

Ateliers ADNX

Le protocole OAI

Ce document contient l'ensemble de la documentation liée à l'atelier sur le protocole OAI proposé l'Association pour la documentation numérique en XML (ADNX, <http://adnx.org>). Cette documentation peut également être consultée sur Internet à l'adresse <http://adnx.org/documentation/oai/>.

Ce document PDF a été généré le 2006-02-02.

Table des matières

Atelier ADNX / OAI.....	2
Présentation de l'atelier.....	3
Le protocole OAI et les normes associées.....	4
Les concepts fondamentaux.....	5
Les objets documentaires.....	7
L'architecture technologique.....	8
Les formats de données.....	10
Les instructions et paramètres OAI.....	13
Pour en savoir plus.....	19
Panorama des utilisations du protocole OAI dans le monde et en France.....	20
Méthodologie de mise en œuvre de l'OAI.....	21
Mettre en place une archive institutionnelle avec DSpace.....	23
Transformer une base de données MySQL en entrepôt OAI avec phpoi2.....	24
Configurer un entrepôt OAI depuis une application SDX.....	26
Moissonner des sites distants avec SDX.....	28

Atelier ADN X / OAI

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

L'association pour la documentation numérique en XML (ADNX) propose à ses membres et à toute la communauté des ateliers. Un de ces ateliers concerne le protocole OAI-PMH, ou *Open Archives Initiative Protocol for Metadata Harvesting*. Ce document constitue la documentation associée à cet atelier.

Présentation de la journée

Quelques informations générales sur l'atelier.

Présentation du protocole OAI et des normes associées

Un parcours de la spécification avec une approche pédagogique et des exemples.

Panorama des utilisations du protocole OAI dans le monde et en France

Qui utilise l'OAI ? Pourquoi ? Comment ?

Méthodologie de mise en oeuvre du protocole OAI

Par où commencer ? Qui doit intervenir ?

Mettre en place une archive institutionnelle avec Dspace

Installation, configuration, publication de documents, entrepôt OAI.

Transformer une base de données en entrepôt OAI avec OAI PHP

Un exemple issu d'une base de données sur des œuvres de musées.

Utilisation de SDX pour mettre en oeuvre un entrepôt OAI

Configuration d'un entrepôt SDX sur l'application `sdxtest` et sur une application Dublin Core.

Moissonner des sites distants avec SDX

Configuration d'un moissonneur OAI dans une application SDX pour aller chercher des données externes provenant de différents systèmes.

L'ensemble de la documentation est également accessible [en format PDF](#).

Présentation de l'atelier

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

L'atelier OAI fait partie de la série des [ateliers ADNX](#). Il se déroule sur une journée et est organisé par Martin Sévigny, avec différents autres contributeurs que nous remercions.

L'atelier ne comporte pas de travaux pratiques, mais donne une large place aux démonstrations pratiques. De plus, les échanges avec les participants sont fortement encouragés.

Il touche à la fois aux aspects conceptuels et aux aspects techniques de l'OAI. C'est pourquoi il s'adresse à des personnes ayant des profils variés, du chef de projet à l'informaticien en passant bien sûr par le documentaliste.

Pour être en mesure de suivre et, espérons-le, apprécier cet atelier, il suffit de remplir ces conditions :

- avoir une certaine expérience du Web pour comprendre la notion de mise en réseau
- avoir déjà vu du code XML ou ne pas être effrayé par une balise de type `<dc:subject>OAI</dc:subject>`
- avoir déjà utilisé une base documentaire tel un catalogue de bibliothèque informatisé

L'essentiel de la matinée est consacré d'une part à comprendre ce qu'est le protocole OAI et ensuite à se faire une idée de son utilisation dans le monde.

Ensuite, l'après-midi sera consacré à des démonstrations de mise en œuvre simple de l'OAI. Ces démonstrations permettront de se faire une meilleure idée des compétences et outils nécessaires pour réaliser son projet OAI.

Le programme détaillé de cet atelier est le suivant :

Heure	Description
09h00 – 09h30	Présentation de l'atelier et des enjeux (Martin Sévigny, AJLSM)
09h30 – 10h45	Présentation du protocole OAI et des normes associées (Martin Sévigny)
10h45 – 11h00	Pause
11h00 – 12h00	Panorama des utilisations du protocole OAI dans le monde et en France (Jean-Paul Ducasse, Université Lyon 2)
12h00 – 12h30	Méthodologie de mise en œuvre du protocole OAI (Martin Sévigny)
12h30 – 14h00	Déjeuner libre
14h00 – 14h20	Mettre en place une archive institutionnelle avec Dspace (Martin Sévigny)
14h20 – 15h00	Transformer une base de données en entrepôt OAI avec phpoi2 (Martin Sévigny)
15h00 – 15h30	Utilisation de SDX pour mettre en œuvre un entrepôt OAI (Martin Sévigny)
15h30 – 15h45	Pause
15h45 – 16h45	Moissonner des sites distants avec SDX (Martin Sévigny)
16h45 – 17h00	Conclusion

Le protocole OAI et les normes associées

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Le protocole OAI est relativement simple à comprendre et, de plus, à tester. Ces deux aspects sont probablement à l'origine de son succès. Derrière cette simplicité, toutefois, il convient de faire attention à certains concepts pour en mesurer toute la portée ou, bien souvent, les limites.

Nous tentons d'aborder le protocole OAI du général au particulier, avec dans l'ordre :

Les concepts fondamentaux

Qu'est-ce qu'un entrepôt OAI ? Qu'est-ce qu'un moissonneur OAI ?

Les objets documentaires et l'OAI

Qu'est-ce qu'une ressource ? Qu'est-ce qu'un item OAI ? Qu'est-ce qu'un enregistrement OAI ?

L'architecture technologique et l'OAI

Du XML dans des flux HTTP. Le modèle requête / réponse.

Les formats de données

Le Dublin Core. Les autres formats. Les identifiants. Les dates.

Les instructions et paramètres OAI

Les instructions de moisson. La moisson sélective par date et par ensemble. Les enregistrements supprimés.

Pour en savoir plus

Le protocole. Les guides d'implémentations. Les outils de test.

Les normes associées telles que XML, Dublin Core, HTTP, seront abordées au cours de cette présentation, là où elles sont nécessaires.

Les concepts fondamentaux

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Le protocole OAI est une norme qui aide à mettre en place des **infrastructures d'interopérabilité** pour des informations documentaires. Elle est basée sur le principe de **collecte de métadonnées**, opération nommée *moisson* (*harvesting*) dans les spécifications du protocole.

