

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). **Novática** edita asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (UPGRADE European Network).

<<http://www.ati.es/novatica/>>
<<http://www.ati.es/reicis/>>
<<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIT**, **ASTIC**, **RITSI** e **HispanLinux**, junto a la que participa en **ProInnova**.

Consejo Editorial
Antoni Carbonell Nogueras, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte, Francisco López Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepát, Miquel Sàrries Grifó, Asunción Yturbe Herranz

Coordinación Editorial
Llorenç Pagés Casas <pages@ati.es>

Composición y autedición
Jorge Llácer Gil de Ramales

Traducciones
Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>, Dpto. de Sistemas Informáticos - Escuela Superior Politécnica - Universidad Europea de Madrid

Administración
Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores
Acceso y recuperación de la información
José María Gómez Hidalgo (Universidad Europea de Madrid), <jmgomez@uem.es>
Manuel J. María López (Universidad de Huelva), <manuel.maria@diestia.uhu.es>

Administración Pública electrónica
Francisco López Crespo (MAE), <flc@ati.es>
Gumersindo García Arribas (MAP), <gumersindo.garcia@map.es>

Arquitecturas
Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>
Jordi Tubella Morgadas (DAC-UPC), <jordi@ac.upc.es>

Auditores IT/ITe
Marina Tourinho Troilito, <marinatourinho@marinatourinho.com>
Manuel Palaço García-Suelto (ASIA), <manuel@palaço.com>

Derecho e tecnologías
Isabel Hernando Colliázos (Fac. Derecho de Donostia, UPV), <ihernando@legalek.net>
Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>

Enseñanza Universitaria de la Informática
Joaquín Ezpeleta Mateo (CPS-UZAR), <ezpeleta@posta.unizar.es>
Cristóbal Párraga Flores (DSIP-UCM), <cparraga@sip.ucm.es>

Entorno digital personal
Alonso Álvarez García (TID), <aag@tid.es>
Diego Gachet Páez (Universidad Europea de Madrid), <gachet@uem.es>

Basión del Conocimiento
Joan Baiget Solé (Cap Gemini Ernst & Young), <joan.baiget@ati.es>

Informática y Filosofía
José Ángel Olivas Varela (Escuela Superior de Informática, UCLM)
Karim Ghemri Martín (Indra Sistemas)

Informática Gráfica
Miguel Chover Sellés (Universitat Jaume I de Castellón), <chover@lsi.uji.es>
Roberto Vivo Hernández (Eurographics, sección española), <rvivo@dsic.upv.es>

Ingeniería del Software
Javier Dolado Cosin (DLSI-UPV), <dolado@si.ehu.es>
Luis Fernández Sanz (PRIS-El-UEM), <lufernan@pris.esi.uem.es>

Inteligencia Artificial
Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV)
<vbotti_vinglada@dsic.upv.es>

Información Persona-Computador
Julio Abascal González (FI-UPV), <julio@si.ehu.es>

Lengua e Informática
M. del Carmen Ugarte García (IBM), <cugarte@ati.es>

Lenguajes Informáticos
Andrés María López (Univ. Carlos III), <amarin@dit.uc3m.es>
J. Angel Velázquez Irujibe (ESCET-URJC), <a.velazquez@escet.urjc.es>

Lingüística computacional
Xavier Gómez Guinovart (Univ. de Vigo), <xggo@uvigo.es>
Manuel Palomar (Univ. de Alicante), <mpalomar@dsi.ua.es>

Mundo estudiantil
Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM), <a.vazquez@ieee.org>
Federico G. Mon Troiti (RITSI), <gnu.fede@gmail.com>

Formación Informática
Rafael Fernández Calvo (ATI), <rflcalvo@ati.es>
Miquel Sàrries Grifó (Ayto. de Barcelona), <msarries@ati.es>

Notas y servicios telemáticos
José Luis Marzo Lázaro (Univ. de Girona), <jose.luis.marzo@udg.es>
Josep Solé Pareta (DAC-UPC), <pareta@ac.upc.es>

Seguridad
Javier Arellito Bertolin (Univ. de Deusto), <jarellito@eside.deusto.es>
Javier López Muñoz (ETSI Informática-UMA), <jlm@cc.uma.es>

Sistemas de Tiempo Real
Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <aalonso_ipeunte@dit.upm.es>

Sustentación
Jesus M. González Barahona, Pedro de las Heras Quiros (GSYC-URJC), <jlgp.pheras@gsyc.es>

Tecnología de Objetos
Jesus García Molina (DSIC-UJM), <jmolina@um.es>
Gustavo Rossi (LIFA-UNLP, Argentina), <gustavo@sol.info.unlp.edu.ar>

Tecnologías para la Educación
Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>
Julia Mingullón i Alfonso UOC), <jmingullona@uoc.edu>

Tecnologías y Empresa
Didac López Butifull (Universitat de Girona), <didac.lopez@ati.es>
Francisco Javier Cantais Sanchez (Indra Sistemas), <fjcantais@gmail.com>

TIC y Turismo
Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)
<aguayo_guevara@cc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de **cc-by** o **copyright** elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Padilla 66, 3º dcha., 28006 Madrid
Tfno. 91 4029391; fax. 91 3093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia
Av. del Reino de Valencia 23, 46005 Valencia
Tfno./fax 96 3303032 <secreval@ati.es>

Administración y Redacción ATI Cataluña
Via Laietana 46, ppal. 1º, 08018 Barcelona
Tfno. 93 4125235; fax 93 4127713 <secregen@ati.es>

Redacción ATI Andalucía
Isaac Newton, s/n, Ed. Sadleir,
Isla Cartuja 41092 Sevilla, Tfno./fax 95 4460779 <secreand@ati.es>

Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza.
Tfno./fax 9 76235181 <secreara@ati.es>

Redacción ATI Asturias-Cantabria <gp-astucan@ati.es>
Redacción ATI Castilla-La Mancha <gp-clmancha@ati.es>

Suscripción y Ventas
<<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid

Publicidad
Padilla 66, 3º dcha., 28006 Madrid
Tfno. 91 4029391; fax. 91 3093685 <novatica@ati.es>

Imprenta
Derra S.A., Juan de Austria 66, 08005 Barcelona.

Deposito legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVACQ

Partida: "Alan Turing & friends" (variaciones sobre una foto tomada de www.turing.org).

RFCADA / © Rafael Fernández Calvo 2007
Diseño: Fernando Agresta / © ATI 2006

editorial

La enseñanza de la Informática en España

> 02

en resumen

Un estándar, dos estándares

> 02

Llorenç Pagés Casas

noticias IFIP

Actividades del IFIP TC6 Technical Committee on Communication Networks

> 03

Ramón Puigjaner Trepát

monografía

Formato de Documento Abierto (ODF)

(En colaboración con UPGRADE)

Editores invitados: Jesús Tramullas Saz, Piedad Garrido Picazo, Marco Fioretti

Presentación: OpenDocument, estándar para documentos digitales

> 04

Jesús Tramullas Saz, Piedad Garrido Picazo

Abierto desde el diseño: el Formato de Documento Abierto para aplicaciones ofimáticas

> 06

Erwin Tenhumberg, Donald Harbison, Rob Weir

¿Es OpenDocument un estándar abierto?: ¡Sí!

