

EAD: Aplicaciones prácticas.

Dr. EDUARDO PEIS
Dr. ANTONIO A. RUIZ-RODRÍGUEZ
DAVID GÓMEZ DOMÍNGUEZ

U. de Granada- Facultad de Biblioteconomía y Documentación

RESUMEN:

La "sociedad de la información" ha obligado a los archivos a hacer un esfuerzo en lo relativo a normas descriptivas. Por otra parte, la "sociedad digital" ha provocado que los archiveros reformulen la manera de preparar y presentar la información de sus archivos. Era necesario codificar electrónicamente los instrumentos de descripción de archivo. De esta forma surgió el modelo EAD (Encoded Archival Description), una DTD (Document Type Definition), en un principio SGML (Standard Generalized Markup Language) y ahora XML (eXtensible Markup Language), basada en normas previas, independiente de plataformas informáticas y que emplea poderosas herramientas para buscar, recuperar, mostrar y navegar instrumentos de descripción.

1.- INTRODUCCIÓN.

Incluso antes de obtener resultados del esfuerzo normalizador internacional que supuso la ISAD (G), algunos investigadores apreciaron la necesidad de "explorar" la información del archivo en el contexto Web. En esto, como en casi todo, se podía haber seguido los pasos de los esfuerzos de adaptación de servicios de información como las bibliotecas. Sin embargo, debido precisamente a las especiales características del material de archivo, en este caso concreto el desarrollo de un modelo descriptivo codificado supuso un paso adelante sin precedentes en el contexto archivístico. Por otra parte, los continuos avances tecnológicos y su aplicación al tratamiento y gestión de la información y los documentos han modificado los comportamientos de los usuarios de la información. Estos nuevos comportamientos han de provocar que los archiveros reformulen la manera de preparar y presentar la información de sus archivos, procurando rentabilizar los escasos recursos. Es necesario, por tanto, enfocar la atención sobre las normas de la tecnología de la información que influyen y gobiernan estos nuevos usos de la información.

La ISAD (G) ha supuesto un "paso de gigante" hacia la

necesaria búsqueda de rentabilidad en la explotación de los recursos archivísticos, ya que permite compartir, a nivel internacional, una misma estructuración descriptiva, lo que posibilita el intercambio. Sin embargo, el contenido de los elementos descriptivos que componen dicha estructura no está normalizado (tan sólo alguno de ellos), los productos que obtenemos de su aplicación (instrumentos de descripción) no son homogéneos y sus posibilidades de desarrollo informatizado no aseguran la uniformidad en cuanto a la recuperación y el acceso.

En efecto, es modelo fruto del consenso internacional que procura la normalización de la estructura de datos. Es decir, qué información descriptiva debe aparecer en cada caso, y cómo debe aparecer, es decir, cómo se debe estructurar dicha información. De hecho, es un modelo generalizado en su concepción global, pero poco transferible en su aplicación particular, que además no resuelve muchas de las necesidades percibidas: presentar de forma extensiva e interrelacionada la información descriptiva contenida normalmente en los instrumentos de descripción, preservar las relaciones jerárquicas que existen entre niveles de descripción, representar información descriptiva que es "heredada" de un nivel jerárquico a otro, "navegar" en una arquitectura de información jerárquica y realizar indización y recuperación de elementos específicos.

Estas necesidades se podían satisfacer con éxito creando una versión "digital" de dichos instrumentos de descripción, codificando la información descriptiva estructurada con un lenguaje de "etiquetado" normalizado internacionalmente. Además, dicha versión "digital" presentaría las mismas posibilidades de explotación que cualquier recurso electrónico en la Web. Dicha codificación fue emprendida utilizando un metalenguaje estándar, de carácter descriptivo y con enormes posibilidades representativas y funcionales, como SGML.

Efectivamente, el primer intento de aplicación de una codificación normalizada a los instrumentos de descripción archivística iniciado por la Universidad de California en

Berkeley y dirigido por Daniel Pitti (1999), seleccionó SGML (Standard Generalized Markup Language) (ISO, 1986) como técnica ideal para llevar a cabo dicha codificación. El producto desarrollado fue EAD, que en su versión 2002 emplea XML (eXtensible Markup Language) (W3C-XML, 2004) para su representación.

Para conocer las posibilidades de EAD, por lo tanto, es necesario abordar mínimamente las características de los metalenguajes empleados para su diseño.

2.- METALENGUAJES

SGML (ISO, 1986) es un metalenguaje, es decir un medio de describir formalmente un lenguaje, en este caso, un lenguaje de codificación etiquetado. Es un sistema "descriptivo" que se sirve de códigos que simplemente ofrecen nombres para categorizar e identificar partes de un documento. Esto significa que SGML es una norma elaborada para expresar estructuras de contenido en lugar de apariencia de documentos. Es decir, usa códigos de marcaje (etiquetas) que proporcionan nombres para categorizar las partes de un documento. Con metalenguajes de etiquetado descriptivo, como SGML, se diferencia claramente entre contenido y presentación, esto permite que el mismo documento pueda ser procesado fácilmente con muchos tipos diferentes de software, cada uno de los cuales puede aplicar diferentes instrucciones de procesamiento.

Todo ello es posible porque SGML introduce la noción de "tipo de documento", es decir una document type definition (DTD). La DTD es la herramienta para "validar" un documento SGML. En la DTD, el tipo de un documento es definido formalmente por sus partes constituyentes y su estructura. Es el "libro de instrucciones", que permite definir libremente vocabularios de etiquetas que podrán ser procesados sobre cualquier tipo de plataforma e implica que diferentes documentos del mismo tipo pueden ser procesados de una manera uniforme. Por otra parte, SGML proporciona un mecanismo de aplicación general para la sustitución de cadenas (string substitution), que es una forma simple de asegurar la independencia de sistemas concretos.

Efectivamente, la norma no determina conjuntos de códigos que se deban emplear para etiquetar estructuralmente un texto. Lo que hace es proporcionar las herramientas que permiten diseñar las instrucciones para que una máquina "entienda" un documento por su definición estructural.

