

Introducción a Encoded Archival Description (EAD): Mitos y oportunidades

Alejandro Delgado Gómez

Ayuntamiento de Cartagena-Servicio de Archivos y Bibliotecas
Centro Cultural Ramón Alonso Luzzy. C/ Jacinto Benavente, 7. 30203-Cartagena

Tel.: +34 968 128855

E-Mail: alejandro.delgado@ayto-cartagena.es

3000 Informática

Avda. Pío XII, s/n.. 30203-Cartagena

Tel.: +34 968 522330

E-Mail: adgomez@3000info.es

RESUMEN

El presente texto lleva a cabo una introducción al lenguaje de codificación de instrumentos de descripción de archivos Encoded Archival Description (EAD). Puesto que se trata de un lenguaje con cierto grado, este documento propone una aproximación a los retos a que debe enfrentarse a corto plazo. Para ello, se definen los principios que debieran regir un lenguaje de metadatos y, a partir de ellos, se analizan tanto la definición de EAD como sus principios de diseño y su relación con otras normas. Se contrastan las características actuales del lenguaje con aquéllos principios, con el fin de sugerir debilidades y fortalezas. En nota se incluyen todos los ejemplos considerados necesarios a efectos de clarificación. Las conclusiones exponen los valores actuales que EAD debe conservar, así como las nuevas realidades técnicas que debe afrontar.

PALABRAS-CLAVE:

Encoded Archival Description, EAD, Descripción archivística, Metadatos

INTRODUCCIÓN

Como su título indica, el presente texto tiene la finalidad de realizar una introducción al lenguaje *Encoded Archival Description (EAD)* [1]. Por motivos de oportunidad y de tiempo, no pretendemos llevar a cabo un análisis detallado de todas las consideraciones a tener en cuenta al emprender un proyecto EAD, ni un tutorial acerca de su utilización. En el sitio web oficial del lenguaje se encuentran disponibles todos sus documentos canónicos [2] y, por nuestra parte, ya hemos intentado tanto el análisis detallado como el tutorial en otros lugares [3]. Además, creemos que, en el entorno archivístico, EAD se ha convertido en un lenguaje lo suficientemente popular como para repetir el énfasis en conceptos –marcado, metadatos, etiquetas, atributos, etc.- que sólo definimos en nota [4]; de igual modo, incorporamos en el mismo lugar los ejemplos textuales.

Esta introducción se enfoca sobre algunas de las cuestiones que, pensamos, el lenguaje ha dejado pendientes, o que han emergido con posterioridad a los documentos oficiales, dejándolos en algún punto obsoletos. No es nuestra intención, en modo alguno, generar polémica, sino intentar contribuir a la consolidación a corto plazo de un lenguaje ya de por sí

robusto. Para ello, estructuramos el presente texto de la siguiente manera: en primer lugar, definimos los principios y conceptos alrededor de los cuales gira nuestra argumentación. En segundo lugar, definimos el lenguaje que nos ocupa e introducimos algunas matizaciones. En tercero, contrastamos EAD con otros lenguajes de descripción o de codificación. Seguidamente, enunciamos los principios de diseño de EAD y discutimos aquellos aspectos que nos parecen insuficientes o que merecen una actualización. Finalmente, proponemos algunas conclusiones. En anexo explicamos en filigrana nuestro entorno técnico de trabajo con EAD.

ENFOQUE DE LA ARGUMENTACIÓN

Nuestra posición, en términos generales, es de reconocimiento del rigor, estabilidad y robustez de EAD. Sin embargo, creemos que no ha explotado suficientemente algunas de sus características potenciales, explotación que le proporcionaría mayor garantía de futuro.

Como se verá en las siguientes páginas, la mayor parte de nuestras argumentaciones se apoyan en una serie de principios que, habida cuenta de la complejidad creciente del vasto mundo de los metadatos, consideramos fundamentales, a la hora de gestionar de manera útil aquéllos. Estos principios son, en filigrana:

Modularidad: “La modularidad de los metadatos es un principio clave de organización de entornos caracterizados por fuentes de contenido, estilos de gestión de contenidos, y aproximaciones la descripción de recursos ampliamente diversos. Permite a los diseñadores de esquemas de metadatos crear nuevos conjuntos basados en esquemas de metadatos establecidos y beneficiarse de las mejores prácticas observadas, mejor que reinventar elementos de nuevo” [5].

Extensibilidad: “Los sistemas de metadatos deben permitir extensiones para poder acomodar las necesidades particulares de una aplicación dada. Algunos elementos de metadatos es probable que se encuentren en muchos esquemas de metadatos... otros serán específicos de aplicaciones o dominios particulares. Las arquitecturas de metadatos deben

acomodar fácilmente la noción de un esquema base con elementos adicionales que ajusten una aplicación dada a necesidades locales o específicas de dominio sin comprometer indebidamente la interoperabilidad proporcionada por el esquema base. Otra aplicación que encuentre tales extensiones debiera ser capaz de ignorarlas, aun haciendo uso de los elementos comprendidos por ambas” [6].

Refinamiento: “Los dominios de aplicación diferirán según el grado de detalle que es necesario o deseable. El diseño de normas de metadatos debiera permitir a los diseñadores de esquemas elegir un nivel de detalle adecuado a una aplicación dada. Poblar bases de datos con metadatos es costoso, de modo que existen incentivos económicos para crear metadatos con el detalle suficiente para satisfacer los requisitos funcionales de una aplicación, pero no más” [7]. En otros contextos, el refinamiento es similar a lo que se conoce por granularidad o nivel de detalle.

Multilingüismo: “Es esencial adoptar arquitecturas de metadatos que respeten la diversidad lingüística y cultural. La Web, en cuanto sistema de información global, es importante por el hecho de que afronta un acceso sin precedentes a recursos de alcance global. Sin embargo, a menos que tales puedan estar disponibles para los usuarios en sus lenguas nativas, en series de caracteres adecuados, y con metadatos apropiados a la gestión de recursos, la Web no logrará su potencial como sistema de información global. Las normas tratan típicamente estas cuestiones mediante los procesos complementarios de internacionalización y localización: el primer proceso se refiere a la creación de normas “neutrales”, mientras que el último se refiere a la adaptación de tales normas neutrales a un contexto local” [8].

A estos principios fundamentales, añadiremos otros dos conceptos, relacionados de manera más estrecha con los entornos informáticos:

Interoperabilidad, o “la capacidad de los sistemas de información para operar en conjunción unos con otros, adecuando protocolos de comunicación, hardware, software, aplicaciones y capas de compatibilidad de datos” [9].

Reusabilidad, o “el grado en que un módulo de software u otro producto de trabajo puede utilizarse en más de un programa de ordenador o sistema de software” [10].

Incluimos, junto a estos principios y conceptos razonablemente aceptados, otro, que deriva de nuestra experiencia, pero que nos parece igualmente importante: un lenguaje de metadatos será tanto más valioso cuanto más formalizada se encuentre su gramática –su semántica y su sintaxis.

Una vez definidos estos principios, podemos comenzar la evaluación propuesta de EAD.

DEFINICIÓN DE EAD

De acuerdo con la versión 1.0 de EAD, éste es “una serie de reglas para diseñar las partes intelectuales y

físicas de instrumentos de descripción archivísticos, para que la información contenida a este respecto pueda ser buscada, recuperada, visualizada e intercambiada de manera predecible e independiente de la plataforma. Las reglas EAD están escritas en la forma de una Definición de Tipo de Documento (DTD) [11] de *Standard Generalized Markup Language (SGML)* [12], que utiliza representaciones codificadas para un eficiente procesamiento por máquina mediante software de creación y visualización SGML” [13].

En la versión del año 2002 no se modifica esta definición, de modo que debemos considerar que, en cuanto definición, sigue siendo válida. Ello nos permite especificar, con matizaciones, los precisos límites de EAD, puesto que en los últimos años ha existido cierto grado de mitología acerca del status del lenguaje, de su origen, o acerca de qué era o no era capaz de hacer.

En primer lugar, EAD no es una norma, sino una serie de reglas. Esto significa sencillamente que no ha sido aprobado aún por un organismo de normalización, aunque puede ser aprobado en un futuro. Cuestión aparte es que EAD pueda ser considerado, si no una norma *de iure*, al menos una norma *de facto*, especialmente en el contexto norteamericano.

En segundo lugar, EAD es un lenguaje que está basado en la gramática SGML. Aunque en el desarrollo de la DTD definitiva se tomaron las medidas necesarias para asegurar la compatibilidad con *Extensible Markup Language (XML)* [14], -de hecho, tanto en las *Directrices de aplicación* como en la versión 2002 de la DTD, y en un buen número de proyectos, se ha venido minimizando el valor de SGML a favor de XML- existen algunas sustanciales diferencias entre ellos, o, en sentido más estricto, entre una DTD y un schema XML [15], que enunciaremos algo más adelante. De modo que, aunque compatible con XML, EAD es básicamente una DTD para SGML [16].

En tercer lugar, ni la definición ni los documentos acerca de la historia del desarrollo del lenguaje [17] implican que EAD esté basado en ISAD(G) (*International Standard for Archival Description*) [18]. Únicamente en la versión del 2002 [19] se explican los cambios realizados para adaptarse a la versión del año 2000 de la norma internacional. ISAD(G) comenzó a gestarse en 1990 [20], y en años sucesivos conoció diversos borradores, hasta ser aprobada la primera edición en 1993 y publicada en 1994, cuando los responsables del proyecto EAD –que aún no se llamaba EAD- ya habían delineado su método de trabajo [21]. Nuestra impresión es que ambos lenguajes se desarrollaron de manera independiente, aunque casi simultánea, y son compatibles, más por azar que por voluntad de sus respectivos responsables. Volveremos sobre las relaciones entre EAD e ISAD(G) algo más adelante.

En cuarto lugar, EAD no es, en sentido estricto, un lenguaje de descripción de materiales archivísticos – como pueda serlo ISAD(G)-, sino un lenguaje de codificación, y, en esta medida, destinado a ser interpretado por máquinas para generar procesos de búsqueda y recuperación precisos, y para devolver una

visualización legible por humanos. Ello implica, con independencia de que puedan desarrollarse herramientas amigables para el usuario [22], que hoy por hoy EAD requiere el esfuerzo de conocer su código, o, al menos, la parte inmediatamente utilizable por el archivero, es decir, su repertorio de etiquetas y atributos [23].

Por último, EAD es un lenguaje para codificar instrumentos de descripción –en términos convencionales, una guía, un inventario, o una combinación de ambos-, no los materiales archivísticos en sí mismos. Es decir, aunque es posible codificar otro tipo de herramientas, como un catálogo o un cuadro de clasificación, también es posible que existan lenguajes más adecuados que EAD para codificar tales herramientas distintas a los instrumentos de descripción. Además, y aunque esta afirmación no queda explícita en la definición mencionada y responde sólo a nuestra experiencia, EAD es eficaz, por evidentes motivos de probables modificaciones y movimientos, en la codificación de instrumentos de descripción relativos a materiales archivísticos cerrados, cuya vida activa ha expirado; no lo es tanto en el caso de herramientas de codificación relativas a registros abiertos, activos, administrativos, o como se les prefiera llamar. Para ellos se han desarrollado un buen número de lenguajes de metadatos, cuya compatibilidad con EAD valdría la pena examinar. También trataremos de ello en apartado posterior.