> 13

David A. Wheeler

Trampas ocultas en OpenDocument y efectos secundarios en el software libre y de código abierto

> 19

Marco Fioretti

ISO-26300 (OpenDocument) vs. MS-Office Open XML

> 22

Alberto Barrionuevo García

Interoperabilidad: ¿se impondrá el verdadero formato universal de ficheros?

> 28

Sam Hiser, Gary Edwards

ODF: el Formato de Documento emergente a elección de los gobiernos

> 36

Marino Marcich

Promoción del uso de los formatos abiertos de documentos por los Programas

> 39

IDA e IDABC

Miguel A. Amutio Gómez

Una historia resumida de los estándares abiertos en Dinamarca

> 42

John Gøtze

Formatos estándares abiertos y software libre en la Administración Pública de Extremadura

> 44

Luis Millán Vázquez de Miguel

secciones técnicas

Enseñanza Universitaria de la Informática

Acciones y reacciones en el camino de la mejora docente universitaria

> 46

Alfonso Blesa Gascón, Pablo Bueso Franc, Carlos Catalán Cantero,

Raquel Lacuesta Gilaberte, Mariano Ubé Sanjuán

Informática Gráfica

Programación de Aplicaciones Gráficas con OpenGL y Java

> 51

Óscar Belmonte Fernández

Redes y servicios telemáticos

Algoritmo bioinspirado para la optimización de rutas en Internet

> 56

José Luis Gahete Díaz, Fernando Gómez González

Referencias autorizadas

> 63

sociedad de la información

Futuros emprendedores

Step by Step: Mens sana in corpore sano

> 70

Miguel Angel Ramos Barroso, Javier Cantón Ferrero, Javier Fernández Rodríguez,

Juan María Laó Ramos

Novática interactiva

Competencia entre estándares, ¿va a ser posible su coexistencia?

> 74

Foro de Debate

Programar es crear

Polígonos en malla (CUPCAM 2006, problema A, enunciado)

> 75

Dolores Lodares González

asuntos interiores

Coordinación editorial / Fe de erratas / Programación de Novática

> 76

Normas para autores / Socios Institucionales

> 77

Monografía del próximo número: "Buscadores en la Web"

Sam Hiser, Gary Edwards
OpenDocument Foundation

<{sam.hiser, gary.edwards}@opendocument.us>

Interoperabilidad: ¿se impondrá el verdadero formato universal de ficheros?

Traducción: Rafael Fernández Calvo (Grupo de Trabajo de Lengua e Informática de ATI)

"Lo que dan con la letra grande lo quitan con la letra pequeña"

Tom Waits (*Small Change*, 1976)

1. Introducción

Nos encontramos en una encrucijada en lo que se refiere a estándares, con la conectividad futura de ordenadores de sobremesa, dispositivos y sistemas de empresa como recompensa. Hay dos implementaciones de XML (*eXtensible Markup Language*) que pueden abarcar los horizontes de este universo emergente y el mundo está esperando a ver cuál de ellas prevalece.

Es tiempo de decidir.

Presentemos a nuestros dos contendientes: **OpenDocument Format** (ODF) y **ECMA Office Open XML** (EOOXML) de Microsoft (**N. del T.:** ECMA es la *European Computer Manufacturers Association*). Microsoft declara que son interoperables, con lo que la promesa de XML de una transformación fluida se habría alcanzado. Sin embargo, los partidarios de ODF no piensan lo mismo; ODF es el único formato universal de ficheros que vale la pena tener en cuenta.

No hay necesidad de tomar una decisión, insiste Microsoft. Si Vd. está haciendo uso de aplicaciones Microsoft y ejecuta EOOXML, entonces este formato es su opción. Si Vd. es uno de los 485 millones de usuarios de Microsoft Office para Windows, encadenados durante décadas a ficheros binarios heredados, y construye sus procesos de negocio diarios sobre estos ficheros, entonces EOOXML es para Vd.. Y sólo EOOXML.

Si Vd. está haciendo uso de cualquier otro tipo de aplicación, entonces ODF es su mejor elección. Microsoft está encantado de consentir la elección entre estos dos estándares de documentos [sic] ya que, mientras nosotros estamos hablando, Microsoft está trabajando en conseguir la interoperabilidad entre los dos formatos. Después de todo, tanto ODF como EOOXML son XML. ¿Verdadero?

¡Falso! Y ahí radica el cuento sobre una decisión que determinará el futuro. La interoperabilidad y la transformación sencilla son el distintivo de XML. Pero no es esto lo que está sucediendo aquí.

Resumen: en este artículo se comparan las dos soluciones que se enfrentan para convertirse en el estándar de formato universal de ficheros: ODF (OpenDocument Format), de la OpenDocument Foundation, y EOOXML (ECMA Office Open XML) de Microsoft. Ambas están basadas en XML (eXtensible Markup Language) y se analizan tanto desde un punto de vista técnico como en su repercusión futura sobre las empresas, teniendo en cuenta principalmente su interoperabilidad real y su capacidad para obtener la mayor fidelidad posible en la conversión de ficheros heredados.

Palabras clave: conversión de ficheros, EOOXML, estándares, formato universal, interoperabilidad, ODF, XML.

Autores

Sam Hiser es Vicepresidente y Director de Negocio de la OpenDocument Foundation. Posee un MBA por la Universidad de Duke. Fue director del proyecto de Marketing de OpenOffice.org y junto a Tom Aldelstein es el autor del libro "Exploring the JDS Linux Desktop". Contribuye regularmente a la sección "Digital Business" (Negocios Digitales) del *Financial Times*.

Gary Edwards es el Presidente de la OpenDocument Foundation y socio fundador del Comité Técnico del OASIS Open Office XML. Radicado en Redmond (California), es Consultor Tecnológico especialista en el diseño e implantación de soluciones de reingeniería de procesos de negocio.

Microsoft, por supuesto, hará todo lo posible para hacer creer que el sueño de la interoperabilidad se ha hecho ya realidad, incluyendo el fraude de la firma de un elaborado conjunto de acuerdos con un débil proveedor de Linux y OpenOffice para demostrar que Microsoft está comprometido a cumplir los requerimientos del usuario. Si lo examina detenidamente verá que el único requerimiento del usuario en el que Microsoft se centra es en la extensión al nuevo mundo de Internet, XML y SOA (*Service Oriented Architecture*) de las viejas características propietarias de sus documentos. Si da un paso atrás y tiene en cuenta todo lo anterior, se dará cuenta de que el objetivo último es extender el monopolio de los ordenadores de sobremesa a los servidores, dispositivos y sistemas de información globalmente conectados. Esta vez juegan con todas las cartas.

¿Hay problemas legales y técnicos de importancia en determinadas secciones del acuerdo de compartición de tecnología entre Microsoft y Novell? Sin lugar a dudas, sí. Cada grupo de interesados tiene diferentes reservas sobre distintas facetas del acuerdo. Los desarrolladores de software libre y los que se acogen a la licencia GPL (*General Public Licence*) se ven afectados negativamente por el acuerdo de no litigación sobre patentes; la competencia formada por los distribuidores de Linux y los desarrolladores de software libre está comprensiblemente irritada con la espuria protección de paten-

tes así como con acuerdos cruzados de virtualización y de colaboración en ventas; mientras que los usuarios y los gobiernos están verdaderamente preocupados por si los acuerdos de 'interoperabilidad' establecidos entre Novell y Microsoft sobre el paquete Office pudiesen limitar sus opciones.

