Anexo I – Hoja informativa 2

ADAPTACIÓN DE DSPACE AL ESTÁNDAR ESE V3.4 DE EUROPEANA

EUROPEANA:OBJECT – USO DEL SOFTWARE XPDF

Este documento es una guía técnica para la adaptación del software de repositorios institucionales *DSpace* al estándar *Europeana Semantics Elements v3.4*, en adelante *ESE*, profundizando en la generación de thumbnails (miniaturas) en el repositorio recomendadas por dicho estándar. El proceso consiste en hacer una correspondencia entre los metadatos en formato DC, que se almacenan en los registros de Dspace, y los metadatos especificados en el estándar ESE. Esto supone que alguno de los pasos a seguir puede variar ligeramente dependiendo de los metadatos que cada repositorio utilice para sus registros.

Además de los metadatos, también deben sustituirse diferentes cabeceras para adaptarlas al formato ESE y añadir algunos elementos obligatorios del estándar (por ejemplo, Europeana Provider).

El resultado final será un fichero XML que será accesible a través de la aplicación OAI de Dspace.

Esta adaptación está probada a partir de la versión 1.6.2 de *DSpace*.

Paso 1 - Instalación del plugin *ESECrosswalk*

El primer paso que hay que realizar es instalar un plugin para la diseminación de los metadatos Dublin_Core de Dspace en formato ESE en su versión 3.4. Para ello hay que descargar el archivo .java de la url http://vbanos.gr/ese/plugin/v4/ESECrosswalk.java y situarlo en [dspace-src]/dspace-oai/dspace-oai-api/src/main/java/org/dspace/app/oai/.
En esta clase java hay que realizar una serie de cambios fundamentales para cumplir el estándar.

1.1 El primero de ellos es sustituir la línea en la que pone:

```
metadata.append("<europeana:record
xmlns:europeana=\"http://www.europeana.eu/schemas/ese/\"
xmlns:dc=\"http://purl.org/dc/elements/1.1/\">");
```

por la siguiente:

```
metadata.append("<europeana:record xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
instance\" xmlns:europeana=\"http://www.europeana.eu/schemas/ese/\"
xmlns:dc=\"http://purl.org/dc/elements/1.1/\"
```

```
xsi:schemaLocation=\"http://www.europeana.eu/schemas/ese/
http://www.europeana.eu/schemas/ese/ESE-V3.4.xsd\">");
```

Así nos aseguramos que el esquema corresponda a la versión 3.4 de *ESE*.

1.2 Tras esto, añadimos las siguientes líneas que hacen de contadores para los elementos dc.format y dc.identifier.

```
int contadorDcIdentifier = 0;
int contadorDcFormat = 0;
```

1.3 El siguiente cambio se refiere al bloque comprendido en las líneas:

```
if (!notAcceptedLanguageString.contains(language)){

    language = " xml:lang=\""+language+"\">";

}else{
    language = ">";
}
```

Hay que eliminar ese bloque por éste:

```
language = ">";
```

De esta manera se elimina la etiqueta xml:lang=es, como manda la recomendación de Europeana.

1.4 Cambiamos cualquier elemento dc.identifier.citation por dc.source añadiendo tras el bloque:

```
if ("contributor".equals(element) && "author".equals(qualifier))
{
  element = "creator";
}
```

la línea:

```
if("identifier".equals(element) && "citation".equals(qualifier)){
  element = "source";
}
```

1.5 Tras la línea:

```
xmlMatcher.appendTail (valueBuf);
```

Hay que poner:

```
if ("identifier".equals(element)){
contadorDcIdentifier++;
}

if ("format".equals(element)){
contadorDcFormat++;
}
```

Que es un contador para controlar que sólo haya un dc.identifier y un dc.format.