EAD EN RELACIÓN CON OTROS LENGUAJES DE DESCRIPCIÓN Y CODIFICACIÓN

El formato MARC

Como es bien sabido, el formato MARC [24] es una estructura de datos, basada en la norma ISO 2709 (*Information and documentation – Format for information exchange*) [25], y de orientación inicialmente bibliográfica. En el contexto norteamericano, y a medida que se hizo evidente la necesidad de informatizar los depósitos de archivos, comenzó a utilizarse una adaptación del formato USMARC: el llamado *USMARC Archives and Manuscripts Control* (MARC AMC). En la actualidad, la descripción archivística en formato MARC se encuentra integrada en el formato MARC21, que sigue teniendo vigencia en cuanto formato de descripción de archivos, en combinación sobre todo con la norma de contenido [26] *Archives, Personal Papers and Manuscripts (APPM2)* [27].

Sin embargo, ya en 1997 Daniel V. Pitti [28], padre de la DTD de EAD, mencionó algunas de las razones que hacían desaconsejable el uso de MARC en la descripción archivística: en primer lugar, los registros MARC tienen una longitud máxima de cien mil caracteres, y las descripciones archivísticas con frecuencia superan este límite; en segundo lugar, MARC acomoda de manera muy pobre la información estructurada jerárquicamente, como lo es la archivística; en tercer lugar, la comunidad de usuarios MARC no había sido capaz de reconducir el estado del arte hacia el desarrollo de hardware y software de bajo coste.

A nuestro juicio, siete años después estas razones han perdido vigencia, básicamente porque los softwares han bajado de coste, al tiempo que han incorporado técnicas

de programación web y de programación orientada a objetos que minimizan, cuando no eliminan, los problemas de la extensión y la jerarquía. No obstante, y siempre según nuestra opinión, el formato MARC ha encontrado nuevos retos que no ha sido capaz de resolver adecuadamente. Básicamente, en los últimos años ha aparecido un volumen ingente de lenguajes de descripción o de codificación especializados en determinadas áreas de conocimiento, y MARC ha pretendido encajar un buen número de ellos en su estructura, ya mediante la concordancia, ya mediante la adición de elementos. Por otra parte, a medida que el proceso de codificación deviene más sofisticado –y en un entorno dominado por redes cada vez más globales, por metadatos cada vez más complejos y por necesidades de recuperación que exigen manejabilidad, la sofisticación nos parece inevitable, siquiera sea a efectos de precisión-, se requieren más códigos, ya no externos, sino derivados de la propia estructura MARC. Por último, existen necesidades locales –en cuanto responsables de procesos de migración y conversión podemos dar fe de que estas necesidades locales son numerosas y por completo imprevisibles-, que también requieren acomodo. Pero el formato MARC, en la medida en que su estructura se define posicionalmente, es limitado, e intentar dar cabida a todas estas nuevas realidades es un propósito que está conduciendo a la saturación y la inestabilidad. Creemos que MARC es un excelente contenedor de datos, por cuanto permite fragmentar de manera coherente la información y, en consecuencia, realizar más procesos y más complejos con ella. Pero también creemos que, si pretende sobrevivir, el formato tendrá, en primer lugar, que cambiar fuertemente de orientación; y, en segundo, reconocer que no ha sido diseñado para “cualquier cosa” y definir sus límites.

EAD también es una estructura de datos, pero, a diferencia de MARC, no se define posicionalmente, de modo que permite una extensión ilimitada de elementos para satisfacer tanto las necesidades de compatibilidad con otros lenguajes, como las derivadas de su propia estructura y las necesidades locales. Sin embargo, debemos tomar precauciones y aprender del ejemplo MARC. En este momento, EAD es manejable, queriendo decir con ello que no está saturado de elementos y es estable. Si queremos que siga manteniendo esta capacidad de maniobra, no debiéramos plantearnos la extensión *ad infinitum* dentro de la propia estructura EAD, ni la modificación permanente de la DTD para cubrir necesidades en constante emergencia. Aunque hablaremos de ello más adelante, hoy en día existen recursos técnicos suficientes para facilitar la interoperabilidad y la reusabilidad de lenguajes, y creemos que el procedimiento correcto para que EAD conserve sus más que razonables niveles de saturación y de estabilidad pasa por explotar esos recursos, minimizando de esta manera el riesgo de trabajar con una DTD en permanente movimiento.

EAD e ISAD(G)

Ante todo, somos de la opinión de que las normas del Consejo Internacional de Archivos ISAD(G) e ISAAR(CPF) (*International Standard Archival*

Authority Record for Corporate Bodies, Persons and Families) [29] poseen un enorme valor estratégico, en la medida en que han conseguido el acuerdo de la comunidad archivística internacional respecto a un asunto tradicionalmente tan falto de orden como la descripción. Tales normas son lo suficientemente conocidas como para entrar a describirlas nuevamente. Sin embargo, puesto que, en líneas generales, se ha asumido que las normas eran buenas sin discusión, e incluso que servían para más de lo que realmente sirven, creemos que conviene extraer algunas conclusiones de las características que la norma ISAD(G) declara acerca de sí misma, con el objeto de poner las cosas en su sitio [30].

En primer lugar, la norma se define como de uso general, y declara explícitamente que debe ser completada por reglas nacionales, o servir de desarrollo a éstas. En la práctica, esta adaptación no se ha llevado a cabo y se utiliza la norma “tal y como está”. EAD también ha sido diseñado para ser utilizado en una amplia diversidad de entornos; pero, a diferencia de ISAD(G), que tiene una estructura básicamente cerrada, EAD es susceptible de modularidad, extensibilidad e interoperatividad. Cuestión diferente es si el lenguaje ha explotado estas posibilidades.

De igual modo, en ISAD(G) la estructura y contenido de la información quedan remitidos a normas nacionales. Esta adaptación aún está oficialmente pendiente, al menos en España y, hasta donde sabemos, también en Portugal; pero, aún más, de la declaración de la norma se implica que no es ninguna estructura informática de datos. Esta estructura informática también debe definirse. EAD, por el contrario, sí es una estructura informática de datos, y está, insistimos, en condiciones de integrarse con otras estructuras informáticas. ISAD(G), por supuesto, puede servir de base para la generación de una estructura tal –ya sea mediante una DTD, dentro de una base de datos relacional, o por cualquier otro método; pero lo cierto es que hasta el momento no se ha definido oficialmente ninguna gramática formal para la norma.

En tercer lugar, la norma incide en la representación del contenido, el contexto y la estructura de la información, aunque, posteriormente, al definir sus elementos, adopta una posición débil al respecto. EAD nació en un momento en el que se estaba razonablemente seguro de que los metadatos en general –y los de archivo de manera muy específica– debían documentar el contenido, contexto y estructura de un objeto de datos [31], y dispone de elementos suficientes para llevar a cabo esta documentación.

En general, somos de la opinión de que ISAD(G), tanto por su carácter de norma marco como por la posición débil adoptada por sus creadores al definir su estructura, está en peores condiciones que EAD –nacida al menos con una estructura más consolidada y evaluada, y con la intención obvia de ser utilizada en entornos informáticos– para afrontar el reto tecnológico presente y el reto aún mayor que se prevé a corto plazo. Incluso más que EAD, pensamos que ISAD(G) necesita una profunda revisión, un proceso de evaluación más

estricto y, si se demuestra necesario, una tercera versión.

DISCUSIÓN DE LOS PRINCIPIOS DE DISEÑO EAD

El Grupo de Trabajo EAD publicó, como parte de la documentación oficial, unos principios generales de diseño [32], que han de mejorar la utilización de EAD y que debieran utilizarse en combinación con la versión 2.0. Nos parece que puede resultar esclarecedor re-examinar estos principios, a la luz de las nuevas realidades ya mencionadas.

En primer lugar, puesto que EAD pretende adecuarse a una gran diversidad de realidades archivísticas a nivel internacional, debe incorporar elementos de carácter general, y también elementos muy específicos. En cuanto a los elementos de carácter general, es incuestionable que los responsables del lenguaje EAD han conseguido su objetivo. En cuanto a los elementos específicos, o muy dependientes del contexto, resulta más cuestionable. Tengamos en cuenta que, aunque los archivos responden, en términos generales, a una estructura común, sus contenidos, por utilizar una expresión algo exagerada, son tan diversos como archivos hay en el mundo. Esto no significa que EAD no sea utilizable en esta diversidad de contextos; pero, como se adelantó, somos de la opinión de que, en su estado actual, EAD ha alcanzado un sólido grado de estabilidad que correría peligro si se pretendiera tratar con exhaustividad toda la gama de contenidos de archivo existente a lo largo y ancho del planeta. Creemos que, lejos de intentar ser exhaustivo, EAD debiera orientar sus esfuerzos hacia la mejora de su modularidad y de su extensibilidad, con el fin de facilitar su uso combinado y coherente con otros lenguajes de marcado y de metadatos de carácter más específico.

En segundo lugar, aunque los responsables de EAD han manifestado su voluntad de reducir el giro anglocéntrico del lenguaje, lo cierto es que la DTD está escrita en inglés, y los nombres de etiquetas y atributos EAD tienen su raíz en la lengua inglesa. Sin embargo, no creemos que esto deba ser preocupante. Como dijimos, EAD no es un lenguaje de descripción, sino de codificación, y el usuario debe hacer el esfuerzo de conocer el código que utiliza. Después de todo, el formato MARC, compuesto por miles de elementos, se expresa mediante códigos numéricos y alfabéticos, y hemos llegado a conocerlo. Otro tanto sucede, por ejemplo, con las etiquetas HTML. Somos de la opinión de que, en lo que se refiere al multilingüismo, los responsables de EAD debieran orientar su esfuerzo en un doble sentido. En primer lugar, promover el desarrollo de aplicaciones amigables para el usuario de cualquier lengua [33]. Téngase en cuenta que EAD es pariente de otros lenguajes planos, como SGML y XML, que a su vez tienen otros muchos hijos. En estas condiciones, es poco probable que a una compañía le interese desarrollar un software para trabajar exclusivamente con EAD, pero sí podemos exigir a nuestro software de gestión de archivos que admita EAD, o cualquier otro lenguaje que consideremos necesario en el archivo [34]. En segundo lugar, facilitar la traducción y la generación de documentación y

herramientas de ayuda para el usuario. En España, la *Biblioteca de etiquetas* y las *Directrices de aplicación* han sido excelentemente traducidas [35]. Hasta donde sabemos, no en Portugal. Lamentablemente, existe una enorme cantidad de documentación complementaria, y a veces no tan complementaria, escrita en inglés, con la evidente limitación que ello supone [36].