On retrouve deux types d'acteurs dans une infrastructure OAI :

- Les **fournisseurs de données**, qui exposent des métadonnées de manière **passive** en implémentant le protocole OAI ; on les nomme **entrepôts OAI**.
- Les **fournisseurs de services**, qui recueillent de manière **active** des métadonnées à l'aide du protocole OAI afin de fournir des services à valeur ajoutée sur ces données.

Les deux premiers concepts fondamentaux sont donc ceux d'**entrepôt OAI** et de **moissonneur OAI**, que nous allons réutiliser tout au long de cette documentation en lieu et place de fournisseur de données et fournisseur de services, qui sont moins spécifiques à l'OAI.

Le **rôle du protocole OAI est de normaliser les échanges** (au sens informatique du terme) entre un moissonneur OAI et un entrepôt OAI. Les spécifications du protocole OAI n'indiquent en aucun cas comment implémenter un entrepôt OAI et un moissonneur OAI, ni comment organiser ou gérer les données exposées ou recueillies.

Les définitions ci-dessus introduisent un autre concept, celui de **métadonnées**, un mot qui peut signifier plusieurs choses. Nous allons clarifier un peu plus loin les différents **objets documentaires** introduits par le protocole et cela permettra de mieux comprendre ce que signifie métadonnées ici. Pour l'instant, nous allons nous contenter de dire qu'il s'agit de données descriptives à propos d'un objet, données qui peuvent être très riches jusqu'au point de représenter l'objet au complet s'il le faut.

En terminant la présentation de ces concepts fondamentaux, nous proposons un exemple *humanisé* d'échange informatique pour bien indiquer ce qu'est un protocole en informatique. Examinons le scénario suivant :

Deux individus parlant le français se rencontrent dans un café.
Ils décident d'échanger sur différents sujets, et ils débutent leur échange ainsi :

Individu A : Bonjour, qui êtes-vous ?

Individu B répond : Je m'appelle Michel Tremblay et je suis écrivain

Individu A poursuit : Quels livres avez-vous écrit dans la suite des Chroniques du plateau Mont-Royal ?

Individu B répond : La grosse femme d'a côté est enceinte ;
Thérèse et Pierrette à l'école des Saints-Anges ;
La duchesse et le roturier ;
Des nouvelles d'Edouard ;
Le premier quartier de la Lune ;
Un objet de beauté

Dans ce scénario, tout peut être relié à l'OAI. D'abord, **l'individu A est le moissonneur** (il initie les échanges) alors que **l'individu B est l'entrepôt** (il est passif, il ne fait que répondre). Le fait que les deux individus se rencontrent dans un café et qu'ils se comprennent en parlant le français décrit l'**architecture technologique** sur laquelle s'appuie le protocole OAI.

La conversion exprime deux **instructions OAI** et leur réponse. La première est tout simplement une demande d'identification pour savoir à qui on parle. La seconde est une demande pour l'envoi de métadonnées à propos d'objets (des livres) mais pas n'importe lesquels (ils font partie d'une suite). Il s'agit donc d'une moisson sélective de métadonnées.

Ce scénario est toutefois particulièrement utile pour montrer les limites du protocoles OAI. Par exemple, on n'a aucune idée de la manière dont l'individu B est devenu écrivain où comment il fait pour se rappeler de tous les livres qu'il a écrit dans cette série. De plus, on ne sait pas ce que l'individu B fera de l'information recueillie lors de la conversation ; peut-être l'oubliera-t-il quelques minutes plus tard.

C'est la même chose pour le protocole OAI : il définit les échanges de données, mais ne prescrit rien sur l'origine de ces données ni sur leur utilisation une fois recueillies.

Les objets documentaires

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Le protocole OAI définit trois objets fondamentaux : la **ressource**, l'**item** et l'**enregistrement** (*record*).

Ressource

La ressource est le véritable objet dont il est question dans une application documentaire. Autrement dit, si des métadonnées sont *des données à propos de quelque chose*, la ressource est ce quelque chose.

Une ressource peut être physique ou numérique. Un livre imprimé peut être une ressource, et une monographie numérique (numérisée par exemple) peut également en être une.

Le protocole OAI ne définit pas d'instructions qui s'intéressent aux ressources comme telles. Mais il est important de bien comprendre la nature de cet objet pour comprendre celle des deux autres.

Item

L'item est un **ensemble de données descriptives, structurées et numériques**, à propos de la ressource. Il s'agit en quelque sorte de la fiche signalétique, de catalogage, à propos de la ressource.

L'item est un objet bien réel pour un entrepôt OAI, mais le protocole ne spécifie pas comment ces items sont gérés, stockés, etc.

Enregistrement

L'enregistrement est un **jeu de métadonnées dans un format spécifique**. Il s'agit donc d'une manifestation réelle d'un item. Ce sont des enregistrements qui sont échangés par le protocole OAI.

Pour mieux comprendre ces trois types d'objets, prenons le cas relativement simple d'un catalogue de bibliothèque, catalogue qui serait géré à l'aide d'une base documentaire.

Dans ce catalogue, les *ressources* sont les documents que possède la bibliothèque : monographies, périodiques, etc. Le catalogue ne gère pas directement ces ressources.

Pour chaque document que possède la bibliothèque, le catalogue contient une fiche qui décrit ce document : titre, auteur, date de publication, etc. Concrètement, les utilisateurs – y compris les utilisateurs avertis – n'ont pas besoin de savoir comment est stockée ou gérée cette fiche et l'ensemble des fiches. Est-ce une base de données relationnelle ? Une base de données XML ? Un index lié à un moteur de recherche ? Peu importe. Ces fiches constituent les *items* au sens OAI.

Enfin, dans un souci d'interopérabilité, la base documentaire est en mesure d'exporter chaque fiche en XML, soit en Dublin Core, soit en MarcXML. Ainsi, pour chaque document que possède la bibliothèque, et donc pour chaque item, nous sommes en mesure d'obtenir deux fiches en format XML. Chacune de ces fiches constitue un *enregistrement* au sens OAI.

Le protocole OAI insiste également – et avec raison – sur l'importance des **identifiants** dans une infrastructure d'interopérabilité. Ainsi, un **item doit avoir un identifiant unique** qui l'identifie de manière unique parmi tous les items de l'entrepôt. Et c'est bien l'item qui possède cet identifiant unique, et ce même si cet item peut se manifester en plusieurs enregistrements, qui feront tous référence à ce même identifiant.

Ces définitions nous permettent déjà de préciser un point par rapport à une [méthodologie de mise en œuvre du protocole OAI](#). En effet, il sera fondamental de bien identifier les items et les enregistrements qui devront être gérés par notre système, que ce soit pour un entrepôt ou un moissonneur, ainsi que leur identifiant.

L'architecture technologique

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

L'architecture technologique du protocole OAI est très simple et s'articule autour de deux normes universellement supportées de nos jours : HTTP¹ et XML². Le protocole HTTP est utilisé pour le transport des données entre un entrepôt et un moissonneur. Le format XML est utilisé pour représenter concrètement les données échangées.