El documento concreto, que se denomina "modelo de documento" será un texto (en nuestro caso un instrumento de descripción) etiquetado. Las etiquetas utilizan delimitadores para ser distinguidas del texto en sí. Los delimitadores son "<" (inicio de etiqueta) y ">" (final de etiqueta). Normalmente una etiqueta inicio indicará el comienzo de un elemento y una etiqueta final (añadiendo una barra inclinada al inicio de etiqueta "</") determinará

su terminación (decimos normalmente ya que el final de un elemento puede venir indicado por el inicio del siguiente, en cuyo caso se pueden aplicar reglas de minimización).

Las etiquetas indican la aparición de un elemento. Elemento (*element*) es la unidad estructural simple SGML (existen, también, atributos —que normalmente son cualificadores del contenido de elementos—; y entidades —partes nominadas de un documento marcado, independiente de cualquier consideración estructural—). Cada elemento deberá ser definido (declarado) atendiendo a su contenido estructural.

La declaración de elemento especificará el nombre (identificador genérico) de dicho elemento y su modelo de contenido (es decir qué elementos lo pueden constituir). La DTD presentará jerárquicamente las declaraciones de todos los elementos (junto a las declaraciones de entidades y de listas de atributos) que puede contener el tipo de documento en cuestión. En consecuencia, una representación gráfica imaginaria de la DTD sería similar a un árbol genealógico, iniciándose en un ancestro (componente superior) que será la declaración del tipo de documento y descendiendo hasta los elementos más inferiores, cuyo modelo de contenido ya no serán otros elementos, sino cualquier cadena de caracteres.

El uso de SGML constituyó una elección obvia para dominios relacionados con los recursos de información, debido a que posibilitaba un tratamiento informático de los datos adecuado, podía ser fácilmente aplicable a las descripciones, dado que esta información contiene texto y posee una estructura lógica rigurosa (las normas de descripción), adecuada para ser descrita en una DTD y ofrecía una flexibilidad insuperable. Su versatilidad y, al mismo tiempo, sus posibilidades estructurales, lo convierten en el metalenguaje ideal para estructurar, de forma relacional, metadatos (DeRose, 1997).

No es de extrañar, por tanto, que el primer intento de aplicación de una codificación normalizada a los instrumentos de descripción archivística iniciado por la Universidad de California en Berkeley y dirigido por Daniel Pitti (1997), seleccionase SGML como técnica ideal para llevar a cabo dicha codificación.

3.- EAD

El resultado del proyecto, iniciado en 1993 por Pitti (1997) en la Biblioteca de la Universidad de Berkeley, fue el diseño de una DTD que definía una clase de documentos (instrumentos de descripción) que, en términos generales, constaban de una página de título opcional, la descripción del material archivístico y unos apéndices también opcionales. La página de título incluiría variados elementos como la identificación del material o el tipo de instrumento de descripción. La descripción, propiamente dicha, constaría

de una breve descripción del material descrito, una más amplia descripción narrativa y cualesquiera partes segregables (incluyendo elementos etiquetables como título, fechas, alcance y contenido) y una lista de las partes que contendría el conjunto descrito.

El diseño de la versión beta de EAD puso rápidamente de manifiesto algo que, de forma natural, ocurrió en muchas otras aplicaciones SGML. Las ventajas de uso de SGML son enormes, pero debido precisamente a esto, las posibilidades de implementación con la tecnología actualmente operativa son nulas. Era indispensable aprovechar las capacidades de la Web, para ello podría haberse utilizado HTML (que nació como una DTD SGML). No obstante, HTML es un lenguaje de anotación genérico y además híbrido que especifica muy poco sobre la estructura del documento, combina formato y contenido al mismo tiempo, está ligado a una serie de significados y no tiene una estructura arbitraria. La solución pasaba por utilizar XML.

Por todo ello, las últimas versiones de EAD, emplean vocabulario XML, una adaptación de SGML para posibilitar su uso en la Web.

Las principales adaptaciones que presenta XML con respecto a SGML son: para procesar muchos documentos XML no es necesario una DTD (en el caso de XML, Document Type Declaration); los "modelos de documento" pueden incluir "instrucciones de procesado"; en los modelos de contenido de las declaraciones de elementos no se puede emplear el conector "&" (ya que si aparecen varios elementos deben aparecer todos en el orden especificado), no se permite usar reglas de excepción y no se emplean los indicadores de minimización ya que, normalmente, siempre se requiere la aparición de ambas etiquetas; se admite el uso generalizado de elementos "vacíos"; se emplean las denominadas secciones "CDATA" y las *notation declarations*, entre otros nuevos componentes; a menos que se indique lo contrario, los espacios en blanco, cambios de línea o "retornos" son significativos; y, se usa el lenguaje EBNF (Extended Backus-Naur Form) para expresar la sintaxis XML.

La versión 1.0 (de 1998) ya empleaba XML al igual que la última versión de 2002. Tanto la DTD, como la numerosa documentación asociada, es accesible a través del servidor Web oficial EAD (SAA-EAD, 2004), que forma parte del Web de la Library of Congress, siendo respaldado todo ello por la Society of American Archivist.

3.1.- Estructura

De forma general, la EAD es una estructura de datos normalizada que reproduce en formato digital los instrumentos de descripción archivística.

La DTD ha sido diseñada para reflejar la jerarquía natural que presenta la organización de los fondos, en conjunción con la jerarquía intelectual que imponen los archiveros con

sus prácticas descriptivas. En esencia, EAD determina los tipos de elementos utilizables, los atributos que éstos pueden tener asociados y especifica el contenido que dichos tipos de elementos pueden incluir.