EAD se orienta a la codificación de información sobre recursos archivísticos susceptible de ser compartida públicamente. Es decir, no es un lenguaje de codificación destinado a satisfacer todas las necesidades relativas a todas las funciones archivísticas. Ello habla nuevamente de la obligación de mejorar la modularidad y la extensibilidad del lenguaje, de tal manera que resulte optimizable la interoperabilidad, la reusabilidad y las posibilidades de comunicación coherente con otros lenguajes orientados hacia otras funciones que debe cumplir un archivo. Nos encontramos en un momento en el que los metadatos parecen el único tema de moda y de discusión. Constantemente se crean vocabularios, series, conjuntos de metadatos, en ocasiones duplicados, para funciones más o menos específicas [37]. A nuestro juicio, EAD, en la medida en que se ubica en el bloque más popular de metadatos –los metadatos descriptivos [38]–, se encuentra en una posición inmejorable para poner en orden, establecer compatibilidades y definir reglas de juego –reusabilidad, interoperabilidad, extensibilidad, integridad, etc.– para los distintos tipos de metadatos relacionados con las funciones archivísticas, desde que se recibe o se crea un documento que abre un registro, hasta que se decide su eliminación o su conservación permanente.

EAD es una estructura de datos y no una norma de contenido, puesto que no prescribe cómo deben introducirse los datos que aparecen en un elemento dado. Esta función corresponde a normas de contenido externas nacionales o internacionales. En los casos norteamericano y canadiense sí se dispone de tales normas –la citada APPM y RAD (*Rules for Archival Description*) [39], respectivamente. Australia abandonó oficialmente la tarea de definir normas de contenido [40]. En España no existen normas oficiales de contenido –y, hasta donde sabemos, tampoco en Portugal. A nivel internacional, las recomendaciones de ISAD(G) son claramente insuficientes –la propia norma, como hemos visto, lo dice. En el mejor de los casos, es posible trabajar con traducciones de APPM y RAD; pero estas normas, aun resultando extremadamente valiosas, son adaptaciones explícitas de la descripción bibliográfica, que no siempre se corresponde adecuadamente a la descripción archivística [41]. Especifiquemos que no somos enemigos de, digamos, la “interoperabilidad disciplinar”. De hecho, somos firmes partidarios de la misma: en un entorno global, bibliotecas, archivos, museos, etc., tendrán que alcanzar un mayor grado de acuerdo; pero también creemos que sus objetos de información no son exactamente iguales, tienen peculiaridades y a veces no triviales. Por otra parte, somos conscientes de que la tarea de elaborar normas de contenido no corresponde a los responsables de EAD, y de que se trata, en cualquier caso, de una tarea compleja, en la medida en que una norma nacional de

contenido debe satisfacer muy diversas situaciones. Sin embargo, es un problema al que creemos que las comunidades archivísticas deben dar respuesta. Esta respuesta, a nuestro juicio, debiera elaborarse sobre la base del análisis de los elementos de las normas estructurales, del tipo de información que contiene cada uno de ellos, y ofrecer como resultado prescripciones acerca del modo más lógico de representar esta información. Parece ser la orientación del Proyecto Custard [42], un esfuerzo conjunto de la Society of American Archivists y el Bureau of Canadian Archivists, para establecer un modelo de contenido a partir de las normas APPM y RAD, que dé satisfacción a las normas estructurales MARC y EAD, siendo al mismo tiempo acorde con ISAD(G). Además, somos partidarios de que estas normas de contenido, en la medida en que han de responder al tradicionalmente ambiguo y espinoso asunto de la descripción, debieran encontrarse integradas en los módulos de ayuda de los sistemas de gestión integral de archivos. Es, no obstante, una cuestión que dejamos abierta.

EAD es un lenguaje de comunicación basado en la sintaxis SGML. Aunque se declare que también se basa en la sintaxis XML, lo cierto es que existen diferencias entre “ser compatible” –de acuerdo con el diccionario, “que tiene aptitud o proporción para unirse o concurrir en un mismo lugar o sujeto”, y, en informática, “que puede funcionar directamente con otro dispositivo, aparato o programa”–; y “estar basado en” –“asentar algo sobre una base, fundar, apoyar, hacer que algo descansa sobre otra cosa”. La diferencia reside en el grado de dependencia: mientras que en el primer caso se implica independencia entre los dos lenguajes, en el segundo se implica que uno depende del otro. A efectos de validación y conformidad –y también de interoperabilidad con otros lenguajes– la diferencia no es, ni mucho menos, trivial.

Por otra parte, y sin salir de las recomendaciones de tipo tecnológico, los documentos oficiales de EAD siempre han relativizado el valor del lenguaje en el contexto de las tecnologías de bases de datos relacionales u orientadas a objetos. En efecto, el entorno natural de EAD son las aplicaciones basadas en SGML o XML, y que utilizan texto plano. Pero debemos tener en cuenta al menos tres consideraciones. En primer lugar, EAD está orientado a la codificación de instrumentos de descripción, es decir, correspondientes a ítems archivísticos de niveles superiores, y el mayor volumen de registros archivísticos corresponde a los niveles inferiores –unidades documentales compuestas o simples– [43], que, dependiendo además del tipo de archivo, estarán sujetos a restricciones de acceso, modificaciones posteriores, etc. Por otra parte, gran número de estos registros se encuentran ya almacenados en bases de datos relacionales [44], con limitaciones de maniobra, pero también con mayor tradición en cuanto a estructuración detallada de datos, tratamiento confidencial de éstos, seguimiento de las modificaciones, etc. Finalmente, XML es capaz de trabajar en una amplia diversidad de entornos tecnológicos, incluyendo bases de datos. Podemos, por supuesto, migrar los registros a aplicaciones SGML o XML; o utilizar dos aplicaciones: una de ellas para

codificar instrumentos de descripción de niveles superiores y otra para almacenar los registros de niveles inferiores, definiendo los enlaces pertinentes entre una y otra. Creemos, sin embargo, que el esfuerzo en dinero, personal y equipamiento no merece la pena, de manera especial si se tiene en cuenta que, en el estado actual de las nuevas tecnologías, la incorporación de texto plano, la navegación mediante hipervínculo o el mapeado automático entre lenguajes dentro de un mismo contenedor de datos ya no es un problema, sino un hecho. El escenario que prevemos utilizaría lo mejor de ambas tecnologías, para optimizar el funcionamiento del archivo.

EAD es una estructura de datos, no un lenguaje de presentación de los mismos. Esto significa que la mayor fortaleza de EAD descansa en su capacidad para codificar información de manera estructurada, con el fin de poder realizar procesos detallados sobre ella. Sin embargo, no podemos olvidar que uno de los principales procesos es la recuperación y visualización de la información, con todo lo que ello implica: el modo en que se le presente la información al usuario será capital, a efectos de legibilidad, comodidad, rapidez de lectura, etc. [45] Por ejemplo, una presentación por defecto en XML resultaría a todas luces confusa e incómoda para el usuario final. Esta carencia de EAD, que comparte con ISAD(G), no es en sí misma un problema, en la medida en que EAD acomoda el uso de hojas de estilo –CSS (*Cascading Style Sheets*), XSL (*Extensible Stylesheet Language*) o cualquier otra normalizada [46]. No obstante, los criterios de presentación de instrumentos de descripción archivísticos pueden responder a necesidades específicas, incluso dentro del mismo depósito; es decir, a diferencia, digamos, de la descripción bibliográfica normalizada con presentación ISBD, no existen normas que prescriban cómo debe aparecer un instrumento de descripción en pantalla. Ello implica un esfuerzo adicional: o bien la normalización, dentro de un depósito, de los elementos que deben utilizar los codificadores y de acuerdo con determinada estructura; o bien la necesidad de crear diferentes hojas de estilo para diferentes instrumentos de descripción [47]. No es, ni mucho menos, un esfuerzo insalvable –en realidad, existen numerosos ejemplos de hojas de estilo de libre acceso para instrumentos EAD, aunque no todas las hojas de estilo sirven bajo cualquier circunstancia-, pero en ningún caso debemos creer que por el hecho de haber codificado un instrumento de descripción, el resultado final en pantalla gozará de cualidades de estética, legibilidad, homogeneidad y coherencia, ya interna, ya externa. Creemos que no estaría de más el establecimiento de recomendaciones generales para la visualización de resultados en pantalla, similares, aunque no necesariamente idénticas, a las desarrolladas por otras instituciones relacionadas con el mundo de la información. Por ejemplo, la IFLA sí ha publicado recomendaciones acerca del modo más eficaz de visualizar información en un OPAC [48]. De igual modo, sería conveniente comenzar a tener en cuenta las recomendaciones WAI del W3C, citadas algo más arriba.

EAD se apoya en una DTD. En el momento en que

comenzó a desarrollarse el lenguaje –1993-, existía una masa crítica de usuarios que trabajaban bajo SGML; pero esto ha cambiado desde entonces, y ahora se encuentra en alza –y parece que será el lenguaje dominante a no muy largo plazo- el entorno de trabajo XML [49]. Es cierto que los responsables de EAD previeron la compatibilidad con XML, y que desde XML se puede trabajar contra una DTD. Pero también es cierto que una DTD es menos rica, expresivamente, que un schema XML. Por ejemplo, una DTD, a diferencia de un schema XML, no permite la definición de tipos, no permite la definición de tipos definidos por el usuario, no permite el agrupamiento de atributos o no permite la asignación de espacios de nombre. Básicamente, una DTD se queda corta en un contexto como el actual, en el que se tiende a la interoperabilidad entre lenguajes, a la reusabilidad, a la modularidad, a la extensibilidad y a lo que se ha dado en llamar web semántica [50]. EAD comparte algunas de estas carencias –por ejemplo, no es posible indicar una cardinalidad detallada, ni se ha definido explícitamente un espacio de nombre para el lenguaje. Por otra parte, existen distintos tipos de software, comerciales o libres, capaces de convertir automáticamente una DTD en un schema XML [51]. Por lo que a nuestra experiencia se refiere, tales conversiones no son en líneas generales perfectas, y se precisa de un determinado grado de edición y retoque; no obstante, es indudable que la penosa tarea de definir un schema XML se vuelve mucho menos penosa si gran parte de ese trabajo lo hace una máquina, a partir de una gramática previa.

EAD declara su voluntad de mantenerse estable. De hecho hemos insistido en ello, la estabilidad es una de las mayores virtudes del lenguaje, a diferencia de lo que está sucediendo con el formato MARC. Es el método para mantener dicha estabilidad el que nos parece cuestionable: los responsables de EAD se inclinan por la adición más que por la sustitución de elementos. Esto es correcto, en la medida en que la sustitución puede devenir en un caos técnico que dificulte la compatibilidad y la lectura de instrumentos de descripción elaborados con diferentes versiones del lenguaje. Pero la adición, aun reconociendo que será inevitable la incorporación de elementos si se descubren carencias, tampoco nos parece el mejor medio para evitar la inestabilidad. Éste ha sido el procedimiento utilizado por el formato MARC, y el resultado es un grado de incertidumbre notable. Creemos que, en su estado actual, EAD está bien como está, aunque, en un entorno tecnológico que tiende a la interoperabilidad, a la reusabilidad y al mantenimiento de registros de metadatos para facilitar aquéllas, EAD debiera hacer el esfuerzo de garantizar su interoperabilidad con otros lenguajes, de tal manera que pueda utilizarse en combinación con ellos, de acuerdo con diferentes necesidades, dejando la adición de nuevos elementos para casos extremos.