On peut s'arrêter ici pour mentionner l'un des aspects les plus intéressants du protocole OAI : on peut **jouer au moissonneur OAI à l'aide d'un simple navigateur Web qui peut afficher des documents XML**. En effet, un navigateur Web a pour principale fonction d'effectuer des requêtes HTTP et d'en recevoir les réponses. De plus, tous les navigateurs Web d'aujourd'hui permettent d'afficher très correctement du XML.

Nous avons donc un outil de test et de pédagogie sur tous nos postes de travail ou presque. Pour vous en convaincre, si vous consultez ce document de manière électronique en étant connecté à Internet, cliquez sur le lien suivant :

http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=GetRecord&metadataPrefix=oai_dc&identifieur=oai:archivesic.ccsd.cnrs.fr:sic_00000028

Vous obtiendrez à l'écran un affichage semblable à celui-ci (il a été tronqué) :

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH
  xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2005-06-01T03:32:50Z</responseDate>
  <request metadataPrefix="oai_dc"
    verb="GetRecord"
    identifieur="oai:archivesic.ccsd.cnrs.fr:sic_00000028">
    http://archivesic.ccsd.cnrs.fr/perl/oai20
  </request>
  <GetRecord>
  <record>
  <header>
    <identifieur>oai:archivesic.ccsd.cnrs.fr:sic_00000028</identifieur>
    <datestamp>2002-05-17</datestamp>
    <setSpec>sic_comm</setSpec>
    <setSpec>sic_docu</setSpec>
  </header>
  <metadata>
  <oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:creator>Le Coadic, Yves-François</dc:creator>
    <dc:description>Après avoir évoqué le rôle du nombre et de la
      mesure lors de l'évaluation des revues scientifiques...
    </dc:description>
    <dc:title>N(OMBRE) ou LUMIERE - usage des X-métries en science de
```

1 <http://www.w3.org/Protocols/>

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

```
    l'information et en science de la communication</dc:title>
    <dc:language>fr</dc:language>
    <dc:subject>Information retrieval</dc:subject>
    <dc:subject>Theory of information/communication</dc:subject>
    <dc:date>2002-05-17</dc:date>
    <dc:identifiant>
      http://archivesic.ccsd.cnrs.fr/sic_00000028.en.html
    </dc:identifiant>
    <dc:type>Text</dc:type>
  </oai_dc:dc>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
```

Félicitations ! Vous venez d'effectuer votre première requête OAI, sans installer d'outils, et sans complexité apparente. Concrètement, vous avez récupéré un enregistrement en format Dublin Core d'un ouvrage d'une base de données du CCSD au CNRS.

Bien entendu, le protocole OAI a été défini dans le but de permettre des applications plus sophistiquées que le simple affichage de données XML dans un navigateur Web, mais cette facilité de test est un élément clé du succès du protocole.

Une **requête OAI** est donc toujours une requête HTTP, qu'elle soit de type GET ou de type POST. Une requête OAI est toujours constituée d'une adresse de base propre à un entrepôt. Cette URL de base est complétée par une liste d'arguments sous la forme de paires `nom = valeur`, comme dans toute requête HTTP par ailleurs. Le protocole OAI définit justement quelles peuvent être ces paires `nom = valeur`.

Une **réponse OAI** (réponse faite par un entrepôt suite à une requête effectuée par un moissonneur) est toujours en format XML, sous la forme d'un seul document XML bien formé.

L'élément racine de ce document est toujours `OAI-PMH`, et il contient trois grandes informations : la date de la réponse, un rappel de la requête OAI qui a mené à cette réponse, et enfin le contenu de la réponse elle-même, qui est soit une erreur soit les données demandées par la requête. Le document XML retourné par la requête ci-dessus montre bien ces différentes parties d'une réponse OAI.

Cette relative simplicité dans l'architecture technologique du protocole OAI ne doit pas masquer le besoin de porter attention aux petits détails qui peuvent rendre une implémentation plus complexe que prévue : jeu de caractères, gestion des erreurs, compression des données, définition des espaces de noms, etc. Le protocole OAI regorge de petits **pièges** sur ces sujets, mais heureusement ce sont les personnes qui implémentent des outils OAI – et non ceux qui mettent en œuvre des services OAI avec ces outils – qui se font prendre dans ces pièges.

Mais si vous évaluez un outil OAI, prenez la peine de le soumettre à des tests variés et à le placer dans des conditions anormales pour vous assurer de sa conformité. Nous recommandons en particulier un outil de vérification des entrepôts qui effectue un diagnostic très détaillé de tout entrepôt accessible sur le Web. Il s'agit d'une fonctionnalité du site [Open Archives Initiative – Repository Explorer](#).

Les formats de données

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Les requêtes OAI sont des requêtes HTTP ; une réponse OAI est un document XML. Au-delà de ces considérations générales, nous allons nous attarder ici à une des composantes d'une réponse OAI lors d'une requête qui demande un enregistrement : le format de cet enregistrement.

Autrement dit, lorsqu'un moissonneur reçoit un enregistrement depuis un entrepôt OAI, quel est le format de données de cet enregistrement ?

L'utilisation des formats

Puisque la totalité de la réponse est dans un flux XML, le format de cet enregistrement est nécessairement un format XML. Mais la norme XML est générique, et recevoir n'importe quel XML pourrait être difficile à manipuler pour un moissonneur et les services associés. Par ailleurs, imposer un format spécifique dans un protocole généraliste de type OAI n'est pas souhaitable.

Pour résoudre ce dilemme, le protocole OAI a pris une approche pragmatique : **tout item dans un entrepôt OAI doit être disponible en format Dublin Core non qualifié**, mais **tout item dans un entrepôt OAI peut être disponible dans d'autres formats XML** qui ne sont pas prescrits par le protocole.

Bref, le protocole impose une contrainte de format (le Dublin Core simplifié) mais laisse une grande latitude pour utiliser d'autres formats. **Il faut donc en finir avec le mythe qui réduit le protocole OAI au format Dublin Core simplifié.**

Pour le protocole OAI, un format est en fait un *préfixe de métadonnées*. D'un point de vue technique, ce préfixe n'est pas à proprement parler un préfixe au sens des *XML Namespaces*, il s'agit vraiment d'un code qui identifie un format disponible dans un entrepôt. Ce code doit être accompagné d'un espace de nom (au sens XML, donc une URL en général) et de l'URL d'un schéma XML qui définit ce format.

L'instruction ListMetadataFormats permet de connaître les formats supportés par un entrepôt OAI. Voici un exemple de réponse à cette requête, effectuée par l'URL :

<http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListMetadataFormats>

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2005-06-01T04:11:06Z</responseDate>
  <request verb="ListMetadataFormats">
    http://archivesic.ccsd.cnrs.fr/perl/oai20
  </request>
  <ListMetadataFormats>
    <metadataFormat>
      <metadataPrefix>ccsd_tel</metadataPrefix>
      <schema>http://tel.ccsd.cnrs.fr/OAI/2.0/ccsd_tel.xsd</schema>
      <metadataNamespace>
        http://tel.ccsd.cnrs.fr/OAI/2.0/ccsd_tel
      </metadataNamespace>
    </metadataFormat>
    <metadataFormat>
      <metadataPrefix>oai_dc</metadataPrefix>
```

```
<schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</schema>
<metadataNamespace>
  http://www.openarchives.org/OAI/2.0/oai_dc/
</metadataNamespace>
</metadataFormat>
</ListMetadataFormats>
</OAI-PMH>
```