Hoy día nuestras preocupaciones se centran en la letra pequeña de este último aspecto del acuerdo Microsoft-Novell y en situar el asunto en el contexto más amplio de la ISO (*International Organization for Standardization*), la SOA y el futuro de las altamente avanzadas y extremadamente productivas cadenas de proceso de la información en las que se incluyen ordenadores de sobremesa, servidores y dispositivos.

Office Open XML ("El proyecto conjunto interempresarial para desarrollar un mecanismo para facilitar la conversión entre ficheros OpenDocument, ODF, y ficheros Microsoft/ECMA, EOOXML") está lleno de problemas. Se presenta a sí mismo como una solución de 'interoperabilidad' entre Microsoft Office y OpenOffice.org. Y no lo es¹. Incluso a primera vista, la especificación de Microsoft para EOOXML tiene graves defectos y es imposible de implementar en su totalidad por quienes no sean desarrolladores de Microsoft. Por lo que tiene serios problemas para ser aceptada por la ISO como estándar internacional. Pero sólo con entender algunos de los serios problemas de EOOXML, se descubre claramente el carác-

ter de acción subrepticia contra la libre competencia que tienen este proyecto de 'interoperabilidad' y el acuerdo entre Microsoft y Novell.

2. Migrando a XML ficheros heredados de Microsoft

Todo el mundo está de acuerdo en que XML es ciertamente el formato de documentos del próximo futuro. XML es tanto el futuro de Internet como el núcleo central de cualquier SOA. No sólo es "el futuro", sino también el puente entre el lugar en el que estamos hoy (la infraestructura que hemos heredado) y lo que necesitamos hacer para llegar a ese futuro.

Microsoft quiere 'tirar' de los clientes hacia su versión propietaria de XML (hablar de "XML propietario" es indudablemente una incongruencia, pero dejaremos esto a un lado por ahora); la comunidad ODF quiere conseguir que todos usemos su implementación universal y portable de XML; y los clientes están claramente confundidos ya que la opción de Microsoft parece aceptable, aunque cara.

Además, la decisión de no decidirse entre EOOXML y ODF — ya que ambas opciones son aceptables — no pondría en peligro el puesto de trabajo de un Director de Sistemas de Información. Dando por hecho que la gente confía en Microsoft, y mientras que la gente esté de acuerdo en que Microsoft sólo se preocupa de los intereses de los clientes (como de costumbre), esta decisión de no decidirse parece segura. Incluso si la gente se mantiene escéptica acerca de los motivos de Microsoft, existe la bien arraigada creencia de que decantarse por Microsoft es simplemente inevitable, debido al poder del aparato comercial del monopolista en el mercado. Y, por otro lado, la opción de ODF (aunque lleve dos años de adelanto en el proceso de estandarización de la ISO) sigue pareciendo poco articulada y le falta integración en la cadena de procesos de negocio. Parece razonable.

Pero ocurre que ODF es creíble y ofrece a quienes la adopten primero (en particular Administraciones Públicas) algunos beneficios realmente interesantes en lo que se refiere a acceso perpetuo a la información, bajo coste y nueva flexibilidad en la adquisición de sistemas (ver más detalles sobre la SOA más adelante). Estos beneficios se hacen realmente evidentes para una organización cuando ha conseguido realizar un proceso de migración de forma satisfactoria y empieza a crear documentos en una implementación abierta basada en XML.

Los argumentos a favor de ODF los entienden perfectamente quienes se han comprometido con este formato (ellos saben quiénes son). Ahora bien, están preocupados sobre todo por: a) cómo consigo pasar a ODF mis documentos Microsoft heredados;

b) cómo puedo gestionar formatos mixtos y sistemas mixtos durante e incluso mucho después del proceso de transición que ha realizado mi organización; c) cómo puedo resolver las irregularidades introducidas en los procesos normales de negocio por la travesía de documentos a través de diferentes sistemas tanto dentro como fuera de mi organización; y d) cómo puedo hacer uso de documentos de estándares abiertos XML para separar mis desarrollos de las APIs (*Application Program Interfaces*) de la capa aplicativa — en particular de las que son propiedad de una empresa sociópata conocida por lanzar una segunda colección de libros acerca de sus propias APIs?

Por esto es por lo que la palabra 'interoperabilidad' está en boca de todos. Con XML en el horizonte, todos los consumidores están martilleando a Microsoft y a la comunidad ODF para que resuelvan los problemas de incompatibilidad de formatos de fichero que han afectado al comportamiento de los documentos tanto dentro como fuera de la familia de formatos ofrecida por Microsoft. Por consiguiente, Microsoft se ha enamorado de la palabra 'interoperabilidad' pero no de las acciones necesarias para alcanzarla.

El mandato emitido por los consumidores es que quieren que tanto los servidores como los dispositivos de los sistemas se integren completamente con aplicaciones de escritorio Windows | Microsoft Office. Al hablar de integración se refieren exactamente a que quieren un formato abierto de ficheros XML independiente de la aplicación, y de la plataforma, capaz de conectar las aplicaciones de sobremesa MSOffice tanto al servidor como a los dispositivos de los sistemas. Resumiendo, los consumidores de tecnologías de la información quieren una propiedad completa y clara sobre su información y sobre el procesamiento de ésta, control sobre la información diaria y los procesos de negocio en los que se basan sus flujos de trabajo y sus servicios.

Pero Microsoft se muestra reacio a facilitar que los consumidores abandonen los formatos que controla. En realidad, Microsoft conoce perfectamente sus formatos secretos. *Perfectamente*. Cualquier empresa que pueda sincronizar una conversión perfecta y fiel entre los miles de millones de ficheros binarios y EOOXML, bien de forma nativa bien mediante un conector, podría también proporcionar fácilmente una fidelidad del 100% usando ODF en un plazo de pocas semanas de desarrollo. Las ventajas estructurales comparativas de ODF sobre EOOXML son tan arrolladoras que, si no fuera por la estrategia de hacer progresar y extender el monopolio, Microsoft tendría todas las razones técnicas para elegir ODF en vez de EOOXML.

Para los consumidores 'interoperabilidad' significa un 100% de fidelidad en los ficheros al pasar de una aplicación a otra, tanto si son del mismo tipo como si no lo son, sin importar la plataforma subyacente. El requerimiento no es complicado y ha sido definido claramente en Massachusetts en la petición realizada por la ITD (*Information Technology Division*) de este Estado para obtener información sobre un *plugin* ODF para MS Office.

Uno de los trabajos más tramposos que se están realizando hoy en toda la cristiandad es el empeño de Microsoft para 'tirar' (como ellos lo llaman de forma eufemística) de sus cerca de quinientos millones de clientes hacia una (o dos) versión moderna y con soporte de sus sistemas operativos, aplicaciones y tecnologías de formato de ficheros. Debido a sus apetitos lujuriosos, trabajo duro, liderazgo sin escrúpulos y una cultura competitiva sociópata, la empresa ha agitado a su clientela con tantas versiones que su principal competidor es *ella misma*. (Un derivado de ese empeño principal, el de vender, es la necesidad de llegar a dominar las dificultades que plantea poner fin a las versiones anteriores sin llamar la atención de los reguladores o sin irritar a los consumidores más allá de un determinado punto).