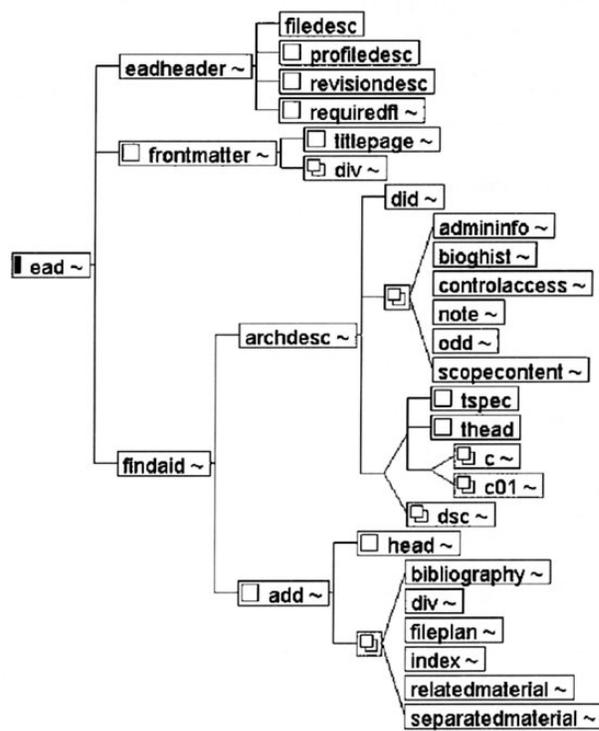
Contiene varios tipos de elementos: aquellos que codifican puntos específicos en la descripción de partes componentes del instrumento de descripción o el material que describe (elementos descriptivos - título de la unidad, fecha de la unidad, productor, etc.-); los utilizados para el acceso (nombre de entidad

-corpname-, nombre de persona, etc.); de enlace y aquellos que podrían codificar cualquier característica del documento (elementos genéricos).-

La versión 1.0 de EAD establece un número de 145 elementos, estableciendo para cada uno de ellos un nombre completo y un identificador genérico. Por ejemplo, para el "título de la unidad" el nombre completo que contempla la norma es "title of the unit" y su identificador genérico "unititle".

La especificación del contenido de cada elemento (modelo de contenido en la DTD) significa aclarar si es de uso obligatorio u opcional, si es repetible o no, si incluye componentes que deben presentarse en una secuencia concreta, etc.

Los principales componentes informativos que constituyen el modelo se pueden representar mediante una estructura arbórea que refleja claramente las relaciones jerárquicas entre los elementos:



EAD usa el término instrumento de descripción (finding aid) para referirse a cualquier herramienta jerárquica que haya sido codificada usando EAD y que permitirá a un creador de registros o a un usuario acceder a los materiales que están siendo descritos.

A un nivel muy básico, un documento "instrumento de descripción" codificado utilizando EAD, consta de tres segmentos: uno que proporciona información sobre el instrumento de descripción en sí mismo (su título, compilador, fecha de compilación), <eadheader>; un segundo componente que incluye las cuestiones preliminares necesarias para la publicación formal del instrumento de descripción, <frontmatter>; y un tercero que proporciona la descripción del material archivístico en sí misma, además de la información contextual y administrativa asociada, <findaid>.

El <eadheader> abarca cuatro subelementos (<filedesc>, <profiledesc>, <revisiondesc> y <requiredft>) para capturar o contener la mayoría de la información que normalmente es registrada sobre la creación, publicación y la distribución de un documento "instrumento de descripción". Otra información adicional sobre el instrumento de descripción puede ser recogida en el elemento <frontmatter> que, incluida en dos elementos como <titlepage> y <div>, refleja detalles de tipo introductorio (prefacio, introducción, etc.) necesarios, como ya hemos indicado, para la publicación formal del instrumento de descripción.

En el elemento <findaid> pueden presentarse dos tipos de información, información de tipo descriptivo e información complementaria, de tipo bibliográfico y administrativo. El elemento "descripción archivística" o <archdesc> presenta información jerárquicamente organizada que describe una unidad de registros junto con sus partes componentes o divisiones. Muchos elementos EAD pueden contener un elemento <head>, para incluir información de "entrada", <archdesc>, puede presentar, además, el elemento <runner>, que permite incluir una cabecera para el conjunto de los datos descriptivos. Estos datos descriptivos se reúnen en un elemento contenedor (<did>) llamado identificación descriptiva.

EAD permite incluir, también, información suplementaria opcional que no describe directamente los registros pero facilita su uso por parte de los investigadores, por ejemplo, una bibliografía (en la versión 1.0 los elementos que estructuraban esta información eran agrupados en el elemento "complemento a los datos descriptivos" o <add>); e información de descriptiva pero de tipo "administrativo", como detalles sobre el ingreso, datos de valoración o condiciones de acceso (información representada por elementos agrupados en <admininfo> de la versión 1.0).

Los descriptores (de nombre, de familia, geográficos, de materia, etc.) pueden ser representados en EAD con los elementos <name>, <famname>, <geogname>, <subject>, etc. agrupados en el genérico <controlaccess> (Controlled Access Headings).

Para enlazar un instrumento de descripción a copias digitalizadas de los materiales descritos o a cualquier otro recurso, EAD proporciona el elemento <dao> (Digital Archival Object), que lógicamente facilita la navegabilidad. Los mecanismos de enlace EAD son compatibles con XML Linking and Addressing Language, pero no se ajustan a la especificación. La localización del dao se especifica mediante <daoloc>, y ambos pueden estar vacíos o contener información que describa el dao contenida en el elemento <daodesc>. Para otro tipo de enlaces, como el que conectara con otro instrumento de descripción se emplea el elemento <extref> (extended reference), que estaría sujeto a condicionantes semejantes a los de dao. El componente dao es subelemento de did, como los más importantes elementos de tipo descriptivo como identificación de la unidad (<unitid>), título de la unidad (<unititle>), fecha (<unitdate>), descripción física (<physdesc>), etc.