Il existe donc quelques contraintes importantes pour l'utilisation d'un format avec le protocole OAI :

- il doit s'agir d'un **format XML**
- un **schéma XML** (norme du W3C) doit être disponible sur Internet pour définir ce format XML (cela exclut notamment l'usage d'autres normes de grammaires XML telles que les DTD ou RelaxNG)
- un **espace de nom** (XML Namespace) doit être associé au format (ce qui n'est pas strictement nécessaire en XML)

Ces contraintes sont relativement légères mais peuvent entraîner des efforts supplémentaires dans la mise en œuvre de l'OAI.

Le Dublin Core simplifié

La *Dublin Core metadata Initiative*³ est un forum engagé dans la définition de formats de métadonnées interopérables pour des usages sur Internet. Cette initiative est mieux connue pour son format éponyme *Dublin Core*⁴, mais en même temps ce format n'est pas toujours aussi bien connu qu'on le croit.

D'abord, **il s'agit d'un format conceptuel, documentaire, et non d'un format informatique**. Bref, le format définit des champs et ce qu'on devrait y mettre, mais il ne prescrit pas de méthodes pour représenter, stocker ces champs. Le Dublin Core ne spécifie pas, par exemple, qu'une notice doit être stockée en XML, et encore moins avec quels éléments.

Toutefois, le Dublin Core **propose** des représentations des métadonnées qu'il définit⁵, et certaines sont en format XML. Plus particulièrement, il propose des schémas XML⁶, mais ces schémas ne sont pas complets car ils ne définissent pas, notamment, l'élément racine qui contient les métadonnées.

Le protocole OAI impose un schéma XML pour un enregistrement en format Dublin Core⁷, en reprenant le schéma recommandé par le Dublin Core et en lui ajoutant un élément racine. Cela donne une structure XML telle que celle-ci :

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
  http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:title xml:lang="en">Grassmann's space analysis</dc:title>
  <dc:creator>Hyde, E. W. (Edward Wyllys)</dc:creator>
  <dc:subject>LCSH:Ausdehnungslehre; LCCN QA205.H99</dc:subject>
  <dc:publisher>J. Wiley & Sons</dc:publisher>
  <dc:date>Created: 1906; Available: 1991</dc:date>
  <dc:type>text</dc:type>
  <dc:identifiant>
    http://resolver.library.cornell.edu/math/1796949
```

3 <http://www.dublincore.org/>

4 <http://www.dublincore.org/documents/dcmi-terms/>

5 <http://www.dublincore.org/documents/usageguide/#whichtsyntax>

6 <http://www.dublincore.org/schemas/xmls/>

7 http://www.openarchives.org/OAI/2.0/oai_dc.xsd