Mencionemos dos recursos de distinto tipo que contribuyen a facilitar esta reusabilidad e interoperabilidad, y, además, se configuran como herramientas básicas para afrontar el reto de una Web semántica.

El primero de estos recursos es la especificación RDF (*Resource Description Framework*) [52] y especificaciones y recomendaciones asociadas. El concepto y la finalidad de RDF quedan bien expresados en la recomendación que sirve de introducción al resto del cuerpo de especificaciones: “El Resource Description Framework (RDF) es un lenguaje para representar información sobre recursos en la World Wide Web. Está dirigido particularmente a representar metadatos sobre recursos Web, tales como título, autor y fecha de modificación de una página web, información sobre derechos y licencia de un documento web, o el calendario de disponibilidad de un recurso compartido. Sin embargo, generalizando el concepto de un “recurso web”, RDF también puede utilizarse para representar información sobre cosas que pueden identificarse en la Web, incluso aunque no puedan recuperarse directamente en la Web. Algunos ejemplos incluyen información sobre ítems de facilidades de compra disponibles en línea (por ejemplo, información sobre especificaciones, precios y disponibilidad), o la descripción de preferencias de suministro de información de un usuario Web. RDF está dirigido a situaciones en que la información tiene que ser procesada por aplicaciones, más que simplemente desplegadas a la gente. RDF proporciona una estructura común para expresar esta información para que pueda ser intercambiada entre aplicaciones sin pérdida de significado. Dado que es una estructura común, los diseñadores de aplicaciones pueden tener la ventaja de disponer de herramientas de validación y procesamiento comunes. La capacidad para intercambiar información entre diferentes aplicaciones significa que la información puede estar disponible a aplicaciones distintas de aquellas para las que se creó originalmente” [53].

El segundo recurso es la creación y uso generalizado de registros de metadatos. De acuerdo con la primera parte de la norma ISO/IEC 11179, todavía en borrador, un registro de metadatos es “un sistema de información para registrar metadatos” [54]. Algo más explícita resulta la explicación facilitada por el Proyecto CORES: “Tal y como se utiliza en el Proyecto SCHEMAS, el término “registro” se refiere, idealmente, a una base de datos que recoge varios tipos de vocabularios de metadatos, desde sus mantenedores en la Web. En respuesta a demandas, un registro tal debiera proporcionar documentación a nivel de término de definiciones y uso, junto con anotaciones contextuales. En efecto, debiera funcionar como herramienta de indexación para actualizar, reunir y servir de manera dinámica un amplio corpus de asientos de “diccionario” para términos de metadatos. El contexto para tal registro es la noción de Web semántica, donde cualquier persona u organización puede declarar un vocabulario de metadatos y afirmar una relación entre ese vocabulario y cualquier otro vocabulario en la Web” [55]. La idea de disponer de algo así como un diccionario universal de metadatos, con el objeto de facilitar la homogeneidad y la reusabilidad, y evitar duplicados, no es nueva, aunque los procedimientos ideados no siempre han sido exitosos [56]. En cualquier caso, y habida cuenta del nivel de crecimiento en la generación y uso de metadatos, las esperanzas

depositadas en estos registros son muchas, y es de desear que, una vez aprobadas las seis partes que componen la norma ISO/IEC 11179, tales esperanzas se conviertan en realidad.

Sin embargo, en lo que se refiere al lenguaje que nos ocupa, hasta donde sabemos, EAD no ha hecho de momento ningún movimiento oficial para adecuarse a RDF [57], ni para figurar en los principales registros de metadatos.

ALGUNAS CONCLUSIONES

Esperamos que de lo expuesto en las páginas anteriores se implique una argumentación algo más sólida de nuestra afirmación inicial. En cualquier caso, podríamos concluir con las siguientes propuestas:

En el apartado de los pros:

En primer lugar, EAD es un lenguaje de metadatos riguroso y estable, en la medida en que ha sufrido un largo proceso de análisis, definición y chequeo, antes y después de la aparición de su primera versión.

En segundo lugar, el lenguaje, en su estado actual, sirve con eficacia a la finalidad para la que fue creado: la codificación de instrumentos de descripción para su colocación en la Web.

En tercer lugar, los desarrolladores del lenguaje fueron lo suficientemente astutos como para garantizar su compatibilidad con tecnologías de futuro.

Pero, en el apartado de los contras:

En un entorno dominado por el uso creciente de metadatos, EAD se ha acomodado a su status, y no plantea batalla en un terreno que ganaría sin excesiva dificultad, teniendo en cuenta sus potenciales características de modularidad, extensibilidad, interoperabilidad, etc.; características que ha renunciado a explotar.

Además, la compatibilidad con XML ya no garantiza su reusabilidad y estabilidad en el futuro. Creemos que la transformación de la DTD en un schema, no sólo no es difícil, sino que además reforzaría la posición del lenguaje.

Por último, los responsables y/o desarrolladores de EAD no han querido, sabido o podido, exigir de la industria de desarrollo de software la implantación de herramientas amigables que faciliten y extiendan su uso, en un entorno de gestión integral de archivos.

Se trata, en definitiva, de retos de futuro, que el lenguaje, sin pérdida de su rigor o estabilidad, debe afrontar, si no queremos que se convierta en una opción técnica, profesional y geográficamente marginal

APÉNDICE: INCORPORACIÓN DE EAD A UN ENTORNO DE TRABAJO RELACIONAL

El software que utilizamos para trabajar en la creación y/o conversión de instrumentos de descripción es Archivo 3000, una aplicación de gestión integral de archivos, y, por tanto, no orientado exclusivamente a EAD. Sin embargo, entre sus diferentes módulos se ha previsto la posibilidad de generar instancias de documentos EAD, así como cualquier otro tipo de documentos basados en DTDs conformes con XML y en schemas XML; así como mapear estos documentos a

ISAD(G) y MARC. Estos mapeados se llevan a cabo por pares, pueden ser decididos por los responsables del archivo, y toman como base de referencia MARC, por el hecho de que aún es la estructura de contenedor de datos más amplia y la que ofrece más juego a la hora de definir concordancias

En primer lugar, Archivo 3000 permite, desde su módulo de descripción importar cualquier fichero de texto con cualquier extensión, de manera que si ya disponemos de instancias externas de documento codificadas en EAD, podemos incorporarlas directamente a nuestro catálogo.

Sin embargo, está solución, aunque cómoda y eficaz, es la más rústica. También podemos crear un documento base que, a modo de plantilla, facilite al codificador la introducción de información en campos seleccionados. El codificador sólo debe insertar la información entre las etiquetas de apertura y cierre.

Si no existen plantillas, el codificador puede teclear el código haciendo uso del editor de texto asociado a la descripción. Este editor de texto, a su vez, también permite generar plantillas.

También es posible crear una máscara para satisfacer los campos predeterminados de manera amigable, por ejemplo mediante rótulos significativos en lenguaje natural, al codificador. Además, para cada campo se pueden definir y predeterminar cierto número de propiedades.

La característica más llamativa, sin embargo, es que el programa despliega, en el formulario de descripción, todas las etiquetas y atributos de EAD, para que el codificador sólo tenga que seleccionar el de su interés con un clic. Estos elementos se encuentran enlazados, a nivel de contenedor interno, a los correspondientes campos MARC y elementos ISAD, si la concordancia existe.

Con este mecanismo de desplegables de elementos y atributos es posible también generar un perfil o plantilla por defecto, a efectos de coherencia de las codificaciones.

El programa permite igualmente enlazar las descripciones a autoridades y navegar, tanto a través de relaciones horizontales como jerárquicas.

De igual modo, permite reorganizar el orden de los campos, únicamente a efectos de visualización, ya que a nivel interno sigue respetando la estructura EAD.

En lo que se refiere a la edición, el programa permite crear hojas de estilo para la visualización de un solo registro o un conjunto de ellos.

Pero, si por determinadas circunstancias, se precisa que uno de los registros tenga una apariencia diferente, es posible editar el estilo desde el módulo mismo de descripción.

En cualquier caso, y sea cual sea el lenguaje de descripción y/o codificación elegido, las pantallas de visualización mapean automáticamente la información a EAD bajo XML, generado simultáneamente el código fuente.

Mediante alguno de estos procedimientos, o de la combinación de varios, es posible generar un registro que constituya todo un instrumento de descripción EAD, incluyendo niveles subordinados. Sin embargo, el programa almacena mejor registros separados para cada nivel de descripción, creando referencias y enlaces entre

todos ellos.

NOTAS

[1] EAD DTD. Version 2002. Society of American Archivists, 2002. URL:

<http://www.loc.gov/ead/ead2002a.html>. Encoded

Archival Description Tag Library / Prepared and Maintained by the Encoded Archival Description Working Group of the Society of American Archivists and the Network Development and MARC Standards

Office of the Library of Congress. Version 2002. EAD Technical Document No. 2. SAA, 2002. URL:

<http://www.loc.gov/ead/tglib/index.html>. Application

Guidelines for Version 1.0: Encoded Archival Description (EAD): Document Type Definition (DTD).

Version 1.0. Technical Document No. 3. Society of American Archivists : Library of Congress, 1999. URL:

<http://www.loc.gov/ead/ag/aghome.html> (Consulta: 26-3-2004)

[2] Encoded Archival Description (EAD): Official EAD Version 2002 Web Site. URL: <http://www.loc.gov/ead/>

(Consulta: 14-3-2004)

[3] Delgado Gómez, Alejandro: Normalización de la descripción archivística: Introducción a Encoded

Archival Description (EAD). Cartagena: Ayuntamiento: 3000 Informática, 2003. Encoded Archival Description

(EAD): Consideraciones administrativas y técnicas. [CD-ROM] Bogotá: 7º Congreso Nacional de Bibliotecología y 1º de Archivística, 2002. EAD como

norma de descripción archivística. Ob. cit.

[4] Siempre que existe una definición EAD, la utilizamos, aunque algunas de ellas estén sujetas a discusión, dependiendo del contexto. “Lenguaje de marcado”:

una manera formal de anotar un documento o colección de datos digitales para indicar la estructura del documento o fichero de datos y los contenidos de sus elementos. Este marcado también sirve para proporcionar a un ordenador información acerca de cómo procesar y desplegar los documentos marcados.

“Metalenguaje”: Una serie de reglas que describen formalmente la sintaxis de un esquema de marcado. SGML es un ejemplo de metalenguaje, o, en otras palabras, es una serie de reglas para establecer lenguajes de marcado. “Elemento”:

Un componente de la estructura definida por una Definición de Tipo de Documento, identificado en una instancia de documento por marcado descriptivo, usualmente una etiqueta de inicio <...> y una de fin </...>. “Atributo”:

Las propiedades nominales de un elemento, que pueden portar diferentes valores dependiendo del contexto en el que aparecen. Los atributos modifican el significado de los elementos a los que se aplican. En: Application

Guidelines... (Consulta: 25-3-2004)

[5] Duval, Erik, Hodgins, Wayne, Sutton, Stuart, Weibel, Stuart L.: Metadata Principles and

Practicalities. En: *D-Lib Magazine*. Vol. 8 N° 4 (April 2002). URL:

<http://www.dlib.org/dlib/april02/weibel/04weibel.html> (Consulta: 25-3-2004)

[6] Ob. cit.