Los subelementos de did, naturalmente pueden presentar atributos que cualifican su contenido como label, type o encodinganalog. Este último atributo permite relacionar cada elemento descriptivo con el correspondiente elemento ISAD o campo MARC, lo que facilita la generación de registros ISAD o MARC desde archivos EAD y potencia la accesibilidad, además de mejorar las posibilidades de indización. De cualquier forma, en la edición EAD 2002, se explicita claramente la compatibilidad entre la codificación EAD y el modelo normativo ISAD (de hecho, en las directrices de aplicación de la versión 1.0 ya se recogía un listado de correspondencias entre elementos EAD y elementos ISAD). En un instrumento de descripción codificado esta compatibilidad se puede poner de manifiesto expresamente mediante el atributo relatedencoding, en el elemento raíz ead, especificando en cada elemento descriptivo el elemento ISAD correspondiente mediante el citado atributo encodinganalog:

```
<ead
  relatedencoding="isad">
  <!--
  .....-
  >
  <dsc type="in-depth">
  <c level="item" encodinganalog="3.1.4">
  <did>
  <unitid encodinganalog="3.1.1"
    countrycode="sp">
```

```

repositorycode="ACT.GR">ES.GR.Archivo
Casa de los Tiros. C3, P15.</unitid>
<unittitle encodinganalog="3.1.2">El
Principe de los Gitanos</unittitle>
<!--
.....-
>

```

Como ya hemos comentado, los atributos son una forma de incorporar características a los elementos, tienen su propio nombre y pueden tomar varios valores. Los atributos en su mayoría son opcionales, aunque los hay obligatorios como el atributo "level" del elemento "descripción archivística":

```
<archdesc level="item"> ... </archdesc>
```

Por otra parte, la EAD (Encoded Archival Description) ha abierto nuevas posibilidades a la representación de la vinculación entre los diferentes niveles de descripción archivística. Presenta un elemento, "descripción de componentes subordinados" <dsc> (en "archdesc") que permite incluir información sobre las unidades de descripción inferiores dependientes, jerárquicamente, de la unidad descrita.

Este elemento se utiliza de forma combinada con el elemento "componente" <c> que proporciona información particular sobre las unidades de descripción subordinadas y a su vez puede subdividirse en otros componentes tantas veces como así lo requiera la jerarquía de las unidades de descripción (<c01>, <c02>, <c03>...).

Con el objeto de detallar el tipo de nivel, se recoge un atributo de nivel (como hemos visto anteriormente "level" en "archdesc"), que puede tomar nueve valores distintos: "collection", "fonds", "recordgrp", "series", "subgrp", "subseries", "file", "item" y "otherlevel".

Para indicar el tipo de estructura jerárquica subordinada, el elemento <dsc> puede tener un atributo "type" que puede tomar los siguientes valores: "analyticover" (estructura jerárquica que desciende hasta nivel de subserie), "combined" (desciende hasta la indicación del contenido de las series o subseries), "in-depth" (estructura jerárquica con información detallada) y "othertype" (otros modelos).

Es posible agrupar conjuntos de instrumentos de descripción mediante el elemento raíz <eadgrp>, que incluye sus propios elementos como <archdescgrp> o <dscgrp>, por ejemplo.

Resumiendo las diferencias de la versión 2002, con respecto a la versión anterior, hay elementos que se desaconsejan

como <organization> (cuya función la asume <arrangement>) y otros que son nuevos, algunos de los cuales eran atributos, como <legalstatus> o <langmaterial>. Otros cambios estructurales, ya citados, consisten en la reprobación de elementos "contenedor" como <add> o <admininfo>, o la jerarquización de algunos elementos dentro de otros en los cuales no se podían imbricar en la versión anterior.

La versión 2002, también incluye mención expresa a su compatibilidad con ISAD (G), cosa que no sucedía en la versión 1.0 (recordemos que la versión 1.0 de EAD fue publicada antes que la primera versión de ISAD -G).

3.2.- Uso.

Como ya hemos comentado, tanto la DTD EAD (junto a sus archivos complementarios) como documentación de apoyo para su implementación son accesibles a través de la Web oficial EAD.

Los componentes de la DTD se dividen en módulos en el archivo ead.dtd y otros cuatro archivos asociados que funcionan como unidad. Los cinco archivos son documentos texto en formato ASCII y pueden ser editados con el bloc de notas, por ejemplo. Los archivos son los siguientes: ead.dtd (archivo principal que contiene detalles sobre las versiones, las declaraciones de elementos y atributos, además de declaraciones de entidad de sistema que "llaman" al resto de archivos del conjunto); eadbase.ent (declaraciones de entidades SGML para EAD); eadnotat.ent (declaraciones de notación para formatos de archivo de imagen -GIF, JPEG, TIFF y MPEG-); eadchars.ent (datos sobre conjuntos de caracteres); y, eadsgml.dcl (declaraciones de elementos y atributos SGML). Lógicamente, eadsgml.dcl y eadbase.ent no se emplean cuando se trata de un archivo XML.

La Society of American Archivist mantiene páginas de ayuda (SAA-EAD Help pages, 2004) en las que además de FAQs y enlaces a recursos, se referencian algunas herramientas, publicadas por SAA, utilísimas para el uso e implementación EAD, como las *application guidelines* (directrices de aplicación), la *taglibrary* ("biblioteca" de etiquetas) o el *cookbook* (directrices de implementación);

En dichas herramientas (las *application guidelines* y la *taglibrary* han sido editadas en papel, en español, por la Fundación Histórica Tavera) hay información relativa a cada uno de los componentes EAD, variables de uso, recomendaciones para la utilización de atributos, consejos para la distribución y publicación de instrumentos de descripción codificados, datos sobre algunas aplicaciones informáticas adecuadas, tanto comerciales como de uso público, etc.

De cualquier forma, para una mejor comprensión de la

edición de un instrumento de descripción codificado con EAD, vamos a referirnos a un ejemplo manufacturado con el bloc de notas.

La cuestión fundamental a tener en cuenta en cualquier proceso de edición de un documento EAD es una obviedad, y es que se trata de un documento XML (de ahí la pertinencia del apartado anterior dedicado al "metalenguaje").

El documento EAD comenzará, como cualquier archivo XML, con la declaración XML:

```
<?xml version="1.0" standalone="no"
encoding="ISO-8859-1"?>
```

La declaración XML es una instrucción de procesamiento que indica al procesador que lo que viene a continuación es un documento XML, en concreto un documento que se ajusta a la especificación XML en su versión 1.0; que es un documento XML válido (es decir un documento XML *bien formado*, pero que además responde a una DTD), lo que se expresa con el atributo `standalone` con un valor `no`; y

que es un documento XML en el que pueden aparecer caracteres acentuados o la letra ñ, indicado por el atributo `encoding` con un valor `ISO-8859-1`, que corresponde a *Latín1*.