```

</dc:identifiant>
<dc:language>english</dc:language>
<dc:rights xml:lang="en">Public Domain</dc:rights>
</oai_dc:dc>

```

Le format Dublin Core comporte un ensemble de champs qui peuvent être qualifiés. Toutefois, le format Dublin Core imposé par le protocole OAI est la version non qualifiée, ou simple, qui est uniquement composée de quinze champs, qui ne peuvent pas être précisés⁸. Ces quinze champs sont définis dans le tableau suivant :

Nom	Description
dc:identifiant	Identifiant
dc:title	Nom ou titre
dc:subject	Indexation sujet
dc:description	Indications sur le contenu du site, description de la ressource
dc:date	Date de création ou de publication ou toute autre date ayant un rapport avec la vie de la ressource
dc:coverage	Couverture spatiale ou temporelle de la ressource. Dans le Dublin Core, ces deux notions ne sont pas distinctes.
dc:creator	Responsable du contenu intellectuel de la ressource. Par exemple, institution ayant réalisé un site Web
dc:contributor	Responsable secondaire du contenu intellectuel de la ressource, par ex. illustrateur, traducteur...
dc:publisher	Responsable de la diffusion de la ressource
dc:type	Nature de la ressource, exemple : image, text, collection.
dc:format	Format des données. Exemple : jpeg, text/html. Dans le Dublin Core, cet élément permet aussi de décrire les formats physiques des documents. Il n'est utilisable ici que pour les formats électroniques.
dc:language	Langue de l'unité documentaire
dc:source	Identifiant d'une autre ressource dont est dérivée la présente unité documentaire. Par exemple dans le cas d'un article, identifiant du périodique dont il est extrait sous la forme titre, volume, numéro
dc:relation	Référence à une autre ressource
dc:rights	Droits de propriété intellectuelle, copyright, etc.

C'est tout ce que vous avez à votre disposition lorsque vous travaillez avec le Dublin Core non qualifié, par exemple dans un contexte OAI. Vous avez un champ pour la date, un champ pour le titre, un champ pour la couverture spatiale et temporelle, etc. C'est peu, mais cela permet une assez grande interopérabilité car c'est facile à mettre en œuvre.

Mais n'oubliez pas : **vous pouvez utiliser d'autres formats avec le protocole OAI.**

⁸ <http://www.dublincore.org/documents/dces/>

Les instructions et paramètres OAI

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Ce document a pour objectif de décrire les différentes instructions OAI. Cette description est intéressante et ce même si par moment elle peut paraître technique. En effet, elle permet de voir la totalité des fonctionnalités du protocole et donc de mieux comprendre comment on peut l'utiliser pour atteindre nos objectifs d'interopérabilité.

De plus, il ne faut pas oublier que l'on peut tester le protocole OAI simplement en utilisant un navigateur Web pour moissonner. En connaissant les différentes instructions OAI et leurs paramètres, vous pourrez ainsi explorer et tester à votre guise.

Il existe également un service pratique sur le Web, le *Open Archives Initiative - Repository Explorer*⁹. Il vous permet, à l'aide d'une interface Web, de lancer des instructions OAI à différents entrepôts et de voir les réponses, et ce sans complexité technique. Vous devez toutefois comprendre le protocole pour utiliser efficacement cet outil.

Les instructions et leurs paramètres

Le protocole OAI est composé de six instructions (*verb* en anglais). Le tableau suivant décrit ces instructions, en mentionnant les paramètres qui peuvent être associés. Un paramètre suivi d'un astérisque indique qu'il est obligatoire, mais l'astérisque ne fait pas partie du nom du paramètre.

Instruction	Paramètres	Description
Identify		Demande des informations à propos de l'entrepôt. Ces informations sont plutôt destinées à une consommation humaine, et non informatique.
ListMetadataFormats	identifiant	Demande la liste des formats disponibles. Sans paramètre, tous les formats disponibles pour au moins un item sont retournés. Avec le paramètre <code>identifiant</code> , les formats disponibles pour un item précis sont retournés.
ListSets	resumptionToken	Demande la liste des ensembles disponibles dans cet entrepôt. La réponse peut être sur plusieurs pages.
ListIdentifiers	from until metadataPrefix* set resumptionToken	Demande la liste des identifiants des items d'un entrepôt. On peut limiter cette demande par date et par ensemble. La réponse peut être sur plusieurs pages.
ListRecords	from until metadataPrefix* set resumptionToken	Demande la liste des enregistrements dans un format spécifique. On peut limiter cette demande par date et par ensemble. La réponse peut être sur plusieurs pages. Il s'agit de l'instruction la plus utile pour moissonner le contenu d'un entrepôt OAI.
GetRecord	identifiant* metadataPrefix*	Demande un enregistrement spécifique, par son identifiant, en indiquant le format souhaité.

9 <http://re.cs.uct.ac.za/>

Un conseil pour retenir le nom exact des instructions et des paramètres : les instructions débutent par une majuscule, mais les paramètres débutent par une minuscule. Dans les deux cas, la première lettre de chaque partie significative suivante du nom est en majuscule.

La syntaxe pour utiliser ces instructions et paramètres dans une requête OAI est de la forme suivante :

[adresse de l'entrepôt]?verb=[instruction]&[paramètre]=valeur&[paramètre]=valeur

Voici un exemple avec deux paramètres :

http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=GetRecord&metadataPrefix=oai_dc&identifiant=oai:archivesic.ccsd.cnrs.fr:sic_00000028

Les paramètres

Le tableau précédent a montré les paramètres qui peuvent être utilisés dans une requête OAI. On en retrouve six seulement, qui sont décrits dans le tableau suivant.

Paramètre	Instructions associées	Description
metadataPrefix	ListIdentifiers ListRecords GetRecord	Indique le format de métadonnées à utiliser, par son préfixe. Le préfixe du format Dublin Core, obligatoire, est oai_dc.
identifiant	ListMetadataFormats GetRecord	Indique un item précis par son identifiant.
from	ListIdentifiers ListRecords	Permet la moisson sélective en indiquant de retourner des enregistrements correspondant à des items modifiés depuis une certaine date. La date doit être en UTC exprimée selon la norme ISO8601, par exemple 2002-02-08T08:55:46Z ou encore 2002-02-08, selon la granularité acceptée par l'entrepôt.
until	ListIdentifiers ListRecords	Permet la moisson sélective en indiquant de retourner des enregistrements correspondant à des items modifiés avant une certaine date. La date doit être en UTC exprimée selon la norme ISO8601, par exemple 2002-02-08T08:55:46Z ou encore 2002-02-08, selon la granularité acceptée par l'entrepôt.
set	ListIdentifiers ListRecords	Permet la moisson sélective en indiquant de retourner des enregistrements qui font partie d'un ensemble précis.
resumptionToken	ListSets ListIdentifiers ListRecords	Un entrepôt peut retourner les réponses à ces trois instructions en plusieurs pages pour limiter les ressources utilisées et les risques d'erreurs. Dans ce cas, avec chaque page, il doit retourner un code qui permet de demander la suite. Ce code doit être indiqué par ce paramètre lorsque le moissonneur demande cette suite.

La moisson sélective

Les paramètres `from`, `until` et `set` permettent une moisson sélective, par date de modification pour les deux premiers et par ensemble pour le dernier.

Par date de modification

Les items exposés par un entrepôt OAI peuvent être modifiés. C'est le cas lorsqu'une fiche descriptive est corrigée. Mais une modification peut aussi être un ajout d'item, ou même la suppression d'un item (ce dernier cas sera traité plus loin).

Il serait inefficace qu'un moissonneur demande systématiquement les enregistrements correspondant à tous les items lorsqu'il moissonne un entrepôt. Un scénario plus logique consisterait, par exemple, à faire une moisson périodique (une fois par jour par exemple) et à demander seulement les enregistrements correspondant à des items modifiés depuis la dernière moisson (la veille par exemple).

Les paramètres `from` et `until` permettent de préciser les dates de modification lorsqu'on demande une liste d'identifiants ou une liste d'enregistrements. Le cas le plus fréquent sera l'utilisation du paramètre `from` avec comme valeur la date de la dernière moisson. Dans ce cas, l'entrepôt doit être en mesure de retourner uniquement les enregistrements qui ont été modifiés (ou ajoutés, ou détruits) depuis cette date. La nature de la réponse ne change pas, seul son contenu sera différent (ce ne seront pas nécessairement les mêmes enregistrements).

Par ensemble

Un entrepôt peut définir des ensembles (il s'agit d'une fonctionnalité optionnelle pour un entrepôt OAI), et dans ce cas un item peut appartenir à un ou plusieurs ensembles. La sémantique des ensembles n'est bien sûr pas définie par le protocole. Il peut s'agir d'une liste de sujets (un sujet est un ensemble), d'une liste de types de documents (un type de documents est un ensemble), etc.

On peut connaître les ensembles disponibles dans un entrepôt OAI avec le verbe `ListSets`. Mais surtout, on peut moissonner des enregistrements qui sont dans un ensemble (et non tous les enregistrements) en utilisant le paramètre `set` avec les instructions `ListIdentifiers` et `ListRecords`. La moisson sélective par ensemble peut être combinée avec la moisson sélective par date.

La nature de la réponse n'est pas modifiée par l'utilisation du paramètre `set`, seul son contenu sera différent (ce ne seront pas nécessairement les mêmes enregistrements).

Les enregistrements supprimés

Un entrepôt OAI peut (ce n'est pas obligatoire) gérer la notion d'enregistrements supprimés. Cette fonctionnalité est très utile lorsque l'on utilise le protocole OAI pour constituer et mettre à jour un ensemble de données depuis différentes sources. Si le moissonneur n'a pas connaissance des enregistrements supprimés, il ne saura pas quels sont les données à supprimer de son côté, et son contenu ne reflétera pas fidèlement le contenu des sources moissonnées.

Lorsqu'on fait une moisson sélective par date, ou lorsqu'on demande un enregistrement précis, un entrepôt qui supporte les enregistrements supprimés doit les indiquer, en ne retournant pas le contenu de l'enregistrement mais uniquement son en-tête, incluant son identifiant et sa date de suppression.

A noter que ce sont des enregistrements qui sont détruits, et non des items. La plupart des outils OAI ne permettent pas de gérer finement cette distinction toutefois.

Quelques exemples

Les URL ci-dessous permettent d'utiliser un navigateur Web pour explorer les différentes facettes du protocole OAI. Pour une exploration plus approfondie, nous vous conseillons d'utiliser *Open Archives Initiative - Repository Explorer*¹⁰.

¹⁰ <http://re.cs.uct.ac.za/>

Instruction *Identify*

L'instruction `Identify` permet de mieux connaître l'entrepôt. Nous allons travailler avec deux entrepôts, que nous présentons ici à l'aide de leur réponse `Identify` :

Universidad Lyon 2 – Thèses électroniques

<http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=Identify>

@rchiveSIC : Sciences de l'Information et de la Communication

<http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=Identify>

La réponse nous indique que tous deux supportent les documents supprimés.

Instruction *ListMetadataFormats*

Les requêtes suivantes permettent de connaître les formats de métadonnées disponibles. Pour @rchiveSIC, nous avons les formats `oai_dc` et `ccsd_tel` :

<http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListMetadataFormats>

Pour l'université de Lyon 2, nous avons les formats `formats oai_dc` et `oai_etdms` :

<http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListMetadataFormats>

On peut également connaître les formats pour un enregistrement spécifique, par exemple :

<http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListMetadataFormats&identifier=dx:demeter.univ-lyon2.fr:8080:theses/documents/lyon2.1998.jserme-principal>

Instruction *ListSets*

L'instruction `ListSets` permet de connaître les ensembles disponibles pour un entrepôt. Voici le résultat pour @rchiveSIC :

<http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListSets>

On constate qu'il s'agit d'une liste de sujets. Pour l'Université de Lyon 2, les ensembles ne sont pas supportés :

<http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListSets>

Instruction *ListIdentifiers*

L'instruction `ListIdentifiers` permet de connaître les identifiants, si ce sont les seules informations nécessaires. Sans critères de sélection, les deux entrepôts de test nous retournent ces informations :

http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListIdentifiers&metadataPrefix=oai_dc

http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListIdentifiers&metadataPrefix=oai_dc

On peut modifier ces requêtes pour obtenir uniquement les enregistrements modifiés pendant l'année 2004 dans l'entrepôt @rchiveSIC :

http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListIdentifiers&metadataPrefix=oai_dc&from=2004-01-01&until=2004-12-31

Idem pour les enregistrements modifiés en janvier 2006 pour l'Université de Lyon 2 :