Quizás sea porque Redmond siente que ha forzado este juego demasiado a menudo, pero ahora estamos observando un cambio de estrategia. MSOffice 2007 es una aplicación de transición, muy ambidextra en el sentido de que para EOOXML tiene una API Win32 heredada vinculada a BoBs (miles de millones de documentos binarios) y un EOOXML diferente para documentos nativos vinculados a la API de Vista-.Net 3.0. De hecho, es posible tener el mismo documento en MSOffice 2007, una versión nativa y otra heredada, con diferentes elementos internos y dependencias EOOXML. ¿Qué pasa?

Si bien nadie lo sabe con seguridad, parece que el punto de bloqueo de clientes se ha trasladado de los documentos y procesos de negocio vinculados a Microsoft Office a una cadena de proceso de la información formada por MSOffice 2007 <=> EOOXML <=> IE 7.0 <=> y el Exchange/SharePoint Hub. Es éste el *hub* de E/S donde se produce el nuevo bloqueo ya que todos los documentos EOOXML son convertidos aquí a unas dependencias basadas en la API de Vista-.NET 3.0. MS Office es sólo la cabecera de esta cadena de proceso.

Desde el *hub* de E/S, hay que esperar que se produzca una acelerada conectividad EOOXML con MSLive, MSN, MS-ERP, SAP, MSSQL Servers y con los sistemas de proceso de transacciones de trastienda (*backend*), a Active Directory, Collaborative Server y sistemas MSOffice Server (por citar

sólo unos pocos). Hay que esperar también que el VSTO 2005 preparado para EOOXML engrase los rieles para migrar los sistemas de proceso de negocio vinculados a MS Office al *hub* E/S y alimente el rápido desarrollo de aplicaciones de negocio listas para EOOXML y capaces de adaptarse bien a las cadenas de proceso de información dominadas por EOOXML. ¡Esto se llama *integración!* O, como dicen en Redmond, "*Interoperability by Design*".

Llegados a este punto, ¿tenemos que preguntarnos si los sistemas de software y servicios competitivos no ligados a Microsoft pueden interoperar dentro de este extraordinario diseño de la cadena de proceso? Quiero decir, ¿imaginas una aplicación de escritorio OpenOffice, o una aplicación de escritorio MAC Office, participando en esta cadena? ¿Y qué hay de los servicios de combinación de contenidos de Google, Yahoo, o Amazon Web 2.0? ¿O del ERP, bases de datos relacionales y sistemas de proceso de transacciones de Oracle? ¿O del Lotus Notes de IBM? ¿O quizás del SOA de BEA? ¿Entiendes ahora lo que está en juego?

Oh, sí. Adiós Adobe.

Así que, ¿de qué estamos hablando de cualquier forma?

Interoperabilidad (la interoperabilidad verdadera y permanente proporcionada por un formato universal de ficheros) es importante para procesos de negocio de todo tipo basados en la forma en que la gente mueve sus documentos a través de una amplia variedad de sistemas. El proceso de negocio de mayor relevancia desde el punto de vista de la legislación antimonopolio y de los efectos sobre la competencia es el proceso de migración que realiza una empresa para pasar a ODF sus ficheros heredados almacenados en formato binario de Microsoft Office. Si se consigue realizar ese proceso de migración entonces puede decirse que es posible competir con el nuevo juego de procesos de negocio de Microsoft. Pero si esa migración no es posible Microsoft mantendrá el bloqueo sobre sus clientes — no sólo en el mercado de paquetes de aplicaciones (*suites*), de oficina sino también en el campo de los procesos de negocio.

Sí, los BoBs de nuevo (aquellos "miles de millones de binarios"): aquellos miles de millones de documentos de Microsoft Office heredados y almacenados en formato binario, los BoBs que Microsoft nos recuerda con tanta frecuencia cuando se discute sobre la adopción de EOOXML como estándar. El convertidor de ficheros Novell-Microsoft Clever Age se limita a conversiones de EOOXML a ODF y no convierte directamente entre ODF y los formatos secretos de

ficheros binarios heredados de Microsoft, que continúan siendo utilizados por Microsoft Office en su proceso interno.

3. El traductor a ODF "Novell-Microsoft-Clever Age" está diseñado a medias intencionadamente

Por al menos cuatro razones, el conversor de ficheros que está desarrollando Clever Age, pagado por Microsoft, e incorporado a la versión de OpenOffice sólo para Novell, nunca será capaz de ofrecer una total interoperabilidad entre Microsoft Office y las aplicaciones que soportan ODF, como OpenOffice.org.

Las razones tienen que ver con características fundamentales tanto del formato EOOXML como del enfoque técnico de Clever Age-Microsoft: XSLT (*eXtensible Stylesheet Language Transformation*). Microsoft sabe de antemano que XSLT es inadecuado.

Esto no es ninguna sorpresa. Bill Gates nunca apuesta sobre nada a menos que tenga todas las cartas. Y mientras la gente piense que EOOXML es solamente XML y que cualquier XML puede ser transformado fácil y universalmente con XSLT, Gates tiene asegurados sus tres ases (95% del mercado de las aplicaciones de escritorio, conversión perfecta y fiel de los BoBs, y un EOOXML vinculado tanto a la API heredada de Windows como a los elementos internos de la API de Vista-.NET 3.0).

3.1. Otros desarrolladores no pueden implementar adecuadamente todo EOOXML

Junto al problema de la dependencia de EOOXML respecto a oscuras instrucciones RTF (*Rich Text Format*), hay un montón de etiquetas en la especificación EOOXML que no pueden ser implementadas adecuadamente por nadie más que Microsoft. Aquí se muestran sólo unos pocos ejemplos recogidos por Ben Langhinrichs (ver <<http://www.geniisoft.com/showcase.nsf/archive/20061027-0829>>):

■ **autoSpaceLikeWord95** (Emulate Word 95 Full Width Character Spacing) - pp. 1378-1379.

■ **footnoteLayoutLikeWW8** (Emulate Word 6.x/95/97 Footnote Placement) - pp. 1416-1417.

■ **lineWrapLikeWord6** (Emulate Word 6.0 Line Wrapping for East Asian Text) - pp. 1426-1427.

■ **mwSmallCaps** (Emulate Word 5.x for Macintosh Small Caps Formatting) - pp. 1427-1429.

■ **shapeLayoutLikeWW8** (Emulate Word 97 Text Wrapping Around Floating Objects) - pp. 1442-1443.

■ **suppressTopSpacingWP** (Emulate WordPerfect 5.x Line Spacing) - pp. 1462-1464.

■ **truncateFontHeightsLikeWP6** (Emulate WordPerfect 6.x Font Height Calculation) - pp. 1467-1468.

■ **useWord2002TableStyleRules** (Emulate Word 2002 Table Style Rules) - pp. 1481-1482.

■ **useWord97LineBreakRules** (Emulate Word 97 East Asian Line Breaking) - pp. 1482-1483.

■ **wpJustification** (Emulate WordPerfect 6.x Paragraph Justification) - pp. 1483-1485.