La DTD a la que responde es, lógicamente, la DTD EAD. Naturalmente, todos los modelos de documento (documentos etiquetados) EAD no tienen que ir acompañados explícitamente de la DTD, ésta es referenciada desde el documento mediante lo que se denomina una referencia de entidad de sistema incluida en la declaración de tipo de documento EAD:

```
<?xml version="1.0" standalone="no"
encoding="ISO-8859-1"?>
<!DOCTYPE ead PUBLIC "-//Society of
American Archivists//DTD ead.dtd (Encoded
Archival Description (EAD) Version 1.0)//
EN" "ead.dtd">
```

A continuación, aparecerían los datos descriptivos de cada caso, estructurados según el modelo comentado en el apartado anterior y de acuerdo con la lógica descriptiva archivística:

```
<?xml version="1.0" standalone="no" encoding="ISO-8859-1"?>
<!DOCTYPE ead PUBLIC "-//Society of American Archivists//DTD ead.dtd (Encoded Archival Description
(EAD) Version 1.0)//EN" "ead.dtd">
<ead relatedencoding="isad">
<eadheader>
<titlestmt>
<titleproper>Codificación EAD</titleproper>
<author>Descripción realizada por Luis Cuenca en Enero de 2004</author>
</titlestmt>
</eadheader>
<findaid legalstatus="public">
<archdesc langmaterial="spa" audience="external">
<dsc type="in-depth">
<!-- .....-->
<c06 level="item" encodinganalog="3.1.4">
<did>
<unitid encodinganalog="3.1.1" countrycode="sp" repositorycode="ACT.GR">ES.GR.Archivo Casa de los
Tiros. C3, P15.</unitid>
<unittitle encodinganalog="3.1.2">El Principe de los Gitanos</unittitle>
<unitdate encodinganalog="3.1.3">1880-1885</unitdate>
<physdesc encodinganalog="3.1.5"><extent>Una Fotografía</extent>
<genreform>Colodio P.O.P ("Printing-OutPaper") en papel sobre cartón de 15x18cm en perfecto estado de
conservación</genreform>
</physdesc>
<!-- .....-->
</did>
```

```

</c06>
<!-- .....->
</dsc>
<!-- .....->
</archdesc>
</findaid>
</ead>

```

A pesar de que el ejemplo quiere representar un nivel de descripción "item", EAD está especialmente indicado para codificar descripciones multinivel hasta el nivel descriptivo de serie, aunque ya hemos comentado en el apartado estructural que permite una aplicación real de las reglas de descripción multinivel.

En apartados anteriores hemos comentado que codificando un instrumento de descripción con XML podríamos aprovechar prácticamente todas las posibilidades de acceso y recuperación de información de la tecnología de redes. Para comentar el cómo, es necesario apuntar algunos detalles sobre la implementación.

3.3.- Implementación

Una vez codificados los recursos para su explotación podrían recogerse las descripciones en listados de una página Web, podrían enlazarse dichas descripciones a registros MARC en catálogos en línea, dichos enlaces podrían realizarse sobre registros ISAD en bases de datos distribuidas, podría proporcionarse acceso directo a los contenidos mediante alguna aplicación o podrían desarrollarse combinaciones de las opciones apuntadas (y algunas más). Todas estas posibilidades se fundamentan, sobre todo, en la flexibilidad de XML, y dicha flexibilidad se debe, en parte, a que XML separa claramente contenido de presentación.

Una vez más hemos de recurrir a la obviedad: los detalles de implementación de EAD son los de cualesquier documentos XML válidos. Y, como cualquier documento XML, para su "presentación" es necesario proporcionarle estilo. La forma más eficaz de proporcionar estilo a un documento XML es enlazándolo a una "hoja de estilo". La hoja de estilo lógica para un documento XML es la hoja de estilo XSL (eXtensible Stylesheet Language). XSL es una especificación complementaria a XML especialmente desarrollada para proporcionar estilo a los documentos XML. El proceso de aplicación de una hoja de estilo XSL está dividido en dos subprocesos: un proceso de transformación del documento XML fuente en un documento XML resultado (subproceso que se lleva a cabo con XSLT -eXtensible

Stylesheet Language Transformation-) y el proceso de proporcionar formato, propiamente dicho, al documento XML resultado (mediante XSL-FO—eXtensible Stylesheet Language Formating Object-).

Otra opción posible sería transformar, para su visualización, el documento XML en HTML. XSLT no es sólo el proceso de transformación para proporcionar estilo, es mucho más. Con plantillas XSLT es posible convertir un documento XML en HTML, variar la organización de presentación de los elementos del documento XML fuente, ordenar subelementos y elementos según criterios preestablecidos, determinar qué elementos (y cuáles no) estarán presentes en el archivo de "salida" (con lo que esto puede significar para la implementación práctica de las reglas de descripción multinivel) e incluso "crear" elementos inexistentes en el documento XML fuente.

Por ejemplo, a continuación vemos parte de una hoja de estilo XSLT que enlazada a un documento EAD produciría un estilo de salida que además de la información descriptiva, presentaría una tabla de contenidos en la zona izquierda de la pantalla, también en HTML. Las primeras líneas de código son la declaración XSL y la declaración de los espacios de nombre de XSLT y HTML. Las declaraciones de espacios de nombre han de efectuarse cuando en un documento XML van a aparecer etiquetas pertenecientes a otras "capas" o relativas a otras especificaciones. En este caso, en una plantilla XSL van a aparecer etiquetas XSLT y etiquetas HTML. Tras el "prólogo" (que, naturalmente, comienza con la instrucción de procesamiento XML) aparecen algunos de los elementos fundamentales XSLT, que reflejan, además, el proceso XSLT: posicionamiento (<xsl:template match="/">), selección de nodo (select="ead/eadheader/eadid") y operación de transformación (<xsl:value-of . . . , por ejemplo). Aparece, también, lo que comentábamos anteriormente, la creación de nodos en salida (por ejemplo <xsl:attribute name="content">, para crear un atributo de nombre "content"), además de etiquetas HTML, e incluso expresiones CSS (Cascading StyleSheet).