[7] Ob. cit.

[8] Ob. cit.

[9] Interoperability Clearinghouse: Glossary of Terms. ICH, 2003. URL: <http://www.ichnet.org/glossary.htm>

(Consulta 15-3-2004)

[10] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE, 1990

[11] De acuerdo con el glosario EAD, una DTD está constituida por “las especificaciones y definiciones formales de los elementos estructurales y el marcado a ser utilizados al codificar tipos específicos de documentos en SGML”. (EAD Application Guidelines for Version 1.0...) (Consulta: 25-3-2004). En términos más cotidianos, una DTD vendría a ser algo así como la gramática SGML para tipos específicos de documentos.

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

International Organization for Standardization: ISO 8879:1986 / A1:1988 (E). Information Processing: Text and Office Systems: Standard Generalized Markup Language (SGML), Amendment 1. International Organization for Standardization, 1988.

[13] Encoded Archival Description (EAD): Document Type Definition (DTD). Version 1.0. Society of American Archivists: Library of Congress, 1998. Technical Document No. 2. URL: <http://www.loc.gov/ead/tglib1998/tintro.html>. (Consulta: 14-3-2004)

[14] Extensible Markup Language (XML) 1.0: W3C Recommendation 04 February 2004. 3rd. Ed. World Wide Web Consortium, 2004. URL: <http://www.w3.org/TR/2004/REC-xml-20040204/> (Consulta: 25-3-2004)

[15] “El propósito de un schema XML es definir y describir una clase de documentos XML utilizando componentes del esquema para restringir y documentar el significado, uso y relaciones de sus partes constituyentes: tipos de datos, elementos y su contenido y atributos y sus valores”. En: XML Schema Part 1: Structures: W3C Recommendation 2 May 2001. World Wide Web Consortium, 2001. URL: <http://www.w3.org/TR/xmlschema-1/> (Consulta: 25-3-2004)

[16] Tengamos en cuenta que en 1993, año en el que comenzó el proceso de elaboración de EAD, los lenguajes de marcado no habían alcanzado el grado de desarrollo que hoy conocemos. Por ejemplo, HTML se estaba usando sólo desde 1990 (Berners-Lee, T., Connolly, D.: Hypertext Markup Language - 2.0. The Internet Engineering Task Force, 1995. Request for Comments: 1866. URL: <http://www.ietf.org/rfc/rfc1866.txt>. (Consulta: 14-3-2004)), y su versión oficial 2.0 está fechada en 1995 (Berners-Lee, T., Connolly, D.: Hypertext Markup Language - 2.0. World Wide Web Consortium, 1995. URL: <http://www.w3.org/MarkUp/html-spec/html-pubtext.html> (Consulta: 14-3-2004)). La versión 1.1 de PICS (PICS Label Distribution Label Syntax and Communication Protocols) (Krauskopf, Tim, Miller, Jim, Resnick, Paul, Treese, Win: PICS Label Distribution Label Syntax and Communication Protocols: W3C Recommendation. Version 1.1. World Wide Web consortium, 1996. URL: <http://www.w3.org/TR/REC-PICS-labels.html> (Consulta: 14-3-2004)) –al que se suele considerar el primer lenguaje de marcado específico, al menos

procedente del W3C (World Wide Web Consortium)-, data de 1996. Solamente Text Encoding Initiative (Text Encoding Initiative. TEI Consortium, 2003, rev. URL: <http://www.tei-c.org/> (Consulta: 14-3-2004)), como lenguaje de marcado de orientación temática, surgió con anterioridad a EAD, en 1987.

[17] Barry, Randall K. (rev.): Development of the Encoded Archival Description DTD. Society of American Archivists, 2002. URL: <http://www.loc.gov/ead/eaddev.html> (Consulta: 25-3-2004)

[18] International Standard for Archival Description (ISAD(G)). 2nd. Ed. International Council on Archives, 1999

[19] Encoded Archival Description Tag Library... (Consulta: 14-3-2004)

[20] ISAD(G): General International Standard Archival Description / Adopted by the Ad Hoc Commission on Descriptive Standards. Final ICA approved version. International Council on Archives, 1994. URL: [http://www.mclink.it/personal/MD1431/sito/isaargrp/isa_d\(g\)e.html](http://www.mclink.it/personal/MD1431/sito/isaargrp/isa_d(g)e.html) (Consulta: 14-3-2004)

[21] De acuerdo con lo descrito en el desarrollo de la DTD, los esfuerzos del Berkeley Finding Aid Project (BFAP) –antecedente inmediato de EAD- al menos entre 1993 y 1995, fecha de aparición del primer prototipo- se enfocaron sobre la selección del lenguaje de marcado a utilizar –SGML- y al contraste con MARC. Entre 1995 y 1997 –período que incluye el trabajo del Bentley Group y los acuerdos de Ann Arbor- la línea de trabajo fue básicamente la misma. Únicamente se comenzó a mencionar ISAD(G) en 1997, en el momento en que se solicitaron revisiones y sugerencias, aunque no consta que éstas fueran consideradas, y no es disparatado suponer que, puesto que hasta la versión 2002 no se cita explícitamente la adaptación a ISAD(G), ésta no fuera tenida en cuenta en la primera versión. Véase: Development of the Encoded Archival Description DTD... (Consulta: 14-3-2004)

[22] Aunque volveremos sobre este asunto más tarde, el software de trabajo para EAD responde a diversas necesidades –creación, edición, chequeo, visualización...-, no siempre es amistoso para profesionales no habituados al trabajo con lenguajes de marcado y metadatos, y no se encuentra integrado en aplicaciones de gestión integral de archivos. En la práctica, esto significa que el archivero debe trabajar simultáneamente con más de una herramienta, y que éstas plantean a menudo un manejo difícil. Algunos ejemplos pioneros quizá ayuden a entender este problema. Digamos, honestamente, que el estado del arte ha avanzado, desde que surgieron los ejemplos que proponemos, pero los problemas de base enunciados siguen siendo los mismos.

En primer lugar, la Universidad de Harvard - Search Harvard Finding Aids. URL: <http://findingaids.harvard.edu/dfap/> (Consulta: 23-3-2004)- informa de su proyecto de la siguiente manera (EAD Help Pages: EAD Sites Annotated. URL: <http://jefferson.village.virginia.edu/ead/sitesann.html> - Harvard%20University (Consulta: 25-3-2004)): “En la medida en que el proyecto implica a ocho depósitos de Harvard, el procedimiento utilizado para el marcado en EAD es ligeramente diferente en cada uno de ellos, aunque todos siguen las recomendaciones establecidas

en Harvard. Los instrumentos de descripción ahora en EAD son principalmente aquellos que se crearon utilizando varios paquetes de procesamiento de texto, aunque unos pocos han sido creados en tiempo real utilizando plantillas EAD creadas por el depósito individual. Algunos archivos han utilizado macros para automatizar parte del marcado. Se utilizan dos paquetes de edición SGML: Author/Editor (University Archives, Peabody, Baker) y WordPerfect 7 y 8 (Houghton, Andover-Harvard, Law, Botany). Schlesinger utiliza una plantilla de Word y valida los instrumentos de descripción utilizando WordPerfect.

Por su parte, la Universidad de Stanford - Special Collections : Finding Aids. URL: <http://www-sul.stanford.edu/depts/spc/findaids.html> (Consulta: 23-3-2004)- informa del siguiente procedimiento (EAD Help Pages: EAD Sites Annotated. URL: <http://jefferson.village.virginia.edu/ead/sitesann.html> - Stanford%20University%20Library.%20Deepar (Consulta: 25-3-2004)): “Los inventarios se crean generalmente en MS Word o en bases de datos FileMaker Pro. La información almacenada en los formularios de bases de datos se exporta como texto separado por tabuladores y se marca entonces utilizando macros de MS Word escritos localmente”.

Por último, la Niels Bohr Library - Finding Aids in the Niels Bohr Library. URL: <http://www.aip.org/history/ead/index.html> (Consulta: 23-3-2004)- informa de un proceso algo más complejo (EAD Help Pages: EAD Sites Annotated. URL: <http://jefferson.village.virginia.edu/ead/sitesann.html> - American%20Institute%20of%20Physics.%20N

(Consulta: 25-3-2004)): “El American Institute of Physics (AIP) reúne todos los formatos electrónicos en ASCII y divide el texto en ficheros separados, basándose en la información descriptiva y en la lista de contenedores. La información descriptiva se corta y pega en una plantilla con el software de edición NoteTab Pro. La lista de contenedores se manipula y etiqueta utilizando scripts de Perl y C++, así como rutinas de sustitución. Después de unir las dos partes, el instrumento de descripción se valida utilizando NSGMLS (SP). El SGML se convierte a XML cortando y pegando los encabezamientos XML en el documento. El AIP utiliza como herramienta de procesamiento XT. Para la impresión y el formateo en pdf se utilizan FOP y WordPerfect”.

[23] Precisemos que existen algunas herramientas, si no numerosas, al menos muy amigables para el usuario: las diferentes plantillas en Word, las desarrolladas en Java por la Bancroft Library (<http://sunsite.berkeley.edu/cgi-bin/template/CUBanc-extended.tmp>), o los javascripts de los Archivos de la Universidad de Notre Dame (<http://classic.archives.nd.edu/ead/ead.htm>)

[24] Library of Congress. Network Development and MARC Standards Office: MARC Standards. URL: <http://lcweb.loc.gov/marc/> (Consulta: 14-3-2004)

[25] ISO 2709: Information and documentation – Format for information exchange. International Organization for Standardization, 1996

[26] Existen convencionalmente dos tipos de normas. Las normas estructurales –MARC o EAD- indican de qué elementos se compone un lenguaje, en qué orden deben aparecer, cuál es su sintaxis, pero no cómo debe introducirse la información en esos elementos. Esa es

tarea de las normas de contenido –APPM o RAD. Aunque en otro lugar hemos indicado que ISAD(G) es una norma estructural, ahora tendemos a pensar que es en cierto modo una norma mixta, de carácter marco.

[27] Archives, Personal Papers and Manuscripts: a Cataloging Manual for Archival Repositories, Historical Societies and Manuscripts Libraries. Comp. Steven Hensen. 2nd ed. Society of American Archivists, 1989

[28] Pitti, Daniel V.: Encoded Archival Description: The Development of an Encoding Standard for Archival Finding Aids. En: *The American Archivist*. Vol. 60, n. 3 (Summer 1997). P. 268-283

[29] ISAAR(CPF): International Standard Archival Authority Record for Corporate Bodies, Persons and Families / Prepared by the Ad Hoc Commission on Descriptive Standards. Final ICA approved version. The Secretariat of the ICA Ad Hoc Commission on Descriptive Standards, 1996. Y, en su segunda versión, pendiente en el momento de escribir este texto de aprobación definitiva: ISAAR (CPF): International Standard Archival Authority Record For Corporate Bodies, Persons and Families / Prepared by the Committee on Descriptive Standards. Draft Second Edition. International Council on Archives, 2002.