Se trata de características lamentables de versiones anteriores de Microsoft Office que se han mantenido en esta nueva edición solamente con el propósito de compatibilidad con anteriores versiones. Casi todas estas etiquetas de la especificación EOOXML están acompañadas por unos pocos 'consejos' repetidos que solamente un cínico podría ver con algún sentido de hilaridad o que, de hecho, provocan risas convulsivas en los conocedores de los formatos de ficheros: "*Para reproducir fielmente este comportamiento, las aplicaciones deben imitar el comportamiento de esa aplicación, que implica muchos posibles comportamientos y no puede ser fielmente ubicado en la narrativa de este estándar Open Office XML. Si las aplicaciones desean replicar ese comportamiento deben utilizar y duplicar la salida de dichas aplicaciones.*"

La especificación y el formato de ficheros EOOXML incuestionablemente contienen cientos de instrucciones de proceso que en su conjunto sólo pueden ser procesadas adecuadamente por Microsoft Office. Es un formato de ficheros propiedad de un solo suministrador, un formato de ficheros que trata el software como un punto de llegada en vez de como un enrutador de información a un conjunto de software de procesos de negocio que no sea el propio y estrechamente integrado de Microsoft.

Las citadas etiquetas específicas de aplicación podrían ser también burbujas binarias en lo que se refiere a aplicaciones ajenas a Microsoft. La funcionalidad de las etiquetas está escondida en la capa de aplicación no especificada de Microsoft Office en vez de estar especificada en el estándar EOOXML adoptado por ECMA. La existencia de esas etiquetas desmiente cualquier pretensión de Microsoft de que EOOXML se ha diseñado para la interoperabilidad, una pretensión que aparece en el borrador final de la especificación ECMA, Parte 1: "*El objetivo es permitir la implementación de formatos Open Office XML para el más amplio conjunto de herramientas y plataformas, fomentando la interoperabilidad entre las aplicaciones ofimáticas y los sistemas para líneas de negocio, así como dar soporte y fortalecer el archivo y la preservación de los documentos, todo de una forma que sea totalmente compatible con*

las grandes inversiones existentes en documentos Microsoft Office." (Subrayado añadido por los autores).

Cierra los ojos otra vez. Se ignora el hecho de que ni las especificaciones para los formatos binarios heredados ni las APIs usadas para las conversiones se pueden localizar en la especificación EOOXML; siguen siendo un secreto muy bien guardado de Microsoft. Toda esta 'compatibilidad' con los formatos binarios se reserva para uso exclusivo de los productos de un único proveedor, Microsoft.

Puede empezarse a ver con qué claridad funciona el modus operandi. Mediante una oscuridad sistemática y una prestidigitación audaz, Microsoft conseguirá que la ISO le garantice un monopolio exclusivo de la migración de ficheros Microsoft Office heredados a XML. ¿A qué 'sabor' de XML? Está claro, al de Microsoft. No importa que Microsoft rehusase participar en el desarrollo del estándar internacional de Open Document XML que la ISO ha adoptado ya. No importa que Microsoft rechazase suministrar soporte nativo para OpenDocument en Microsoft Office. No importa que desde febrero de 2003, ODF haya sido capaz de traducir perfectamente todos y cada uno de los BoBs, incluidas las instrucciones de proceso para aplicaciones específicas y las burbujas binarias.

Cierra los ojos y retrocederás a 1995: el software como punto de llegada.

Obviamente, si uno sigue la lógica empleada por Microsoft y ECMA, los consumidores de software necesitarán todavía una tercera o quizás una cuarta implementación de formato de ficheros XML para documentos de oficina. Uno para traducir los ficheros de Word Perfect a XML, e incluso otro más para el Lotus Notes de IBM. Así pues ¿en qué medida supone la nueva era de formatos XML cualquier tipo de mejora substancial sobre la cacofonía de formatos binarios que han bloqueado toda interoperabilidad durante todas estas décadas?

La inclusión de cientos de etiquetas específicas de proveedor no es la única debilidad de EOOXML, por supuesto. Sólo tres ejemplos más: la especificación EOOXML también sufre de una dependencia inapropiada respecto a las máscaras de bit del sistema operativo Windows, a los parches para los fallos específicos de las aplicaciones de Microsoft y un *parser* ad-hoc XML propietario de Microsoft².

No hay que maravillarse en absoluto de que Microsoft considerase necesario pagar a Novell una gran cantidad de dinero para llevar a cabo una implementación parcial de EOOXML en OpenOffice.org.

3.2. EOOXML es un formato de proveedor cerrado

La enorme complejidad de la especificación EOOXML, alrededor de 6.000 páginas, además de su dependencia de etiquetas específicas de aplicación, de las máscaras de bit de Windows y de los parches para los fallos específicos de Office, se traduce en un formato de proveedor cerrado. Esto significa que nadie excepto Microsoft será capaz de dar un soporte ni medianamente completo a la especificación. Esto tiene obvias consecuencias, tal y como explica Bob Sutor, de IBM³: *"La total y correcta implementación [de EOOXML] requerirá la clonación de una gran parte del producto Microsoft. Y eso en el mejor de los casos, pues llevan una década de adelanto. También, dado que han evitado usar estándares de la industria como SVG y MathML, tendrás que reimplementar muchas cosas al modo Microsoft. Mejor empieza ahora. Por tanto, concluyo que mientras Microsoft puede terminar dando soporte a la mayor parte [de EOOXML] (y tendremos que esperar al producto final para ver cuánto y cómo), otros productos probablemente terminarán dando soporte sólo a un subconjunto [de EOOXML]."*

Esto significa que otros productos y otro software NO serán capaces, en la práctica, de entender cualquier [EOOXML] que se les lance. Es intolerable. Por tanto crearán solamente el poco de lo que necesitan y lo distribuirán. ¿Lo distribuirán a quién? Al único software que lo entenderá: Microsoft Office. *Así es cómo veo que evolucionarán las cosas: [EOOXML] será casi totalmente legible y escribible por los productos Microsoft, pero escribible sólo parcialmente por otros productos. Esto significa que los datos en forma [EOOXML] serán absorbidos ampliamente por el ecosistema Microsoft pero que poco escapará para su uso total y práctico en otros ámbitos".*

(Ver también el artículo de Sutor citado en la nota 3, que incluye gráficos que representan los conceptos anteriores, discutidos en términos más generales, y explora la diferencia en significado entre "interoperabilidad" e "intraoperabilidad").

Sólo en teoría resulta precisa la afirmación de Microsoft de que el formato de ficheros EOOXML puede ser implementado por cualquiera. En la práctica, no será posible la implementación total por parte de otros desarrolladores. A menos que, quizás, Novell se comprometa a realizar una ingeniería inversa de todas aquellas funciones de versiones previas de Office invocadas por esos cientos de etiquetas, máscaras de bit de Windows y parches para errores de Microsoft Office.