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns="http://www.w3.org/TR/REC-html40"
result-ns="">

```

```

<xsl:template match="/">
<xsl:variable name="file">
<xsl:value-of select="ead/eadheader/eadid"/>
</xsl:variable>
<html>
<head>
<style> h1, h2, h3 { font-family: arial} </style>
<title><xsl:value-of select="ead/eadheader/filedesc/titlestmt/titleproper"/>
<xsl:text> </xsl:text><xsl:value-of select="ead/eadheader/filedesc/titlestmt/subtitle"/>
</title>
<!-- .....-
<xsl:element name="meta">
<xsl:attribute name="name">dc.title</xsl:attribute>
<xsl:attribute name="content">
<xsl:value-of select="ead/eadheader/filedesc/titlestmt/titleproper"/>
<xsl:text> </xsl:text>
<xsl:value-of select="ead/eadheader/filedesc/titlestmt/subtitle"/>
</xsl:attribute>
</xsl:element>
<xsl:element name="meta">
<xsl:attribute name="name">dc.author</xsl:attribute>
<xsl:attribute name="content">
<xsl:value-of select="ead/archdesc/did/origination"/>
</xsl:attribute>
</xsl:element>
</head>
<body>
<table width="100%" cellpadding="5">
<tr><td width="20%"></td><td width="80%"></td></tr>
<tr><td valign="top" bgcolor="#E0E0E0"><xsl:call-template name="toc"/></td>
<td valign="top" bgcolor="#FAFDD5"><xsl:call-template name="body"/></td></tr></table>
<!-- .....->
</xsl:stylesheet>

```

XSL presenta, además, la ventaja de contar con una especificación propia para proporcionar estilo al documento XML resultado, XSL-FO (XSL-Formating object). En el siguiente ejemplo se proporciona un estilo de fuente Times new roman, con 12 puntos de tamaño de fuente y de color azul a las fechas de la unidad descrita:

```

<xsl:template
match="archdesc[ @level=' collection' ] /did/
unitdate[ @type=' inclusive' ] "
<fo:block color="navy" font-size="12 pt"
font-family="times new roman">
    Dates:
    <xsl:apply-templates/>
</fo:block>
</xsl:template>

```

XSL incorpora características de DSSSL (Document Style Semantics and Specification Language), especificación desarrollada para proporcionar estilo a los documentos SGML, y de CSS. CSS, es, precisamente, otra de las posibilidades para proporcionar estilo a los documentos XML. Las plantillas CSS pueden ser externas o estar contenidas en un archivo HTML. Una Hoja de Estilo externa es un simple archivo de texto con extensión ".css". Este archivo no necesita ninguna declaración inicial ni otra identificación que su propio nombre con extensión y contiene una serie de instrucciones llamadas estamentos. Una regla es un tipo de estamento que identifica un elemento del documento XML y le indica al navegador el estilo que deberá tener ese elemento.

Cada regla consta de: un selector (titleproper, por ejemplo) que identifica (o selecciona) un elemento de la página Web. Al selector le sigue un bloque de declaraciones que comienza con una llave de apertura ({) y termina con otra llave de cierre (}). Entre las llaves van las declaraciones

(display:block;font-size:20pt), que son las que le indican al browser el estilo para el elemento seleccionado. Las declaraciones, a su vez, tienen dos partes: una propiedad (display, font-size) que consiste en alguna de las palabras claves definidas por el lenguaje, seguida de dos puntos y un valor (block, 20pt) para esa propiedad. Las declaraciones de una misma regla están separadas por un punto y coma.

Las hojas de estilo, lógicamente han de ser referenciadas en el documento XML, normalmente tras la declaración XML. Si queremos enlazar el ejemplo EAD anterior a una hoja de estilo CSS (de nombre eadcss3.css, por ejemplo) añadimos una segunda línea de código:

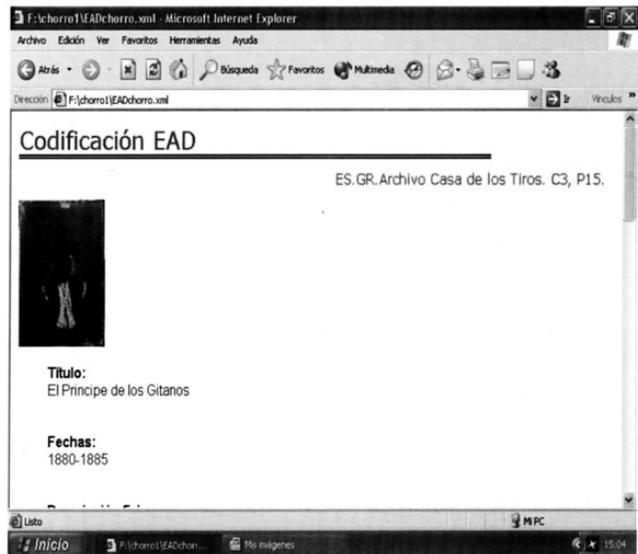
```
<?xml version="1.0" standalone="no"
encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css"
href="eadcss3.css"?>
<!-- .....-->
```

Las reglas de dicha plantilla, podrían ser las siguientes:

```
titleproper {display:block; font-size:20pt;
font-family:Verdana;
color:green; text-align:left; width:600px;
border-bottom:double 4pt navy;}
unitid {display:block; font-size:12pt; font-
family:Verdana; color:red;
text-align:right; margin bottom:10pt;}
unittitle {display:block; margin-left:1cm;
font-size:12pt; font-
family:Arial}
unitdate {display:block; font-size:12pt;
margin-left:1cm; font-
family:Arial; color:black; text-align:left;
bottom:10pt}
physdesc {display:block; font-size:12pt;
margin-left:1cm; font-
family:Arial; color:black; text-align:left;
bottom:10pt}
.....
otherfindaid {display:block; font-size:12pt;
margin-left:1cm; font-
family:Arial; color:black; text-align:left;
bottom:10pt}
```