```
http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListIdentifiers&metadataPrefix=oai_etdms&from=2006-01-01T00:00:00Z&until=2006-01-31T23:59:59Z
```

On constate que les deux entrepôts ne supportent pas la même granularité des dates, ce qui est prévu par le protocole OAI.

Pour une moisson périodique, on pourrait par exemple demander uniquement les enregistrements modifiés depuis décembre dernier :

```
http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListIdentifiers&metadataPrefix=oai_etdms&from=2005-02-01T00:00:00Z
```

On peut également s'intéresser uniquement à des enregistrements en géopolitique avec cette requête sur l'entrepôt @rchiveSIC :

```
http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListIdentifiers&metadataPrefix=oai_dc&set=sic_geop
```

Instruction ListRecords

L'instruction `ListRecords` fonctionne de la même façon que `ListIdentifiers`, mais retourne les enregistrements complets. Nous reprenons ici les mêmes paramètres :

```
http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListRecords&metadataPrefix=oai_dc
```

```
http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListRecords&metadataPrefix=oai_etdms
```

```
http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListRecords&metadataPrefix=oai_dc&from=2004-01-01&until=2004-12-31
```

```
http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListRecords&metadataPrefix=oai_etdms&from=2006-01-01T00:00:00Z&until=2006-01-31T23:59:59Z
```

```
http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=ListRecords&metadataPrefix=oai_etdms&from=2005-12-01T00:00:00Z
```

```
http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=ListRecords&metadataPrefix=oai_dc&set=sic_geop
```

Instruction GetRecord

L'instruction `GetRecord` retourne un enregistrement uniquement. Voici deux URL pour retourner un item en deux formats différents dans l'entrepôt @rchiveSIC :

```
http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=GetRecord&metadataPrefix=oai_dc&identifiant=oai:archivesic.ccsd.cnrs.fr:sic_00000028
```

```
http://archivesic.ccsd.cnrs.fr/perl/oai20?verb=GetRecord&metadataPrefix=ccsd_tel&identifiant=oai:archivesic.ccsd.cnrs.fr:sic_00000028
```

Voici des URL équivalents pour l'entrepôt de l'Université de Lyon 2 :