3.3. Novell y Microsoft no tienen intención de conseguir la interoperabilidad

Es especialmente relevante, a la luz de la complejidad gratuita de la especificación EOOXML, lo que dice Steve Ballmer acerca del conversor de ficheros de EOOXML a ODF desarrollado conjuntamente por Novell y Microsoft: *"Ni el equipo de colaboración intentará construir conversores de ficheros que puedan hacer ficheros cien por cien compatibles entre dos formatos de ficheros. Pero conseguirá el nivel de interoperabilidad que los clientes necesitan para trabajar."*

El director del equipo de desarrollo de OpenOffice.org de Novell, Michael Meeks, está de acuerdo: *"Lo que Ballmer dice es verdad en un sentido; cierto, es probable que la interoperabilidad al 100% entre dos aplicaciones no triviales cualesquiera nunca será posible. Por supuesto eso es engañoso y siempre que la interoperabilidad sea lo suficientemente buena es improbable que la gente eche de menos ese último 2% (o lo que sea). Para alguna gente, el beneficio que se obtiene de añadir el siguiente 1% es tan bajo que a la gente no le preocupa :-)".*

Sin embargo, por lo menos un desarrollador que ha estado realmente luchando a brazo partido con la conversión de formatos de ficheros binarios de Microsoft a OpenDocument está en desacuerdo. Gary Edwards, de la fundación OpenDocument, que representa también a la comunidad OpenOffice.org en el Comité Técnico de OpenDocument de OASIS (*Organization for the Advancement of Structured Information Standards*), dice que el "lo que sea" del Sr. Meeks es substancialmente mayor que el 1 ó 2 por ciento cuando se migran ficheros binarios de Microsoft a OpenDocument usando EOOXML como formato intermedio: *"Los conversores a ODF a los que Ballmer y Novell hacen referencia son realmente los mismos filtros de traducción CleverAge/Microsoft a ODF basados en XSLT con rutinas suplementarias en C# — rutinas que prueban sin ninguna duda que EOOXML no está preparado para XSLT. El convertidor CleverAge será la parte central del pluquin que está siendo desarrollado para el OpenOffice de Novell. Ballmer está en lo cierto al decir que nadie conseguirá en ninguna parte, haciendo uso de métodos XSLT, el 100% de fidelidad que define a la 'interoperabilidad'".*

La razón por la que XSLT nunca funcionará en esta situación es porque XSLT necesita un XML "XPath-perfect" altamente estructurado para una transformación perfecta. ODF fue escrito para ser la estructura XML "XPath-perfect" que le hace falta a XSLT. Sin embargo, EOOXML es todo menos perfecto.

Las deficiencias estructurales de EOOXML que hacen que XSLT sea casi inútil (digamos que consigue un 60% de fidelidad máxima, comparado con la fidelidad, tampoco suficiente, de un 85% que se obtiene con los filtros

tradicionales de ingeniería inversa de OpenOffice.org y que ha sido considerada una hazaña extraordinaria por todo aquel que alguna vez ha intentado hacer este trabajo) están centradas en el modelo de 'estilos' (presentación) con una estructura desajustada que tiene EOOXML.

Esta estructura desajustada y demediada está a su vez directamente relacionada con el sencillo motor de representación (layout) de MSOffice. (Sencillo en comparación con el rico y complejo motor de representación de OpenOffice).

La declaración de Ballmer de que ni Novell ni Microsoft ni CleverAge intentarán alcanzar el 100% de fidelidad significa en efecto que seguirán usando XSLT como método de transformación".

Además, cuando tratamos el software como un enrutador de información en un proceso de negocio, donde la misma información debe ser leída y escrita por un conjunto de diversas aplicaciones, solamente es válida la total interoperabilidad. Un proceso de negocio que permite la pérdida de datos en cada trayecto de una aplicación a otra no es 'interoperabilidad'. Decir "el nivel de interoperabilidad que los clientes necesitan para trabajar" tiene tan poco sentido como decir "parcialmente embarazada".

De hecho, el mismo libro blanco de EOOXML de ECMA dice en la sección 4.6 que cuando se migren ficheros heredados en formato binario Microsoft a XML solamente funcionarán plenamente traducciones de "alta fidelidad":

"Migración de alta fidelidad: OpenXML está diseñado para dar soporte a todas las características de los formatos binarios de Microsoft Office 97-2007. Es difícil exagerar la dificultad de lograr este objetivo y la consiguiente posición única de OpenXML para conseguirlo ... Se pretende que OpenXML permita la futura edición o manipulación al mismo nivel de abstracción que tenga el creador original".

Pero solamente si Microsoft Office es la herramienta de edición. Ver también la sección 4.7 del mismo documento ("*Integration with business data*").

3.4. El proyecto conjunto no satisfará los requerimientos del mercado

A pesar de la pretensión de Novell y Mr. Balmer de que esa migración "suficientemente buena" de formatos Microsoft a ODF es suficiente, en todo el mundo las leyes demandan claramente un 100% de fidelidad en las migraciones de documentos binarios a XML. Ver, por ejemplo, la sección 7001 de ley norteamericana de firma electrónica, se-

gún la cual los registros conservados electrónicamente "*deben reflejar con precisión [] la información contenida en el contrato o en cualquier otro registro*" y estar "*en un formato capaz de ser reproducido con precisión para su posterior consulta, por medio de transmisión, impresión o de cualquier otro método*"; o la sección 7261 de la ley Sarbanes-Oxley (la información financiera no debe "*contener una versión falsa de un hecho material*").

La mediocre migración "suficientemente buena" a ODF de registros heredados contemplada en el acuerdo Novell-Microsoft no es ni de lejos suficientemente buena en este entorno legal. Un único usuario puede ser capaz de convertir un único fichero y luego comprobar manualmente que no se ha perdido ninguna información. La situación es totalmente diferente cuando los documentos deben ser convertidos sucesivamente o en caso de conversión masiva o completamente automatizada de ficheros.

4. Se pueden satisfacer los requerimientos del mercado en cuanto a interoperabilidad

Miguel de Icaza, de Novell, eludió contestar a la siguiente pregunta importante que le hizo Pamela Jones: "*Díganos también por favor, en qué fallan desde su punto de vista, las soluciones de Sun o de la Fundación ODF y si puede explicarnos cuáles son las diferencias entre lo que harán ustedes y ellos*".

Icaza respondió: "*Michael [Meeks], director de nuestro equipo de OpenOffice, que casualmente está visitando la ciudad, dice: 'La solución de la fundación ODF no es software libre; la de Sun no está publicada*".

Contrariamente a la afirmación del Sr. Meeks, la OpenDocument Foundation todavía no ha tomado una decisión final sobre la licencia de los productos de conversión de ficheros que está desarrollando pero se está inclinando decididamente hacia distribuirlos bajo la licencia de software libre y abierto GPL, y ha entablado un diálogo con los líderes de la comunidad ODF.

En lo que respecta a las diferencias entre las soluciones, la solución de Sun es un envoltorio C# sobre OpenOffice.org que usa las tradicionales facilidades de importación-exportación de OOo. Según Edwards, es posible alcanzar una fidelidad de alrededor del 85% en la traducción entre binarios Microsoft y OpenDocument. La base de este método es el mismo proceso de filtrado que ahora usa OpenOffice 2.0. No se puede obtener una 'mejora' por ningún lado. El grupo Microsoft/Clever Age está trabajando en filtros XSLT-C# que traducen entre EOOXML y OpenDocument. Debido a los desafíos estructurales que presenta cualquier método

basado en XSLT, CleverAge debería obtener el premio tecnológico de la década si pudiera alcanzar un máximo de un 60% de fidelidad. Las herramientas de la ODF sin embargo están diseñadas para alcanzar una alta fidelidad superando las trampas de EOOXML, trabajando directamente con los formatos de conversión RTF modificados de Microsoft Office. Una de las herramientas de la fundación es el *plugin* ODF para Microsoft Office, un *plugin* apodado "da Vinci" que usa las mismas APIs internas de Office utilizadas por Microsoft para los formatos que soporta de forma nativa. La otra es una API autónoma XML Infoset de ODF que puede ser usada en aplicaciones (también en el lado del servidor) que soporten ODF para importar y exportar formatos binarios de Microsoft. Una tercera, es un *plugin* da Vinci para OpenOffice.org. Y una cuarta es un "asistente de interoperabilidad" diseñado para garantizar que las estaciones de trabajo OpenOffice integradas en el proceso de negocio y flujo de trabajo de MSOffice produzcan documentos perfectamente interoperables.