```
bibliography {display:block; font-size:12pt;
margin-left:1cm; font-
family:Arial; color:black; text-align:left;
bottom:10pt}
head {display:block; font-size:12pt; margin-
left:1cm; font-
family:Arial; color:black; text-align:left;
bottom:10pt}
```

Y el resultado de salida en explorer 6, el siguiente:



Existen otras posibilidades para presentar documentos EAD: emplear plantillas DSSSL, si el documento EAD es SGML, FOSI (Format Output Specification Instance) o, incluso, formatos de estilo propietarios. De cualquier forma, lo más habitual es recurrir a una combinación de las posibilidades apuntadas, referenciando cada una de las "capas" de trabajo mediante su correspondiente espacio de nombre. Obviamente, sobre un mismo documento pueden ser aplicadas multitud de plantillas y una única plantilla podría aplicarse a muchos documentos.

De hecho, y hasta que software mayoritario no lo permita, los documentos EAD han de ser transformados si han de ser explotados a nivel general. En concreto, lo más habitual, como hemos apuntado, es su transformación en archivos HTML. Este proceso de transformación puede ser realizado en modo batch o en el servidor, de acuerdo con los requerimientos del usuario. Para ello, está generalizado el uso de herramientas software de uso público como Xalan, Xason o MSXML. Otra posibilidad sería obtener ficheros imprimibles a partir de archivos EAD, por ejemplo transformándolos en ficheros HTML e importando éstos

desde un procesador de textos, generando un PDF o un RTF mediante "FO processor o, incluso, generando un fichero Word en XML (Microsoft ha creado un XML Schema que representa la sintaxis de los ficheros Word).

Como ya hemos comentado, en las EAD help pages, distribuidas por la Universidad de Virginia, encontramos documentación muy útil para llevar a cabo el proceso de implementación. Entre dicha documentación, destacamos el EAD cookbook (SAA-EAD Cookbook, 2004) una herramienta extraordinaria para la edición e implementación de archivos EAD, incluyendo enlaces para la descarga de ficheros que facilitan dicha tarea, con detalles relativos, incluso, a la utilización de software comercial.

Para ver las posibilidades de aplicación EAD, haremos un breve recorrido por los principales proyectos cooperativos emprendidos para la implementación EAD.

3.4.- Aplicaciones.

El proyecto "A National Database of Electronic Finding Aids for Australian Literacy Manuscript Collections" (<http://www.library.uwa.edu.au/research/rief>) es un proyecto en el que participan seis bibliotecas australianas, bajo la dirección de la University of Western of Australia, con el objetivo de construir una base de datos nacional de instrumentos de descripción (guías electrónicas) codificados con EAD de las colecciones de manuscritos de literatura australiana. El proyecto persigue mejorar el acceso de los investigadores a una información consistente y detallada sobre las colecciones de manuscritos australianos; y se creó para ser el primer uso conjunto significativo del formato EAD en Australia con la intención de actuar como sitio demostrativo de la capacidad de este formato. La base de datos resultante, consta de instrumentos de descripción para más de 80 colecciones de manuscritos.

El "American Heritage Virtual Archive Project" (<http://sunsite.berkeley.edu/amher>) propuso la creación de un prototipo de "archivo virtual" que integrara, en una única fuente, cientos de instrumentos de descripción de archivo que describiesen y proporcionasen acceso a un gran conjunto de colecciones de documentos primarios de las cuatro bibliotecas de investigación científica principales que documentan la historia y cultura americana.

El OAC (Online Archive of California <http://sunsite2.berkeley.edu/oac>) es un proyecto piloto, que se llevó a cabo en dos años, desarrollado por la Universidad de California con la pretensión de realizar un prototipo de base de datos que recogiese instrumentos de descripción de archivos codificados usando EAD. Esta base de datos sería el punto de partida para el desarrollo de un archivo digital a gran escala del Sistema de la Universidad de California disponible en la Web para comunidades de usuarios

diversas.

Con el objetivo de facilitar el desarrollo de esta base de datos, los participantes del proyecto incluyen en la Web del mismo:

- Plantillas, tanto genéricas como personalizadas.
- Las directrices de conversión retrospectiva como suplemento de las Directrices de la SAA.
- Además de otras herramientas e información relacionadas con el proyecto.

El proyecto que ha producido la California Heritage Collection es el California Heritage Digital Image Project (<http://sunsite.berkeley.edu/CalHeritage>). Este archivo proporciona acceso a fotografías, imágenes y manuscritos de las colecciones de la "Bancroft Library" de la Universidad de Berkeley en California a través de instrumentos de descripción codificados con EAD. Todas las imágenes están disponibles al visualizar las descripciones de las diferentes colecciones, y basta con pinchar en cada una de ellas para visualizarla por completo o ver las imágenes asociadas. Todos los instrumentos de descripción se ofrecen como ficheros HTML convertidos desde SGML a través del servidor DynaWeb de Inso.

MALVINE es un proyecto europeo patrocinado por "K.G. Saur Verlag, Munich, Alemania" (<http://www.malvine.org/malvine>) que persigue mejorar y facilitar el acceso a diferentes colecciones de cartas y manuscritos modernos conservados y catalogados en bibliotecas, archivos, centros de documentación y museos europeos. Se trata de crear una red que conecte estas instituciones en Europa y que sea accesible desde cualquier lugar del mundo vía Web como si se tratase de una base de datos homogénea y unificada.

El proyecto proporcionará una interfaz de usuario multilingüe que emplee una terminología común y ofrezca copias en formato digital de documentos valiosos. De esta forma se facilitará tanto el acceso a manuscritos modernos originales europeos como el intercambio de datos entre sistemas heterogéneos. Todo esto va a contribuir a la protección y conservación de los documentos originales.