[30] Puede encontrarse una descripción más detallada de esta argumentación en Delgado Gómez, Alejandro: Normalización... P. 28-30

[31] Véase, a modo de ejemplo, el clásico texto: Gilliland-Swetland, Anne J.: Setting the Stage. En *Introduction to Metadata: Pathways to Digital Information*. URL:

http://www.getty.edu/research/institute/standards/intro/metadata/2_articles/index.html (Consulta: 23-3-2004). O, de manera más oficial, International Standard: ISO 15489-1: Information and Documentation-Records Management. Part 1, General. International Organization for Standardization, 2001

[32] Design Principles for Enhancements to EAD. URL: <http://www.loc.gov/ead/eaddesign.html> (Consulta: 15-4-2004)

[33] En filigrana, la mayor parte de las aplicaciones no son aún lo suficientemente amigables –algunas de ellas ciertamente muy complicadas-, o es preciso combinar varias aplicaciones, o éstas no se encuentran integradas en aplicaciones de gestión integral de archivos. Por supuesto, muchas de ellas sólo pueden adquirirse con interfaces en inglés. Algunos ejemplos de software utilizado para trabajar con EAD explicarán mejor lo que queremos decir, teniendo siempre en cuenta que EAD implica varias tareas –desde la creación o la conversión hasta la publicación del documento. Es posible crear documentos EAD con un simple editor ASCII, como el bloc de notas de Windows; con procesadores de texto, como WordPerfect o Microsoft Word; a partir de bases de datos -InMagic, Oracle, FileMaker Pro, FoxPro o desarrollos *ad hoc*-; disponemos de procedimientos y/o aplicaciones para importar a EAD registros de bases de datos, como las preparadas por Montoya desde MS Access - Montoya, Gabriela A.: Conversion of Microsoft® Access Databases into EAD-encoded Finding Aids. URL:

<http://sunsite.berkeley.edu/FindingAids/uc-ead/tools/database/> (Consulta: 25-3-2004)-, o el tutorial desarrollado por Alvin Pollock para la importación y exportación de EAD a bases de datos relacionales,

utilizando Perl y el API de Microsoft ADO (ActiveX Data Objects) -Pollock, Alvin: EAD and Databases: Perl and ADO on MS Windows. URL: <http://sunsite.berkeley.edu/ead/eaddb/> (Consulta: 23-3-2004)-. También es posible la importación y exportación desde documentos HTML. Existen aplicaciones dedicadas a la creación y edición de documentos SGML/HTML –como Note Tab-, o XML -XMLSpy, de Altova, y XMetal, de Corel, son excelentes ejemplos. Por supuesto, existen herramientas más sofisticadas, y, en esa medida, también de uso más complejo: GNU EMACS -GNU EMACS. URL: <http://www.gnu.org/software/emacs/emacs.html> (Consulta: 25-3-2004)- en modo PSGML -Information about PSGML. URL: http://www.lysator.liu.se/~lenst/about_psgml/ (Consulta: 25-3-2004)- con el parser NSGMLS -NSGMLS: An SGML System Conforming to International Standard ISO 8879 -- Standard Generalized Markup Language. URL: <http://www.jclark.com/sp/nsgmls.htm> (Consulta: 25-3-2004). En cuanto a procedimientos de salida, EAD informa de los siguientes: XT de James Clark (el paquete de herramientas de James Clark -James Clark's Home Page. URL: <http://www.jclark.com/> (Consulta: 23-3-2004)- comprende, además de abundante documentación, hojas de estilo, parsers y XT, que es una implantación Java de XSL Transformations (XSLT)), Claris Home Page, FoxPro, Programa Perl de Alvin Pollock (el paquete de herramientas de Alvin Pollock -EAD Tools and Resources. URL: http://www.lib.ncsu.edu/archives/tech_serv/eadtools.html#scripts (Consulta: 23-3-2004) incluye macros en WordPerfect y Word, plantillas, scripts en Perl y documentación), Scripts locales de conversión, Dyna Text de Inso, Isite del CNIDR -Center for Networked Information Discovery & Retrieval (CNIDR). URL: <http://www.cnidr.org> (Consulta:23-3-2004)-, ya no sostenido, pero especialmente interesante, por cuanto proporciona, de manera abierta, un completo paquete de creación de un servidor y cliente Z39.50, además de herramientas de búsqueda, como Isearch, o de gestión del conocimiento, como Iknow, desarrollada en Java, Procesador de estilo XML de Microsoft, Scripts Perl o CGI, OpenText, Microsoft Internet Explorer. En cuanto a las herramientas de suministro mencionadas son: RLG Archival Resources del Research Libraries Group (RLG), InQuery, Dyna Text y Dyna Web de Inso, Internet Explorer 5.0, Isite del CNIDR, MySQL -un potente distribuidor abierto de bases de datos, con diversos productos, como servidor de bases de datos, y conectores ODBC y JDBC-, OmniMark, Panorama, Perl, CGI, enlaces HTML, etc. En lo que se refiere a las herramientas de indexación, se citan: Atomz, Cheshire II, OpenText-LiveLink, Verity Query Language. En realidad, las herramientas de indexación también se han desarrollado en alto grado, existiendo un buen número de ellas en el mercado, con distintas funcionalidades: Verticrawl, Intermedia Text de Oracle, KPS de Ariadne, etc.

[34] La práctica de exigir como requisito antes de la contratación de un software que éste sea compatible con EAD está apenas comenzando en España: las Universidades de Alicante y Pública de Navarra lo han pedido, pero no conocemos otros casos.

[35] Descripción archivística codificada: repertorio de etiquetas. V. 1.0. Fundación Histórica Tavera, 2000; EAD: Descripción Archivística Codificada: directrices de aplicación /elaboradas por el Encoded Archival Description Working Group de la Society of American Archivists. V. 1.0. Fundación Histórica Tavera, 2000

[36] Sirvan, a modo de ejemplo: Archives and Museum Informatics: Cultural Heritage Informatics Quarterly. Vol. 12, N. 3-4 (1998). URL: <http://www.kluweronline.com/issuetoc.htm/1042-1467+12+3/4+1998> (Consulta: 23-3-2004). The Encoded Archival Description: Retrospective Conversion Guidelines: A Supplement to the EAD Tag Library and EAD Guidelines. American Heritage Virtual Archive Project : University of California Encoded Archival Description Project. RLG Best Practice Guidelines for Encoded Archival Description. RLG EAD Advisory Group. 2002. Fox, Michael: The EAD Cookbook and Related Tools. URL: <http://jefferson.village.virginia.edu/ead/cookbookhelp.html> (Consulta: 25-3-2004). EAD Tools and Resources. URL: http://www.lib.ncsu.edu/archives/tech_serv/eadtools.html (Consulta: 25-3-2004). UC-EAD. URL: <ftp://library.berkeley.edu/pub/sgml/uc-ead/> (Consulta: 23-3-2004). Encoded Archival Description: Rare Book, Manuscript, and Special Collections Library Duke University. URL: <http://scriptorium.lib.duke.edu/findaid/ead/> (Consulta: 25-3-2004). EAD Help Pages: Helper Files. URL: <http://jefferson.village.virginia.edu/ead/> (Consulta: 25-3-2004).

[37] A vuelapluma, mencionemos metadatos para la gestión de registros –MoReq (*Modelo de requisitos para la gestión de registros electrónicos*), el modelo de Pittsburgh, InterPares, VERS (*Victorian Electronic Records Strategy*), etc.-; metadatos para la preservación –los modelos de las Bibliotecas Nacionales de Australia y Nueva Zelanda, *CURL Exemplars in Digital Archives* (CEDARS), la red *Networked European Deposit Library* (NEDLIB), el proyecto conjunto del Online Computer Library Center y el Research Libraries Group PREMIS (*PREservation Metadata: Implementation Strategies*), etc.-; metadatos de gestión –Indecs (*The Indecs Metadata Framework*), los trabajos de la Workflow Management Coalition, etc.-; metadatos técnicos –MOA2 (*The Making of America II*), los derivados de CWM (*Common Warehouse Metamodel Specification*), etc.-; metadatos descriptivos o de descubrimiento de recursos –el ya citado TEI, Dublin Core, AGLS (*Australian Government Locator Service*), GILS (*Global Information Locator Service*), etc.-; incluso metadatos para mantener el control sobre los metadatos, como es el caso de la norma ISO 11179 (*Information Technology-Metadata Registries*).

[38] No creemos que ésta sea una afirmación arriesgada. Es cierto que existen cada vez más lenguajes, vocabularios y series de metadatos, definidos por su función o por su orientación temática. En el ámbito de los archivos –y específicamente en el de los archivos digitales- un modelo funcional y de información bien aceptado, como OASIS (Consultative Committee for Space Data Systems: Reference Model

for an Open Archival Information System (OAIS) : CCSDS 650.0-B-1 Blue Book, CCSDS Secretariat Program Integration Division ; National Aeronautics and Space Administration, 2002), puede darnos algunas pistas. Si atendemos a su modelo de información, encontramos, como mínimo, metadatos de preservación, metadatos técnicos o de empaquetamiento y metadatos descriptivos. Si atendemos a su modelo funcional, cada una de las funciones de archivo genera metadatos (véase para una explicación detallada de esta argumentación: Werf, Titia van der: The Deposit System for Electronic Publications: A Process Model, Nedlib, 2000). Por último, si atendemos a las relaciones del archivo con otros actores, encontraremos metadatos generados por el productor, por los órganos de gestión y por los usuarios. Todos ellos son necesarios para la adecuada gestión del archivo. Sin embargo, al menos en los últimos años, el debate se ha centrado de manera masiva y casi exclusiva en los metadatos descriptivos, o aquéllos que sirven para que el usuario descubra información; y en las conflictivas relaciones entre los creadores de este tipo de metadatos y los grandes buscadores, que ignoran el uso de tales metadatos (no es el lugar adecuado para definir nuestra posición en este polémica, pero puede consultarse, a efectos de referencia, la excelente síntesis: Méndez Rodríguez, Eva María: Metadatos y recuperación de información: estándares, problemas y aplicabilidad en bibliotecas digitales, Trea, 2002). Aunque creemos que gran parte de esta polémica nace de la notoriedad alcanzada por la *Dublin Core Metadata Initiative* (URL: <http://www.dublincore.org/> (Consulta: 25-3-2004), lo cierto es que EAD, encuadrado, en líneas generales, en el mismo bloque de metadatos, puede beneficiarse del hecho más o menos deseado de encontrarse en el “ojo del huracán”.