```
http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=GetRecord&metadataPrefix=oai_dc&identifiant=sdx:demeter.univ-lyon2.fr:8080:theses/documents/lyon2.2000.chantin_principal
```

http://demeter.univ-lyon2.fr:8080/sdx/sdx/oai/theses/documents?verb=GetRecord&metadataPrefix=oai_etdms&identifiant=sdx:demeter.univ-lyon2.fr:8080:theses/documents/lyon2.2000.chantin_r-principal

Pour en savoir plus

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Il existe de nombreuses ressources pour en savoir plus sur le protocole OAI. Dans ce court document, nous nous concentrons sur les aspects techniques du protocole.

Le point de départ est bien sûr le site *Open Archives Initiative* :

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

Les spécifications du protocole sont en ligne, et elles constituent le seul document de référence sur le sujet :

<http://www.openarchives.org/OAI/openarchivesprotocol.html>

Si vous souhaitez approfondir vos connaissances techniques sur l'OAI, pourquoi ne pas regarder un des nombreux outils libres qui implémentent le protocole ? Une liste est disponible ici :

<http://www.openarchives.org/tools/tools.html>

Mais la page la plus importante pour comprendre les aspects techniques de l'usage de l'OAI est probablement celle qui référence les guides utilisateurs. Ces guides contiennent des recommandations et des bonnes pratiques :

<http://www.openarchives.org/OAI/2.0/guidelines.htm>

Panorama des utilisations du protocole OAI dans le monde et en France

2005-05-25 / 2006-02-02

Jean-Paul Ducasse (Université Lyon 2, France)

Kim Danière (Université Lyon 2, France)

Magalie Prudon (Université Lyon 2, France)

La présentation de Jean-Paul Ducasse est disponible sur Internet :

<http://linux.univ-lyon2.fr/ADNX/oai.html>

Méthodologie de mise en œuvre de l'OAI

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Débuter un projet où le protocole OAI joue un rôle important ou intégrer le protocole OAI à un projet existant sont des activités de plus en plus fréquentes pour des gestionnaires de bases documentaires. Il serait prétentieux de considérer les indications qui vont suivre comme une véritable méthodologie de mise en œuvre de l'OAI. Nous souhaitons plutôt essayer d'identifier les questions importantes à se poser avant de se lancer tête baissée dans un tel projet.

Avant toute chose

Est-ce que le protocole OAI est pertinent ? Est-ce que le protocole OAI sera une réponse efficace à mes besoins ? Ces deux questions ne sont pas identiques, mais elles sont tout aussi importantes.

Le protocole OAI apporte des solutions pour l'échange structuré et automatisé de données à propos de ressources ; il faut d'abord savoir si le projet comporte un tel besoin. D'un autre côté, le protocole OAI n'est pas à proprement parler un outil de recherche multi-sources, comme peuvent l'être les protocoles Z39.50 ou SRU/SRW par exemple. Il ne faut pas se tromper d'objectifs, ou mieux, il faut choisir les bons outils pour atteindre nos objectifs.

Le protocole OAI suppose deux acteurs au moins : un service qui moissonne, un entrepôt qui expose ses données. Quel(s) rôle(s) souhaitez-vous jouer dans votre projet ? Avec qui ? Avec quelle institution, quelle application ?

Les objets documentaires et leur mise en ligne

Pour mettre en place un entrepôt OAI

Le protocole OAI est basé sur la notion d'**item**, qui sont des fiches descriptives de ressources. Avez-vous de telles fiches ? Pour décrire quoi ? Est-ce que vos fiches possèdent des identifiants uniques et pérennes ?

Ou peut-être avez-vous des ressources numériques plus volumineuses, des monographies par exemple ? Dans ce cas est-ce que vous pouvez en extraire des **métadonnées** ?

Pouvez-vous identifier facilement des « **unités documentaires** » qui deviendront des items en OAI ?

Quels **formats** pouvez-vous produire à partir de vos fiches ? Et en particulier pouvez-vous facilement générer le format **Dublin Core** simplifié ?

Quel **vocabulaire** utilisez-vous dans vos différents champs ? Est-ce qu'il est conforme à des standards internationaux ?

Pour un moissonneur OAI

Que souhaitez-vous faire avec les données moissonnées ? Avez-vous des données qui ne proviennent pas d'une moisson OAI ? Sont-elles dans le même format ? Doivent-elles être traitées de la même manière ?

Quel(s) **formats** voulez-vous moissonner ? Allez-vous utiliser un **format pivot** pour harmoniser les données moissonnées avec, éventuellement, vos autres données ?

Le **vocabulaire** utilisé dans les données moissonnées est-il conforme à celui que vous utilisez pour vos propres données ? La langue des données est-elle compatible avec vos propres données ou votre système ?

Les outils

Avez-vous déjà un site Web avec des données ou pour accueillir des données ? Quelle technologie est utilisée ? Y a-t-il un *module OAI* de disponible ? Es-il actif ? Combien coûte-t-il ?

Votre hébergement permet-il l'ajout d'un *module OAI* et permet-il des échanges OAI ?

Qui pourrait intervenir pour mettre en œuvre les aspects informatiques de votre projet OAI ?

Mettre en place une archive institutionnelle avec DSpace

2006-01-27 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Ce document contient quelques informations sur la démonstration qui a eu lieu dans le cadre de l'atelier.

Contexte

DSpace est un logiciel qui permet de mettre en place un site de diffusion de documents numériques. Il a été conçu dans le but de faciliter la mise en oeuvre d'archives institutionnelles ouvertes. DSpace support le protocole OAI : il permet de rendre disponible les métadonnées à propos des documents sous la forme d'un entrepôt OAI.

DSpace est un logiciel libre (licence BSD).

Objectifs

L'objectif est de montrer comment on peut utiliser DSpace pour publier quelques documents et ensuite les rendre disponible via un entrepôt OAI. Pour ce faire, une installation locale de DSpace sera utilisée, avec un utilisateur et quelques documents publiés dans une collection spécifique.

Informations techniques

DSpace est une application Web de type servlet qui fonctionne donc avec Java et un moteur tel que Tomcat. De plus, DSpace s'appuie sur le SGBD PostgreSQL (qui peut être remplacé par Oracle moyennant configuration).

Ces outils sont disponibles sur plusieurs plates-formes. Même si le logiciel a été conçu et documenté pour fonctionner sous Linux, il peut s'installer sur Windows. La procédure doit être suivie pas à pas, mais elle fonctionne bien.

Lorsqu'on installe DSpace, on installe en fait deux applications Web. L'une d'entre elle est permet de gérer le site d'archives ouvertes comme tel. L'autre est l'interface OAI pour accéder à ce site.

Références

- DSpace : <http://www.dspace.org/>

Transformer une base de données MySQL en entrepôt OAI avec phpoi2

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Ce document contient quelques informations sur la démonstration qui a eu lieu dans le cadre de l'atelier.

Contexte

MySQL est un système de gestion de bases de données relationnelles libre. Il est communément utilisé pour des applications dynamiques sur le Web.

PHP est un langage de programmation pour créer des sites Web dynamiques. Très utilisé pour des sites simples en interaction avec des bases de données MySQL notamment.

phpoi2 est un outil écrit en PHP qui permet aisément de transformer une base de données MySQL en entrepôt OAI.

La démonstration se fera sur une application précise : une base de données descriptives des pièces du musée de la Ville de Sarlat.

Cette base de données est conçue en interne à l'aide du logiciel ActiveMuseo. Elle est diffusée maintenant sur Internet grâce au module Internet du logiciel. Ce module est en fait une collection de scripts PHP qui agissent sur les données exportées dans une base de données MySQL. Les outils OAI viennent donc se greffer sur ces scripts, pour les compléter par des fonctions d'entrepôt OAI.

Objectifs

L'objectif est de montrer comment on peut utiliser des technologies libres et populaires pour créer un entrepôt OAI. Nous allons installer et configurer phpoi2 pour exposer les données en format Dublin Core (obligatoire) et en format Aquitaine Patrimoines.

Informations techniques

MySQL version 4.1.9 sur Windows XP est utilisé. La base de données est explorée et modifiée à l'aide de phpMyAdmin version 2.1.6. PHP est en version 4.3.10, appuyé sur un Apache 1.3. Tous ces outils sont libres, avec différentes licences. La version de phpoi2 utilisée est la 1.11. Il s'agit d'un logiciel libre sous licence GPL.

Table MySQL

La table MySQL a été créée et remplie depuis un fichier d'instructions SQL provenant de l'exportation d'une autre base. La structure a été légèrement modifiée pour ajouter des informations sur les dates de modification et permettre les ensembles :