En definitiva, lo que se pretende es establecer una infraestructura que:

- permita un servicio de búsqueda de manuscritos y petición de copias y
- proporcione a los usuarios información de los distintos catálogos codificados con XML utilizando EAD.

Esto va a posibilitar que cualquier institución, independientemente de su tamaño, ponga a disposición de un amplio

público sus colecciones contribuyendo así con los objetivos de la sociedad de la información.

Los principales resultados de este proyecto son:

- Una herramienta de conversión para traducir los catálogos almacenados en distintos formatos locales a un formato común de intercambio. Este formato es SGML o XML usando la DTD EAD. La herramienta está programada con Perl.
- Un conversor EAD online que intenta convertir cualquier fichero dado a un fichero codificado con XML usando también la DTD EAD. Para que el conversor realice la operación con éxito, el fichero que se intenta convertir debe estar entre los formatos nativos que éste conozca.
- Una herramienta de conversión que demuestra cómo la traducción de los diferentes formatos nativos al formato EAD puede hacerse usando diferentes hojas de estilo (XSL).

Todas estas herramientas están disponibles vía web en la página del proyecto. También se recogen ejemplos de los catálogos de colecciones que actualmente están disponibles en línea. La presentación de los registros de los mismos viene determinada por la hoja de estilo que hayamos elegido dentro de la lista que se nos proporciona.

Se permite la visualización de los registros con cinco hojas de estilo diferentes, aunque por el momento sólo dos se encuentran activadas:

- XML Source Code.
- MALVINE Generic.
- NMD Style (formato usado por el "National Museum of Denmark"- está en construcción).
- ÖND (formato usado por la "Austrian National Library"- está en construcción).
- USMARC (En construcción).

Existen muchos más proyectos y desarrollos accesibles desde el sitio oficial EAD, que ponen de manifiesto las posibilidades de aplicación EAD, su versatilidad y flexibilidad, además de demostrar su utilidad.

4.- CONCLUSIONES

EAD es una DTD XML que refleja la estructura lógica y jerárquica de un instrumento de descripción de archivo, que es compatible con la norma internacional para la descripción de material de archivo (ISAD-G) y que posibilita la difusión, acceso y navegabilidad, a través de la tecnología

de redes, de información descriptiva de archivo.

Los documentos EAD pueden "aprovechar" todas las posibilidades aplicables a los archivos XML. Por ejemplo, los archivos EAD pueden ser enlazables entre sí, enlazadas partes de ellos, o integrar imágenes de los documentos descritos empleando XLink (W3C-XLink, 2004), una especificación que permite crear hiperenlaces multidireccionales, anclados a cualquier elemento y de los cuales es posible controlar su activación y su modo de presentación, entre otras cosas. Sería posible, también, utilizar el DOM (W3C-DOM, 2004), una API (Application Program Interface) orientada a objetos para diseñar una interfaz que permita de forma automática obtener distintas opciones de salida con el mismo conjunto de datos EAD (con lo que esto representaría para la aplicación práctica de las reglas de descripción multinivel).

Por otra parte, junto a la capacidad de XML para codificar información descriptiva, hoy día las posibilidades de integración de las especificaciones metadata en un modelo de gestión de registros electrónicos pasan por RDF (Resource Description Framework). RDF, desarrollado, como XML, por el *World Wide Web Consortium* (W3C-RDF, 2004), es una infraestructura que permite codificar, intercambiar y reusar metadatos estructurados. Esta infraestructura fomenta la interoperatividad de los metadatos por que permite aunar convenciones semánticas y codificación sintáctica con conexiones estructurales bien definidas. RDF no obliga a utilizar una semántica concreta, pero las posibilidades más adecuadas de definir una estructura semántica que refleje las necesidades comunes, las ofrece XML (*eXtensible Markup Language*), ya que puede ser una sintaxis común para el intercambio y procesamiento de los metadatos.

Los profesionales deben conocer EAD y acercarse a XML ya que han de intervenir en el necesario proceso de revisión del modelo.

5.- REFERENCIAS

DeROSE, S. J. (1997). Navigation, access, and control using structured information. *American Archivist*, 60 (3), 298-309

International Council on Archives. (2000). ISAD(G): General International Standard Archival Description. Second Edition. Adopted by the Committee on Descriptive Standards. Stockholm, Sweden, 19-22 September 1999. Ottawa: ICA.

ISO-SGML (1986). *International Organization for Standardization. ISO 8879-1986 (E). Information Processing -Text and Office Systems- Standard Generalized Markup Language (SGML)*. Geneva: International Organization for Standardization.

PITTI, D. (1999). Encoded Archival Description. An introduction and overview. *D-Lib Magazine*, 5 (11).

<http://www.dlib.org/dlib/november99/11pitti.html> (febrero, 2000)

PITTI, D. (1997). Encode Archival Description: the development of an encoding standard for archival finding aids. *American Archivist*, 60 (3), 268-283

SAA-EAD (2004). EAD Cookbook.

<http://www.iath.virginia.edu/ead/cookbookhelp.html> (abril, 2004)

SAA-EAD (2004). EAD Help pages.

<http://jefferson.village.virginia.edu/ead/> (abril, 2004)

SAA-EAD (2004). Official EAD Web Site.

<http://www.loc.gov/ead/> (abril, 2004)

W3C-DOM (2004). *World Wide Web Consortium. Document Object Model Level 3. W3C Recommendation 7 April 2004*

<http://www.w3.org/DOM/> (abril, 2004)

W3C-XLink (2004). *XML Linking Language (XLink) Version 1.0 (2004). W3C Recommendation 27 June 2001*

<http://www.w3.org/TR/2001/REC-xlink-20010627/> (abril, 2004)

W3C-RDF (2004). *World Wide Web Consortium. Resource Description Framework (RDF). W3C Recommendation 10 February 2004*

<http://www.w3.org/RDF/#specs> (abril, 2004)

W3C-XML (2004). *World Wide Web Consortium. Extensible Markup Language (XML) 1.1 W3C Recommendation 04 February 2004*

<http://www.w3.org/TR/2004/REC-xml11-20040204/> (abril, 2004).