[39] Rules for Archival Description / prepared under the direction of the Planning Committee on Descriptive Standards. Bureau of Canadian Archivists, 1990

[40] Hurley, Chris: Common Practice Rules HCPR-DEA-2001-01, Monash University, 1998. URL: <http://www.sims.monash.edu.au/research/rcrg/publications/hcprdea200101.htm> (Consulta: 23-3-2004)

[41] La misma descripción bibliográfica vive un momento de transición y cambio –recordemos que la IFLA (*International Federation of Library Associations and Institutions*) ha dado el visto bueno más o menos oficial al modelo FRBR (*Functional Requirements for Bibliographic Records*), que ya era, no una, sino varias realidades de hecho: Cataloguing Section: FRBR Review Group (Functional Requirements for Bibliographic Records). IFLA, 2004. URL: <http://www.ifla.org/VII/s13/wgfrbr/wgfrbr.htm> (Consulta: 15-3-2004). Functional Requirements for Bibliographic Records: Final Report: Approved by the Standing Committee of the IFLA Section on Cataloguing September 1997 / IFLA Study Group on the Functional Requirements for Bibliographic Records. K. G. Saur München, 1998.

UBCIM Publications - New Series Vol 19. URL: <http://www.ifla.org/VII/s13/frbr/frbr.htm> (Consulta: 15-3-2004). Además, los trabajos MARC para llevar a cabo un análisis funcional pueden encontrarse en: MARC and FRBR. URL: <http://www.loc.gov/marc/marc-functional-analysis/frbr.html> (Consulta: 15-3-2004); y

los del Joint Steering Committee for Revision of Anglo-American Cataloguing Rules en: Other documents. URL: <http://www.nlc-bnc.ca/jsc/docs.html#logical> (Consulta: 15-3-2004). El análisis funcional responde básicamente a la necesidad de afrontar las nuevas realidades en el entorno de la catalogación bibliográfica, para que los componentes de un registro bibliográfico sean funcionalmente acordes a las demandas de los usuarios. En la práctica, esto ha significado una cierta “ontologización” de la descripción bibliográfica. Por otra parte, en la última reunión del Committee on Descriptive Standards del International Council on Archives se ha propuesto el desarrollo de un modelo similar para la descripción archivística (Report of the plenary meeting of the International Council on Archives Committee on Descriptive Standards (ICA/CDS): Canberra, Australia, 27-30 October 2003. URL: <http://www.hmc.gov.au/icacds/eng/canberrareport.pdf> (Consulta: 15-3-2004)). Dicho sea de paso, este análisis funcional implicaría la entrada en juego de un nuevo actor –con otros conceptos, otra terminología, otras finalidades, etc.- en el proceso descriptivo.

[42] Statement of Principles for the CUSTARD Project. Society of American Archivists, 2003. URL: <http://www.archivists.org/news/custardproject.asp> (Consulta: 15-3-2004)

[43] Aunque no disponemos de datos estadísticos se trata de una simple cuestión de sentido común, teniendo en cuenta la estructura piramidal de los archivos. Por otra parte, asumimos que los archivos que se encuentran informatizados realizan la descripción a todos los niveles, incluidos los inferiores. Debe tomarse, pues, esta afirmación con las debidas precauciones.

[44] De nuevo, es una afirmación no contrastada empíricamente y que sigue simplemente las líneas de evolución paralelas de la informatización de archivos y el desarrollo de las tecnologías de la información

[45] Para una introducción clásica al tema de la legibilidad, puede consultarse el término en el Diccionario de la edición y de las artes gráficas / dirigido por John Dreyfus y François Richaudeau. Fundación Germán Sánchez Ruipérez, 1990; y dentro de él, también el artículo de François Richaudeau El proceso de lectura. Ob. cit. P. 449-462. Además, en el entorno web, el problema de la legibilidad está adquiriendo cada vez mayor relevancia, como muestran las recomendaciones WAI (Web Content Accessibility Guidelines) del W3C y especificaciones asociadas: Web Content Accessibility Guidelines 1.0: W3C Recommendation 5-May-1999. World Wide Web Consortium, 1999. URL: <http://www.w3.org/TR/WCAG10/wai-pageauth.html> (Consulta: 25-3-2004)

[46] 1) Hojas de estilo en cascada (CSS) (Cascading Style Sheets, level 1: W3C Recommendation 17 Dec 1996, revised 11 Jan 1999. URL: <http://www.w3.org/TR/REC-CSS1> y Cascading Style Sheets, level 2: CSS2 Specification: W3C Recommendation 12-May-1998. URL: <http://www.w3.org/TR/REC-CSS2/> (Consulta: 23-3-2004); las CSS nacieron para permitir a los programadores modificar el estilo por defecto HTML. Básicamente, permiten enunciar, en un fichero de texto, características tales como los márgenes, la sangría, el

tipo de letra, la familia, el grosor o el color. Tienen la ventaja de que se pueden asociar a más de un documento, de tal manera que el estilo de éstos se gestiona en bloque desde un fichero independiente. 2) Lenguaje de estilo extendido (XSL) (Extensible Stylesheet Language (XSL). Version 1.0: W3C Recommendation 15 October 2001. URL: <http://www.w3.org/TR/xsl/> (Consulta: 23-3-2004)): XSL es, en general, la especificación de hojas de estilo para documentos XML. Más potente que las CSS y DSSSL, toma, sin embargo, elementos de ambos lenguajes, y su gestión resulta más compleja. Además, y a diferencia de CSS, no se aplica directamente sobre el árbol de código, sino que crea su propio árbol de salida. Por último, XSL tiene, al igual que DSSSL, capacidades de conversión. 3) *Document Style Semantics and Specification Language (DSSSL) (DSSSL - Document Style Semantics and Specification Language. ISO/IEC 10179:1996* (<http://www.oasis-open.org/cover/dsssl.html>)): DSSSL fue un intento de uniformizar las modificaciones sobre las hojas de estilo realizadas por empresas, y, aunque tiene valor normativo, apenas ha sido utilizado de manera generalizada. 4) Instancia de especificación de formato de salida (FOSI) (Index of /pub/doc/law+ethics/patents/sgml/cals. URL: <http://ftp.cerias.purdue.edu/pub/doc/law+ethics/patents/sgml/cals/> (Consulta: 25-3-2004)): FOSI sólo se ha utilizado en combinación con la CALS DTD (DND CALS DTD. Version 2.1. URL: <http://cals.debbs.ndhq.dnd.ca/cals/english/calsdtd21.jsp> (Consulta: 25-3-2004)), para las actividades del Departamento de Defensa de los Estados Unidos. En la actualidad existen muchas aplicaciones de creación de hojas de estilo, por lo que el proceso de dotar de estética a los documentos ha devenido mucho más simple. NoteTab, por ejemplo, incluye funcionalidades para la edición de CSS, incluyendo enlaces a TopStyle, quizá la aplicación más conocida para la creación de hojas de estilo para HTML. Por su parte, XMLSpy incluye un editor asociado de hojas de estilo XSL: Stylesheet Designer.

[47] Algunas muestras de formularios de consulta y recuperación son: 1) Durham University Library: se le ofrecen al usuario listados de las colecciones codificadas en EAD. Localmente, la distribución se realiza a través de Dynatext, y en la web mediante Dynaweb. También se distribuyen los documentos en formato postscript, y se ha planeado su futura distribución en pdf. 2) Mandeville Special Collections Library: también la Biblioteca Mandeville ofrece un listado de colecciones; pero incorpora alternativamente un breve formulario de búsqueda. La base de datos de instrumentos de descripción suministra éstos en tres modos: EAD, HTML y ASCII. Las tres versiones se almacenan en un solo servidor. 3) Niels Bohr Library: la Niels Bohr Library sí plantea la recuperación de documentos a través de un formulario más detallado. Los instrumentos de descripción se convierten estáticamente desde XML/EAD a XHTML 1.0, con metadatos RDF, vía XSLT. Dentro de esta estructura, los documentos individuales se escriben y validan en HTML, y descansan en CSS. Las instancias XML se indexan utilizando el motor de búsqueda Verity. Los usuarios pueden formar búsquedas booleanas y por

proximidad, en elementos específicos de los instrumentos de descripción, mediante una interfaz guiada de palabras clave. 4) El proyecto Virginia Heritage utiliza formularios de consulta. Sus documentos se indexan a texto completo mediante Open Text y se ofrecen en HTML. 5) Library Of Congress: la Library of Congress comenzó a suministrar sus documentos en SGML mediante Panorama. En la actualidad también se suministran en HTML. El documento etiquetado en SGML se mapea a una base de datos InQuery, de manera regular o bajo demanda. Los documentos HTML se derivan de la base de datos en respuesta a las búsquedas. 6) Berkeley Art Museum/Pacific Film Archive: el Berkeley Art Museum, incorporado al Online Archive of California, utilizó el motor de búsqueda Isite, con interfaz web para realizar búsquedas booleanas por campos, en SGML, HTML, MARC y otros, y que soporta Z39.50. Se personalizó Isite para que convirtiera dinámicamente EAD a HTML, y para que dividiera largas listas de contenedores en páginas de tamaño razonable. En cuanto a los servidores, utilizaron un modelo centralizado/descentralizado, en el que los instrumentos de descripción en texto/SGML se almacenaban centralmente a efectos de búsqueda y visualización, pero las imágenes se almacenaban localmente, en las instituciones que las albergaban, para visualizarlas a partir del instrumento de descripción centralizado y mediante el atributo href.

[48] Guidelines for OPAC Displays / prepared for the IFLA Task Force on Guidelines for OPAC Displays by Martha M. Yee. IFLA, 1998

[49] Esta afirmación resulta algo arriesgada, en la medida en la que carecemos de datos estadísticos. Citemos, sin embargo, un par de opiniones contrastadas. En entrevista a Tom Welsh, consultor del Cutter Consortium, y respondiendo a una pregunta acerca del futuro de XML en los próximos cinco años, éste se expresa del siguiente modo: “No hay duda de que XML impregnará la industria de software –no sólo a causa de su fortaleza técnica, sino también por medio de la cooperación sin precedentes que hha generado. En el desarrollo de software, en las aplicaciones de gestión de datos, en la gestión de operaciones y sistemas... estará en todas partes. Incluso se ha sugerido que el correo electrónico podría basarse en XML en el futuro, con grandes beneficios potenciales. Algunas de las predicciones más optimistas sobre los servicios web, por ejemplo, pueden no quedar satisfechas –no durante muchos años, al menos. Gran parte de ello depende de los esfuerzos permanentes para definir vocabularios u ontologías compartidos; éste es un reto realmente fundamental”. En: XML: One Metalanguage to Rule Them All? CC, 2004. URL:

<http://www.cutter.com/consultants/interview.html> (Consulta: 25-3-2004). Perteneciente al mismo Consorcio, Tom Bragg informa de que, de las empresas encuestadas (norteamericanas), tres cuartos están usando XML, y la cuarta parte de ellas lo están haciendo en proyectos “grandes” o “principales”. En: XML use on the rise. CC, 2001. URL: <http://www.cutter.com/research/2001/edge011002.html> (Consulta: 25-3-2004)

[50] “La Web semántica es la representación abstracta de datos en la World Wide Web, basada en las normas

RDF y otras normas a ser definidas. Está siendo desarrollada por el W3C, en colaboración con un amplio número de investigadores y socios industriales". Nykänen, Ossi: World Wide Web Consortium (W3C). W3C Office, [200-?] URL: <http://www.w3c.tut.fi/talks/2003/0331umedia-on/slide6-0.html> (Consulta: 25-3-2004)

[51] A modo de ejemplo puede citarse, en el ámbito comercial, XMLSpy de Altova, que permite esta conversión de DTDs a schemas XML, así como el trabajo con ambos métodos, el chequeo de la validez y conformidad de los documentos, etc. Como software de uso libre, mencionemos DTD2XS, que convierte automáticamente DTDs a schemas XML, generando al tiempo un exhaustivo informe del proceso.