Edwards dijo que el *plugin* da Vinci y la API OpenDocument Infoset podrán alcanzar el 100% de fidelidad al traducir entre binarios con formato de Microsoft y ODF para aplicaciones que implementen correctamente la especificación de formato OASIS OpenDocument versión 1.2.

La versión 1.1 de la especificación ODF se centra en las tecnologías de ayuda y ha sido recientemente aceptada por el Comité Técnico OpenDocument de OASIS. La versión ODF 1.2 es la equivalente de "interoperabilidad con esteroides" y recibirá toda la atención del comité en enero y febrero de 2007. Edwards aconseja a los usuarios de Microsoft Office 2007 que cambien las opciones por defecto del paquete y guarden sus trabajos en los formatos tradicionales en lugar de en EOOXML con objeto de facilitar la migración a ODF después de que las herramientas de la Fundación estén disponibles.

Pero dice que hay razones más importantes para evitar el uso de EOOXML: "*Una vez que los procesos de negocio vinculados a Microsoft Office se transfieran a EOOXML, esos procesos estarán listos para la migración hacia la cadena de proceso de información basada en Microsoft Exchange/Sharepoint Hub. Las organizaciones que caigan en la trampa de esa 'migración de procesos de negocio' no podrán dejar en mucho tiempo la cadena de procesos de Microsoft. Podrían también tener que firmar un contrato de arrendamiento con Microsoft durante al menos los próximos quince años. Deberían asegurarse de que el acuerdo cubra el gasto de las ocho aplicaciones de escritorio y nueve sistemas del lado del servidor necesarios para expandir y mejorar esos pro-*

cesos de negocio altamente productivos, pero encadenados a EOOXML".

Las herramientas de la Fundación son las únicas de entre las que hasta ahora se han presentado al público que prometen total interoperabilidad entre Microsoft Office (y sus formatos binarios) y las aplicaciones que soportan ODF. Los desarrolladores de la Fundación empezaron con la última documentación pública de las APIs de conversión de ficheros de Microsoft Office antes de que Microsoft decidiera que todas las futuras mejoras serían secretas y así lograron deducir cuál es la naturaleza de las etiquetas secretas. Nuestro *plugin* da Vinci añade soporte nativo para ODF a Microsoft Office, algo que ninguna de las otras herramientas vendedoras ofrecerá. Microsoft Office puede de esta forma ser usado en proceso por lotes como una "bomba" que permita migrar con una incomparable fidelidad documentos en formatos heredados de fichero de Microsoft a formato OpenDocument.

El principal punto de la estrategia de la Fundación es el siguiente: impulsar la creación de cadenas alternativas de proceso de información ODF que puedan competir con las cadenas dominadas por EOOXML. Los procesos de negocio vinculados a MSOffice van a migrar a *hubs* estilo E/S altamente productivos. Nada va a parar o alterar esta migración. La única cuestión es ¿estarán preparados esos *hubs* para EOOXML o para ODF? Así pues no tiene sentido interrumpir los actuales flujos de trabajo y servicios para reescribir esos procesos a alternativas de escritorio listas para ODF.

Lo que tiene sentido es, en primer lugar, meter esos procesos de negocio vinculados a MSOffice en ODF. Meter esos documentos y flujos de documentos y datos en ODF. Para eso está da Vinci. En segundo, migrar esos procesos de negocio vinculados a MSOffice a *hubs* preparados para ODF. Éste es el principal propósito de nuestro ODF Infoset Engine - API; un servidor ligero y un motor en el lado del dispositivo que puede automatizar aplicaciones ODF.

La total fidelidad de esas migraciones sólo puede ser factible debido a que ODF fue diseñado de manera intencionada para facilitar la interoperabilidad e incluye características diseñadas específicamente para interoperar con Microsoft Office. ODF implementa un sistema en el cual los elementos y atributos no reconocidos o extraños son preservados por una aplicación que cumple los requisitos. Ver *OpenDocument specification*, sección 1.5., disponible en <<http://develop.opendocumentfellowship.org/spec/>> (pero nótese que donde dice "debería" y "puede" en esa sección se espera que diga "debe" en la versión 1.2, y que está

previsto que se refine más todavía la interoperabilidad en dicha especificación.)

Estos requerimientos (generalmente denominados por la comunidad de desarrollo de ODF como *foreign elements* y *unknown elements*, o simplemente como *Microsoft tags*) ponen en duda la afirmación de Microsoft de que es necesario un estándar separado para asegurar la compatibilidad con esos miles de millones de ficheros binarios de Microsoft Office heredados.

El *plugin* da Vinci ha llegado a un nivel de desarrollo que va mucho más allá de la etapa de pruebas y ha sido aceptado por multitud de departamentos de administraciones públicas de todo el mundo, incluyendo una demostración en Europa a la que asistieron directivos de Microsoft, quiénes justo al día siguiente anunciaron públicamente su deficiente proyecto ODF XSL Transformer, una acción poco consecuente para una compañía que pretende ser promotora de la interoperabilidad del software.

5. El acuerdo Novell-Microsoft plantea problemas antimonopolio

Cuando se anunció el acuerdo Novell-Microsoft, y hasta ahora, ambas empresas proclamaron que los usuarios estaban demandando interoperabilidad entre las aplicaciones de Microsoft y ODF. Eso, y la necesidad de compartir la tecnología virtualizada de sistema operativo, son las principales justificaciones para el acuerdo, según ambas empresas. Sin embargo es también incuestionable que las dos empresas se dieron cuenta de que su mercado no proporcionaría lo que el mercado requiere en cuanto a interoperabilidad.

En lugar de eso, el acuerdo consolida la no interoperabilidad durante cinco años más, reduciendo en la práctica el nivel de fidelidad que se puede conseguir hoy con los filtros de importación-exportación de OpenOffice.org al desviar a los usuarios desde conversiones de formatos binarios hacia herramientas XSL Transformation que son aún menos fiables.

Cualquier discusión honesta sobre el acuerdo Novell-Microsoft debe comenzar preguntando a ambas compañías: [i] por qué se niegan a proporcionar lo que el mercado pide; y [ii] por qué han sellado esa negativa conjunta mediante un contrato vinculante. Ambas empresas aparentemente son conscientes de que es posible una interoperabilidad total. Pero el acuerdo Novell-Microsoft hasta ahora parece menos un "acuerdo para la interoperabilidad" y bastante más un acuerdo prohibido de eliminación de la competencia y asignación ilegal del mercado de software de productividad de oficina, una forma de restricción del comercio contemplada por la Ley Sherman⁴ de los EE.UU.

Según ese acuerdo, Novell se queda con la cuota de mercado de la producción de software de productividad de oficina basado en OpenDocument y Microsoft se queda con la cuota del mercado correspondiente al software basado en EOOXML. La aparente conspiración protege de sus competidores la cuota de mercado asignada a cada una de las partes mediante una patente espesa e indefinida aceptada públicamente por ambas compañías en forma de pacto para que ninguna demande a los consumidores de pago de la otra. Estos tan publicitados pactos de no litigación amenazan implícitamente con costosos procesos judiciales por infracción de patente a cualquiera que use otros productos.

