement importante et qui ne doit d'ailleurs pas être mise en cache au niveau du navigateur comme du serveur web). Cela peut se faire via Tomcat directement ou plus aisément encore via Apache par exemple si vous l'utilisez comme serveur web frontal à Tomcat.

Voici par exemple via Apache comment procéder : il faudra s'assurer que les modes deflate et header sont chargés :

```
<Location /ori-oai-md-editor>

<IfModule mod_deflate.c>

SetOutputFilter DEFLATE

# Don't compress images
SetEnvIfNoCase Request_URI \
\.(?:gif|jpe?g|png)$ no-gzip dont-vary

# Make sure proxies don't deliver the wrong content
Header append Vary User-Agent env=!dont-vary

</IfModule>

</Location>
```

Enfin, il est aussi intéressant de cacher les images, css et javascript au niveau du serveur Apache (seul la page html représentant le formulaire ne doit pas être mis en cache au niveau du navigateur comme du serveur web, c'est important). Via le mode cache (disk_cache) d'Apache, voici comme cela peut se mettre en place simplement :

```
<IfModule mod_disk_cache.c>
CacheEnable disk /
</IfModule>
```

Personnalisation de formulaires

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

Création / personnalisation de formulaires

ori-oai-md-editor a été pensé pour permettre l'ajout de nouveaux formulaires.

Pour réaliser un nouveau formulaire (pour un nouveau schéma ou un schéma déjà supporté), le plus simple est de copier/coller (en renommant) depuis le répertoire `ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor/xforms` un dossier représentant un formulaire existant : `oaidc-full` par exemple en. Si vous renommez ce répertoire fraîchement copié `oaidc-full-custom` par exemple, vous devriez le voir ensuite aussitôt apparaître dans la liste des formulaires disponibles sur <http://localhost:8280/ori-oai-md-editor>.

Ensuite les modifications effectives d'un formulaire se réalisent en éditant et modifiant les fichiers XForms du répertoire du formulaire : `ori-oai-md-editor/WEB-INF/resources/apps/ori-md-editor/xforms/oaidc-full-custom` par exemple. Le point d'entrée d'un formulaire dans Ori-Oai-Md-Editor étant obligatoirement le fichier `main-form.xhtml` (qui peut suivant les cas, être le seul fichier du répertoire).

Ajout d'une métadonnée dans le formulaire

This page last changed on Oct 14, 2008 by vibonamy@univ-rennes1.fr.

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.

à 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.

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 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.

Formulaire LOM Rennes1

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
```

svn update

Pour ceux qui sont connectés via subversion (cf. **I'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: vbony

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.
```

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 nœud 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>
...
```

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:lomurl="http://www.univ-rennes1.fr/xsd/LOMURL">

    <!-- Rennes1 -->
    <fieldset>
        <legend>
            Rennes1 - Metadonnees specifiques
        </legend>

        <widget:ori-block element="lomurl:lomurl" 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="lomurl:targetPublic" minOccurs="0" maxOccurs="unbounded" parent-
element="lomurl:lomurl">
                    <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>

```

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 - Metadonnées spécifiques

targetPublic ...

UMVF

[Select]

Ajou [Select] etPublic

UMVF

UNIT

UVEO

UNJP

ALUNEGE

UOH

UNSCIEL

Powered by OPS