[52] Resource Description Framework (RDF). World Wide Web Consortium, 2004. URL: <http://www.w3.org/RDF/> (Consulta: 25-3-2004)

[53] Manola, Frank, Miller, Eric, eds.: RDF Primer W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004. URL: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> (Consulta: 25-3-2004). Además de esta introducción, puede resultar útil para conocer los principios básicos de RDF, el artículo: Iannella, Renato: An Idiot's Guide to the Resource Description Framework. En: *The New Review of Information Networking*. Vol. 4 (1998). URL: <http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/> (Consulta: 25-3-2004)

[54] ISO/IEC FCD FCD 11179-1: Information technology — Metadata registries. Part 1, Framework. International Standardization Organization, 2003. P. 14

[55] CORES: A Forum on Shared Metadata Vocabularies: Glossary. URL: <http://www.cores-eu.net/glossary/> (Consulta: 25-4-2004)

[56] Puede consultarse, por ejemplo, una buena historia de la gestión distribuida de metadatos en: Inmon, William H.: A Brief History of Metadata: From Master Files to Distributed Metadata. Billinmo, 2001.

[57] Alguna institución, como la Niels Böhr Library, sí declara haber utilizado EAD en combinación con RDF.

BIBLIOGRAFÍA DE INTERÉS

Archival authority information: Discussion paper 2: Elements of a successful standard and prototype implementation. URL: <http://www.library.yale.edu/eac/discussion2.htm>

Archives and Museum Informatics: Cultural Heritage Informatics Quarterly. Vol. 12, N. 3-4 (1998). URL: <http://www.kluweronline.com/issuetoc.htm/1042-1467+12+3/4+1998>

Archives of American Art Finding Aids. URL: <http://archivesofamericanart.si.edu/findaids/findaids.htm>

Bodleian Library. Department of Western Manuscripts. URL: <http://www.bodley.ox.ac.uk/guides/wmss/>

Bouché, Nicole L: *Implementing EAD in the Yale University Library*. En: *The American Archivist*. Vol. 60, n. 4 (Fall 1997). P. 408-19.

Bryan, Martin: *An Introduction to the Extensible Markup Language (XML)*. URL: <http://www.personal.u-net.com/~sgml/xmlintro.htm>

Comeau, Sheila: *Encoded Archival Description (EAD) and the Creation of Electronic Finding Aids*. En:

National Library of Canada (December 7, 1998). URL: <http://www.nlc-bnc.ca/9/1/p1-257-e.html>

Cover, Robin: *The SGML/XML Web Page*. URL: <http://www.oasis-open.org/cover>

DeRose, Stephen J.: *Navigation, Access, and Control Using Structured Information*. *The American Archivist*. Vol. 60, n. 3 (Summer 1997). P. 298-309.

Descripción archivística codificada: directrices de aplicación. s.l.: Fundación Histórica Tavera, 2000

Descripción archivística codificada: repertorio de etiquetas. V. 1.0. [s.l.]: Fundación Histórica Tavera, 2000

Design Principles for Enhancements to EAD. URL: <http://www.loc.gov/ead/eaddesgn.html>

Development of the Encoded Archival Description Document Type Definition. URL: <http://lcweb.loc.gov/ead/eadback.html>

Dow, Elizabeth H.: *EAD and the Small Repository*. En: *The American Archivist*, Vol. 60, N. 4 (Fall 1997). P. 446-455

EAC: Encoding Archival Context. URL: <http://www.library.yale.edu/eac/>

EAD: Descripción Archivística Codificada: directrices de aplicación /elaboradas por el Encoded Archival Description Working Group de la Society of American Archivists. V. 1.0. [s.l.]: Fundación Histórica Tavera, 2000

EAD Help Pages: EAD Sites Annotated. URL: <http://jefferson.village.virginia.edu/ead/sitesann.html>

EAD Help Pages: Helper Files. URL: <http://jefferson.village.virginia.edu/ead/helper.html>

EAD Help Pages: Software. URL: <http://jefferson.village.virginia.edu/ead/products.html>

EAD Tools and Resources. URL: http://www.lib.ncsu.edu/archives/tech_serv/eadtools.html#scripts

Encoded Archival Context: Working Drafts. URL: <http://jefferson.village.virginia.edu/eac/>

Encoded Archival Description: Rare Book, Manuscript, and Special Collections Library Duke University. URL: <http://scriptorium.lib.duke.edu/findaid/ead/>

The Encoded Archival Description: Retrospective Conversion Guidelines: A Supplement to the EAD Tag Library and EAD Guidelines. Berkeley: American Heritage Virtual Archive Project ; University of California Encoded Archival Description Project

Encoded Archival Description Document Type Definition. V. 2002. Prepared and Maintained by the Encoded Archival Description Working Group of the Society of American Archivists and the Network Development and MARC Standards Office of the Library of Congress. Chicago: Society of American Archivists, 2002

Encoded Archival Description Document Type Definition. File name: eadlocal.ent. Version 2002. Editor, Daniel V. Pitti

Encoded Archival Description Tag Library. Version 2002 / Prepared and Maintained by the Encoded Archival Description Working Group of the Society of American Archivists and the Network Development and MARC Standards Office of the Library of Congress. Chicago: Society of American Archivists. URL: <http://www.loc.gov/ead/tglib/index.html>

Extensible Mark-Up Language (XML). URL:

- <http://www.w3.org/XML/>
Finding Aids for Manuscripts and UCSD Archives. URL: <http://orpheus.ucsd.edu/speccoll/testing/mscl-fa1.html>
- Finding Aids in the Niels Bohr Library.* URL: <http://www.aip.org/history/ead/index.html>
- Fox, Michael: *The EAD Cookbook and Related Tools.* URL: <http://jefferson.village.virginia.edu/ead/cookbookhelp.html>
- Fox, Michael: *Implementing Encoded Archival Description: An Overview of Administrative and Technical Considerations.* En: The American Archivist, Vol. 60, N. 3 (Summer 1997)
- Gartrell, David C.: *Word Processing and Database Software as EAD Encoding Tools.* En Archives and Museum Informatics: Cultural Heritage Informatics Quarterly. Vol. 12, N. 3-4 (1998). URL: <http://www.kluweronline.com/issuetoc.htm/1042-1467+12+3/4+1998>. P. 277-286
- Gilliland-Swetland, Anne J.: *Setting the Stage.* En Introduction to Metadata: Pathways to Digital Information. URL: http://www.getty.edu/research/institute/standards/intro/metadata/2_articles/index.html
- Hensen, Steven L.: *'NISTF II' and EAD: The Evolution of Archival Description.* En: The American Archivist. Vol. 60, n. 3 (Summer 1997). P. 284-97.
- Higgins, Richard: *The Encoded Archival Description: using SGML to create permanent electronic handlists.* URL: <http://www.dur.ac.uk/Library/asc/eadarticle.html>
- International Organization for Standardization: *ISO 8879-1986 (E). Information Processing: Text and Office Systems: Standard Generalized Markup Language (SGML).* Geneva: International Organization for Standardization, 1986.
- International Organization for Standardization: *ISO 8879:1986 / A1:1988 (E). Information Processing: Text and Office Systems: Standard Generalized Markup Language (SGML), Amendment 1.* Geneva: International Organization for Standardization, 1988.
- International Organization for Standardization: *ISO/TR 9573-1988(E). Information processing: SGML Support Facilities: Techniques for Using SGML.* Geneva: International Organization for Standardization, 1988.
- Kiesling, Kris: *EAD as an Archival Descriptive Standard.* En: The American Archivist. Vol. 60, n. 3 (Summer 1997). P. 344-54.
- Lacy, Mary A. y Anne Mitchell: *EAD Testing and Implementation at the Library of Congress.* En: The American Archivist, Vol. 60, N. 4 (Fall 1997). P. 420-435
- List of Collections on Durham University Library Dynaweb Server.* URL: <http://flambard.dur.ac.uk:6336/dynaweb>
- Meissner, Dennis: *First Things First: Reengineering Findings Aids for Implementation of EAD.* En: The American Archivist, Vol. 60, N. 4 (Fall 1997). P. 372-387
- Montoya, Gabriela A.: *Conversion of Microsoft® Access Databases into EAD-encoded Finding Aids.* URL: <http://sunsite.berkeley.edu/FindingAids/uc-ead/tools/database/>
- Morris, Leslie A.: *Developing a Cooperative Intra-institutional Approach to EAD Implementation: The Harvard/Radcliffe Digital Findings Aids Project.* En: The American Archivist. Vol. 60, n. 4 (Fall 1997). P. 388-407.
- Overview of SGML Resources.* URL: <http://www.w3.org/MarkUp/SGML/>
- Pitti, Daniel V.: *Encoded Archival Description: An Introduction and Overview.* En D-Lib Magazine. Vol. 5, n. 11 (November 1999)
- Pitti, Daniel V.: *Encoded Archival Description: The Development of an Encoding Standard for Archival Finding Aids.* En: The American Archivist. Vol. 60, n. 3 (Summer 1997). P. 268-283
- Pollock, Alvin: *EAD and Databases: Perl and ADO on MS Windows.* URL: [http://sunsite.berkeley.edu/ead/eaddb/ResourceDescriptionFramework\(RDF\)](http://sunsite.berkeley.edu/ead/eaddb/ResourceDescriptionFramework(RDF)). URL: <http://www.w3.org/RDF/>
- RLG Best Practice Guidelines for Encoded Archival Description.* RLG EAD Advisory Group. 2002
- Ruth, Janice E.: *Encoded Archival Description: A Structural Overview.* En: The American Archivist. Vol. 60, n. 3 (Summer 1997). P. 310-329
- Seaman, David: *Multi-institutional EAD: The University of Virginia's Role in the American Heritage Project.* En: The American Archivist, Vol. 60, N. 4 (Fall 1997). P. 436-444
- Search Harvard Finding Aids.* URL: <http://findingaids.harvard.edu/dfap/>
- SGML: Getting Started: A Guide to SGML (Standard Generalized Markup Language) and Its Role in Information Management: An Arbortext SGML White Paper.* URL: <http://www.arbortext.com/wp.html>
- Special Collections : Finding Aids.* URL: <http://www-sul.stanford.edu/depts/spc/findaids.html>
- UC-EAD.* URL: <ftp://library.berkeley.edu/pub/sgml/uc-ead/>
- Walsh, Norman: *A Technical Introduction to XML.* En: Arbortext: Think Tank. URL: http://www.arbortext.com/Think_Tank/XML_Resource/A_Technical_Introduction_to_XML/a_technical_introduction_to_xm.html