1.6 El bloque:

```
metadata.append("<dc:").append(element).append(language).append(

valueBuf.toString()).append("</dc:").append(element).append(">");
```

Hay que englobarlo en un bloque if de la forma:

```
if (!(element.equals("subject") && valueBuf.toString().startsWith("CDU::")) &&
!(element.equals("identifier") && !valueBuf.toString().equals("-") &&
!("date".equals(element) && !"issued".equals(qualifier)) && !("identifier".equals(element)
&& contadorDcIdentifier>1) && !("format".equals(element) && contadorDcFormat>1))
  {
metadata.append("<dc:").append(element).append(language).append(
valueBuf.toString()).append("</dc:").append(element).append(">");
}
```

De manera que no se muestre el campo dc.subject con valor CDU::
Si el valor de algún metadato es un guión (-) que no lo muestre.
El dc.date es único y sólo se muestra el cualificador issued.
Se cumple que sólo haya un dc.identifier y un dc.format.

1.7. Justo después hay que poner el siguiente bloque de instrucciones para obligar a que se cumpla el orden de elementos *ESE* establecido según http://pro.europeana.eu/documents/10128/562455/About+the+ESE+3.4+Schema.pdf:

```
StringBuffer europeanaObject = new StringBuffer();
StringBuffer europeanaProvider = new StringBuffer();
StringBuffer europeanaType = new StringBuffer();
StringBuffer europeanaRights = new StringBuffer();
StringBuffer europeanaDataProvider = new StringBuffer();

europeanaObject.append("<europeana:").append("object").append(">").append(thumbnail_
url).append("</europeana:").append("object").append(">");
metadata.append(europeanaObject.toString());

europeanaProvider.append("<europeana:").append("provider").append(">").append(Config
urationManager.getProperty("ese.provider")).append("</europeana:").append("provider").
append(">");

metadata.append(europeanaProvider.toString());

for (int i = 0; i < allDC.length; i++)
{

String e_element = allDC[i].element;
String e_qualifier = allDC[i].qualifier;
String e_language = allDC[i].language;
   String e_value = allDC[i].value;
if(e_element.equals("type"))
{
            if(e_value.equals("Article") || e_value.equals("ArtÃculo") || e_value.equals("Carta") ||
            e_value.equals("Letter") || e_value.equals("Libro") || e_value.equals("Book") ||
  e_value.equals("BoletÃn de noticias") || e_value.equals("News letter")||
  e_value.equals("Carta") || e_value.equals("Letter") || e_value.equals("Discurso") ||
e_value.equals("Lecture") || e_value.equals("Documento de trabajo") ||
e_value.equals("Working paper") || e_value.equals("Expediente Personal") ||
e_value.equals("Personal file") ||
```

```
            e_value.equals("Objeto de aprendizaje") || e_value.equals("Learning Object") ||
e_value.equals("Proyecto fin de carrer") || e_value.equals("Bachelor thesis") ||
        e_value.equals("Tarjeta") || e_value.equals("Card") ||
        e_value.equals("Tarjeta de visita") || e_value.equals("Visiting card") ||
        e_value.equals("Tarjeta postal") || e_value.equals("Postcard") ||
        e_value.equals("Telefonema") || e_value.equals("Telephoned telegram") ||
        e_value.equals("Tesis Doctoral") || e_value.equals("Doctoral Thesis") ||
                e_value.equals("info:eu-repo/semantics/annotation") || e_value.equals("info:eu-
                repo/semantics/article") ||
                e_value.equals("info:eu-repo/semantics/bookPart") || e_value.equals("info:eu-
                repo/semantics/contributionToPeriodical") || e_value.equals("info:eu-
                repo/semantics/workingPaper") || e_value.equals("Ejercicios prÃ¡cticos") ||
                e_value.equals("Examen") || e_value.equals("Folleto") || e_value.equals("info:eu-
                repo/semantics/report") ||
        e_value.equals("info:eu-repo/semantics/book") ||
                e_value.equals("info:eu-repo/semantics/conferenceObject") ||
                e_value.equals("info:eu-repo/semantics/patent") || e_value.equals("info:eu-
                repo/semantics/preprint") || e_value.equals("info:eu-repo/semantics/lecture") ||
                e_value.equals("Programa docente") ||
                e_value.equals("info:eu-repo/semantics/review") || e_value.equals("Software") ||
                                e_value.equals("Tesis de licenciatura") ||
                e_value.equals("info:eu-repo/semantics/masterThesis") ||
                e_value.equals("info:eu-repo/semantics/doctoralThesis") || e_value.equals("info:eu-
                repo/semantics/bachelorThesis") ||
                                        e_value.equals("info:eu-repo/semantics/other"))
                        {
                        europeanaType.append("<europeana:").append("type").append(">").appen
                        d("TEXT").append("</europeana:").append("type").append(">");

                metadata.append(europeanaType.toString());
                                                break;
        }
                                                        if(e_value.equals("Audio") ||
          e_value.equals("Sound"))
                        {
                         europeanaType.append("<europeana:").append("type").append(">").appen
                        d("SOUND").append("</europeana:").append("type").append(">");
                                                metadata.append(europeanaType.toString());
                                                break;

                }

                if(e_value.equals("Imagen") || e_value.equals("Image") ||
                e_value.equals("AnimaciÃ³n: Imagen o video") || e_value.equals("FotografÃa"))
        {
                        europeanaType.append("<europeana:").append("type").append(">").appen
                        d("IMAGE").append("</europeana:").append("type").append(">");
                                                metadata.append(europeanaType.toString());
                                                break;

                }

                if(e_value.equals("VÃdeo") || e_value.equals("Video"))
                        {
                        europeanaType.append("<europeana:").append("type").append(">").appen
                        d("VIDEO").append("</europeana:").append("type").append(">");

metadata.append(europeanaType.toString());
                                                break;
```

```
        }
    else{

                    europeanaType.append("<europeana:").append("type").append(">").appen
                    d("TEXT").append("</europeana:").append("type").append(">");
                                        metadata.append(europeanaType.toString());
                                        break;
                                }
                            }
                                    }
                    europeanaRights.append("<europeana:").append("rights").append(">").app
                    end("http://creativecommons.org/licenses/by-nc-
                    nd/3.0/es/").append("</europeana:").append("rights").append(">");
        metadata.append(europeanaRights.toString());


                    europeanaDataProvider.append("<europeana:").append("dataProvider").ap
                    pend(">").append(ConfigurationManager.getProperty("dspace.name")).app
                    end("</europeana:").append("dataProvider").append(">");
        metadata.append(europeanaDataProvider.toString());
```