```
alter table piece add column (moddate TIMESTAMP);
update piece set moddate=now();
alter table piece add column (oai_set VARCHAR(255));
```

Configuration phpoi2

La configuration de cet outil est stockée dans le fichier `oai2/oaidp-config.php`. L'installation comme telle est très simple et n'a demandé que de configurer la variable qui pointe sur les outils d'accès aux bases de données depuis PHP (PEAR).

Ensuite, il faut configurer l'accès à notre base de données MySQL. Il faut pour cela modifier des valeurs de variables telles que `$DB_HOST`, `$DB_NAME`, etc. Le fichier de configuration est très explicite à ce sujet.

Enfin, il faut modifier des variables concernant notre entrepôt OAI spécifique. Ce sont des variables telles que `$repositoryName` et surtout `$METADATAFORMATS`.

Sortie des formats

Chaque format de sortie est produit à l'aide de son propre fichier PHP, ce qui facilite la maintenance. Il existe déjà un fichier `record_dc.php` pour le Dublin Core que nous avons utilisé comme exemple. Il existe des fonctions dans le script pour faciliter la sortie des informations.

Les commandes du script propre à un format sont exécutées avec comme contexte un enregistrement dans la table de données, cet enregistrement étant dans le tableau `$record`. Il s'agit alors d'un travail PHP pour produire correctement le format souhaité.

Références

- MySQL : <http://www.mysql.com/>
- PHP : <http://www.php.net/>
- phpoi2 : <http://physnet.uni-oldenburg.de/oai/>
- Site Web du musée de Sarlat : <http://www.musee-sarlat.com/>

Configurer un entrepôt OAI depuis une application SDX

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Ce document contient quelques informations sur la démonstration qui a eu lieu dans le cadre de l'atelier.

Contexte

La **plate-forme SDX** est un moteur de recherche documentaire pour corpus de documents XML. Une application SDX peut être constituée de plusieurs bases de documents. Une base de documents est un ensemble logique où s'effectue des recherches (la recherche dans plusieurs bases à la fois est également possible). Toute base de documents SDX peut être transformée en entrepôt OAI par simple configuration.

XToGen est un générateur d'applications SDX. A partir d'un fichier de configuration XML, XToGen fabrique tous les fichiers nécessaires pour obtenir une application SDX complète, avec saisie, affichage HTML et PDF, recherche simple ou avancée, importation, exportation, etc.

Objectifs

L'objectif est de montrer comment on peut transformer une base de documents SDX en entrepôt OAI. La démonstration se fera sur deux applications :

- une application *Dublin Core*, préparée par Michel Bottin et générée par XToGen, verra sa base de documents exposée en entrepôt OAI par simple correspondance de champs.
- l'application *sdxtest*, livrée avec SDX, verra sa base de documents *sdxworld* exposée en OAI en deux formats, un format Dublin Core (obligatoire) et le format natif, à l'aide de transformations XSLT

Informations techniques

La version de SDX utilisée est le dernier CVS de la branche V_22, qui correspond à une version 2.2.2 qui sera distribuée prochainement.

La version de XToGen utilisée pour générer l'application Dublin Core est la 2.0b5.

Pour atteindre un entrepôt OAI hébergé par SDX, l'URL est de cette forme :

[serveur] / [sdx] / sdx/oai / [appli] / [bd]

Ici, [sdx] est le chemin d'accès à SDX sur le serveur ; [appli] est le chemin de l'application concernée, [bd] est le code de la base de documents exposée en OAI.

Configuration de l'application DC

Cette application était déjà préparée pour exposer les données en OAI. Nous lui avons toutefois ajouté la sortie du champ `dc:subject` et nous avons ajouté la possibilité de filtrer le contenu de la base pour exclure des documents de l'entrepôt OAI.

La méthode de correspondance de champs est ici utilisée. Très simple à mettre en œuvre, elle ne fonctionne que pour des formats simples, à plat.

Configuration de l'application sdxtest

L'application `sdxtest` est déjà configurée pour exposer les données en OAI, en format Dublin Core et dans un format SDX qui correspond à la source des documents associés à leurs champs indexés.

Nous avons ajouté un troisième format pour illustrer la méthode de production d'un format à l'aide d'un *pipeline* SDX, plus précisément une XSLT. Cette XSLT très simple modifie les données pour produire un nouveau format défini pour l'exemple.

Références

SDX

- site Web officiel : <http://adnx.org/sdx/>
- site de développement : <http://savannah.nongnu.org/projects/sdx/>
- documentation : <http://www.nongnu.org/sdx/docs/html/doc-sdx2/fr/index.html>
- documentation OAI : <http://www.nongnu.org/sdx/docs/html/doc-sdx2/fr/oai/index.html>

XToGen

- site Web officiel : <http://xtogen.tech.fr/>
- documentation : <http://xtogen.tech.fr/fr/index.html>

Moissonner des sites distants avec SDX

2005-05-25 / 2006-02-02

Sévigny, Martin (AJLSM, France)

Ce document contient quelques informations sur la démonstration qui a eu lieu dans le cadre de l'atelier.

Contexte

La **plate-forme SDX** est un moteur de recherche documentaire pour corpus de documents XML. Une application SDX peut être constituée de plusieurs bases de documents. Une base de documents est un ensemble logique où s'effectue des recherches (la recherche dans plusieurs bases à la fois est également possible). Toute base de documents SDX peut être transformée en entrepôt OAI par simple configuration.

Objectifs

L'objectif est de montrer comment on peut moissonner des entrepôts OAI avec une application SDX. La démonstration utilisera les entrepôts OAI définis préalablement avec DSpace, PHP/MySQL et SDX.

Cette démonstration permettra de voir comment on peut récupérer des données distantes pour constituer un *portail* et les difficultés qui peuvent être rencontrées dans ce genre d'exercice.

Informations techniques

La version de SDX utilisée est le dernier CVS de la branche V_22, qui correspond à une version 2.2.2 qui sera distribuée prochainement.

La version de XToGen utilisée pour générer l'application Dublin Core est la 2.0b5.

Références

SDX

- site Web officiel : <http://adnx.org/sdx/>
- site de développement : <http://savannah.nongnu.org/projects/sdx/>
- documentation : <http://www.nongnu.org/sdx/docs/html/doc-sdx2/fr/index.html>

- documentation OAI : <http://www.nongnu.org/sdx/docs/html/doc-sdx2/fr/oai/index.html>

XToGen

- site Web officiel : <http://www.greenstone.org/>
- documentation : <http://xtogen.tech.fr/fr/index.html>