1.8 Posteriormente comentamos el siguiente bloque:

```
 //if(europeana_object.toString().compareTo("") == 0 && thumbnail_url != null)
// {
//europeana_object.append("<europeana:object>").append(thumbnail_url).append("</e
//uropeana:object>");
//}
```

Lo hacemos para que no se vuelva a añadir el campo object, correspondiente a las miniaturas generadas, ya que lo hemos añadido en el paso previo.

1.9 Después de los cambios en la clase java, hay que compilar el proyecto con Maven ejecutando el comando:

```
mvn package
```

1.10 En el fichero dspace.cfg añadimos la siguiente línea:

```
ese.provider = Hispana
```

Para que Hispana sea el valor del metadato europeana:provider:

```
<europeana:provider>Hispana</europeana:provider>
```

1.11 Se modifica el archivo [dspace-installation]/config/oaicat.properties añadiendo la siguiente línea:

```
Crosswalks.ese=org.dspace.app.oai.ESECrosswalk
```

1.12 Por último se hace un update y un reinicio del servidor dónde esté alojado DSpace. Para ver los metadatos del repositorio en formato ESE, se introduce la siguiente url:

[dspace-oai-url]?verb=ListRecords&metadataPrefix=ese

Paso 2 - Generación de thumbnails

Para que el metadato europeana:object aparezca es necesario que los ítems del repositorio tengan thumbnails (miniaturas) generadas de sus bitstream. Para ello podemos proceder de dos formas:

2.1 Obtener primera página del documento: necesitamos el software XPDF para generar automáticamente thumbnails de las primeras páginas de los documentos. Primero descargamos e instalamos el "Java Advanced Imaging Image I/O Tools."
http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-java-client-419417.html#jaiio-1.0_01-oth-JPR

```
tar -xvf jai_imageio-1_0_01-lib-linux-i586.gz
mvn install:install-file -Dfile=jai_imageio-1_0_01/lib/jai_imageio.jar -DgroupId=com.sun.media
-DartifactId=jai_imageio -Dversion=1.0_01 - Dpackaging=jar
mvn install:install-file -Dfile=jai-1_1_2_01/lib/jai_core.jar -DgroupId=javax.media -
DartifactId=jai_core -Dversion=1.1.2_01 -Dpackaging=jar -DgeneratePom=true
```

Descargamos el software XPDF de su página oficial:
http://www.foolabs.com/xpdf/download.html

Descomprimimos y movemos a la carpeta definitiva:

```
tar -xvf xpdfbin-linux-3.03.tar.gz
mv xpdfbin-linux-3.03 [dspace.dir]
```

Añadimos las siguientes líneas al dspace.cfg:

```
xpdf.path.pdftotext = [dspace.dir]/xpdfbin-linux-3.03/bin64/pdftotext
xpdf.path.pdftoppm = [dspace.dir]/xpdfbin-linux-3.03/bin64/pdftoppm
xpdf.path.pdfinfo = [dspace.dir]/xpdfbin-linux-3.03/bin64/pdfinfo
```

Las entradas del filter.org.dspace.app.mediafilter.X quedan así en dspace.cfg:

```
filter.org.dspace.app.mediafilter.PDFFilter.inputFormats = Adobe PDF
filter.org.dspace.app.mediafilter.HTMLFilter.inputFormats = HTML, Text
filter.org.dspace.app.mediafilter.WordFilter.inputFormats = Microsoft Word
filter.org.dspace.app.mediafilter.PowerPointFilter.inputFormats = Microsoft Powerpoint, Microsoft
Powerpoint XML
filter.org.dspace.app.mediafilter.JPEGFilter.inputFormats = BMP, GIF, JPEG, image/png
filter.org.dspace.app.mediafilter.BrandedPreviewJPEGFilter.inputFormats = BMP, GIF, JPEG,
image/png
filter.org.dspace.app.mediafilter.XPDF2Thumbnail.inputFormats = Adobe PDF
filter.org.dspace.app.mediafilter.XPDF2Text.inputFormats = Adobe PDF
```

Las del plugin.named.org.dspace.app.mediafilter.FormatFilter quedan en dspace.cfg así:

```
plugin.named.org.dspace.app.mediafilter.FormatFilter = \
org.dspace.app.mediafilter.XPDF2Text = PDF Text Extractor, \
org.dspace.app.mediafilter.XPDF2Thumbnail = PDF Thumbnail, \
```

```
org.dspace.app.mediafilter.HTMLFilter = HTML Text Extractor, \
org.dspace.app.mediafilter.WordFilter = Word Text Extractor, \
org.dspace.app.mediafilter.PowerPointFilter = PowerPoint Text Extractor, \
org.dspace.app.mediafilter.JPEGFilter = JPEG Thumbnail, \
org.dspace.app.mediafilter.BrandedPreviewJPEGFilter = Branded Preview JPEG
```

Y las del filter.plugins en dspace.cfg así:

```
filter.plugins = PDF Text Extractor, HTML Text Extractor, \
                 PowerPoint Text Extractor, \
                 Word Text Extractor, JPEG Thumbnail, PDF Thumbnail
```

Empaquetamos

```
mvn -Pxpdf-mediafilter-support package
ant update
```

Para generar los thumbnails es necesario ejecutar el siguiente script

[dspce]/bin/filter-media

Es recomendable ejecutarlo diariamente con el fin de que se generen todos los thumbnails de cada nuevo registro que se añaden al repositorio. Para ello puede colocarse la tarea en el cron del sistema

```
# Run the media filter at 02:00 every day
0 2 * * *  [dspace]/bin/filter-media
```

2.2 Cualquier página del documento: crear un thumbnail de cualquier página o pdf del documento mediante un programa de edición gráfica. Esa imagen se subiría al repositorio bien mediante ingesta masiva a través de itemimport, o se puede cargar la imagen a través de la interfaz XMLUI desde la parte de administración.