Que Novell sabe que es posible una interoperabilidad mucho mayor está ampliamente demostrado por el hecho de que dos días antes de que el acuerdo fuera firmado y anunciado, fichó al máximo responsable tecnológico de la OpenDocument Foundation, Florian Reuter, un experto reconocido mundialmente en la conversión y tratamiento de documentos. Novell sabía indudablemente, antes de firmar con Microsoft, en qué estaba trabajando Reuter. Además, Microsoft no ocultaba que podía proporcionar soporte total para OpenDocument en Microsoft Office.

El anterior Secretario de Administración y Finanzas del Estado de Massachusetts, Eric Kriss explicó que técnicos de Microsoft le dijeron que sería 'trivial' añadir soporte ODF al nuevo Office 2007. La resistencia a hacerlo proviene del lado de la empresa proveedora.

Así pues resulta que Novell y Microsoft sabían que se podía conseguir la interoperabilidad total y sabían que ésta era un requerimiento del mercado, pero conspiraron para asegurarse de que a los usuarios de software no se les diese la total interoperabilidad que demandaban, haciendo uso intencionadamente de herramientas inadecuadas de transformación XSL a manera de "coartada" antimonopolio en un acuerdo para mantener elementos software separados y no interoperables y no competir en un mercado repartido entre ambas empresas. Admiten que el conversor Microsoft-Clever Age-Novell EOOXML-to-ODF no alcanzará la interoperabilidad total y por tanto no es lo que mercado requiere. Un acuerdo para no proporcionar lo que pide el mercado con el fin de repartir un mercado entre dos empresas es descaradamente anticompetitivo.

Novell todavía podría salvar su maltrecha imagen entre la comunidad de software libre y abierto apoyando de manera sólida (y públicamente) las soluciones de la OpenDocument Foundation. De este modo, Novell podría también evitar una predecible ola de demandas antimonopolio tanto en EE.UU.

como en Europa a las que tiene menos probabilidades de sobrevivir que Microsoft.

Novell al fin y al cabo tiene unos abundantes y recientes fondos que atraerán a los tiburones. Pero independientemente de lo que haga, Novell debería dejar de vender su acuerdo con Microsoft como un "acuerdo de interoperabilidad". La verdadera base del acuerdo es embarzosamente transparente y la compañía sólo puede perder más credibilidad si continúa con esta farsa.

Debería recordarse que en un proceso de negocio un formato de fichero es en cada uno de sus bits tanto un protocolo de comunicaciones como un método de almacenar información. La Dirección General de la Competencia de la Comisión Europea ya había decidido anteriormente que el rechazo de Microsoft a revelar sus protocolos de comunicaciones para Windows y servidores Windows a sus competidores era una violación de la libre competencia. Pero la esa Dirección General no limitó su decisión sólo a los protocolos de Windows; ordenó a Microsoft abstenerse "de cualquier acto o conducta... con efecto u objeto equivalente", un tema planteado por el Comité Europeo para los Sistemas Interoperables cuando éste presentó su queja a la Dirección General de la Competencia debido al rechazo de Microsoft a dar soporte a ODF y a su negativa a desvelar las especificaciones para sus formatos de ficheros binarios Office.

Como Novell fue una de las empresas que instigó y tomó parte en el litigio antimonopolio europeo contra Windows, parece algo más que incongruente para Novell convertirse posteriormente en un co-conspirador en la apuesta de Microsoft para usar el secreto de sus formatos de fichero Office y las APIs de conversión de ficheros, APIs con el fin de frustrar la total interoperabilidad con el software que da soporte a ODF.

Esto es especialmente cierto en un acuerdo que pretende también compartir tecnologías para la virtualización del sistema operativo de ambas empresas. ¿Qué hay de bueno en la virtualización de los sistemas operativos si las aplicaciones de negocio que se ejecutan bajo esos sistemas operativos concurrentes no pueden interoperar? ¿No es éste un "efecto u objeto equivalente" al de la negativa de Microsoft de revelar sus protocolos de comunicaciones Windows? De abierto EOOXML sólo tiene el nombre.

Novell conoce perfectamente bien lo que la ley tiene que decir al respecto. De hecho, Microsoft pagó a Novell 536 millones de dólares para que dejase de participar en el caso antimonopolio de la Comisión Europea. Después de haber abogado con éxito para que se obtuviese el mandato judicial,

Novell está en una débil posición para oponerse a su implementación. El argumento de que "nosotros no nos podíamos permitir rechazar un segundo cheque" no es una defensa fuerte para un caso de conspiración antimonopolio, donde un acusado es responsable no sólo de sus actos, sino también de los actos y omisiones de quienes conspiraron junto a él.

6. Conclusión: la guerra fría de XML en un cambiante mercado de software

Para comprender completamente el bloqueo de proveedor y el vínculo legal que Microsoft y Novell están ideando, es importante entender cómo la adopción de EOOXML de aquí a uno o dos años por parte de la ISO/IEC produciría una ampliación legalmente sancionada del monopolio de Microsoft en los formatos de documentos de oficina. A través de su acuerdo de tecnología compartida con Novell y su elaborado mensaje sobre la 'interoperabilidad', Microsoft busca audazmente reafirmar sus viejos bloqueos propietarios adornando su comportamiento con el lenguaje 'abierto' que está hoy de moda.

El problema legal fundamental de la adopción de EOOXML por la ISO como un estándar internacional es que haría que su uso fuera obligatorio en muchas situaciones, tanto en el sector privado como en el gubernamental. Por la misma razón, su uso estaría prohibido en muchas de esas mismas situaciones si no fuera aprobado por la ISO. (Ver el artículo 2 del Acuerdo sobre barreras técnicas al comercio y el artículo VI del Acuerdo sobre compras de las Administraciones Públicas — ambos de la Organización Mundial del Comercio, OMC). Estos tratados pretenden, entre otras cosas, estimular la competencia mediante la promoción del desarrollo de estándares abiertos con valor legal, eliminar la multiplicación de estándares donde haya solo uno que sea suficiente y exigir el uso de los estándares aprobados. Así pues, la capacidad de desarrolladores no Microsoft para implementar EOOXML es un asunto crucial para evaluar la adecuación de EOOXML como candidato a convertirse en un estándar internacional.

Pero para comprender plenamente la guerra desencadenada en torno a OpenDocument y EOOXML, se debe primero entender que las reglas básicas del mercado de software de oficina se hallan en un estado de cambio continuo. Vivimos en la era de Internet. La era del acceso e intercambio universal. La era de la conectividad universal y de la computación colaborativa. Las cosas están cambiando continuamente. En los tiempos que estamos dejando atrás, el software de oficina fue diseñado como un *punto de llegada*. El camino para alcanzar la interoperabilidad significaba que cada uno de los usuarios de una oficina, y quienes intercambiaban fiche-

ros con esa oficina, utilizaran el mismo software. Los programas de los diferentes proveedores usaban formatos de fichero diferentes e incompatibles. Intercambios y flujos de información estaban totalmente vinculados a una API. Era una situación similar a la Torre de Babel que en último término desembocaba en que un único proveedor de software — Microsoft Office — alcanzase el monopolio principalmente mediante una combinación de incompatibilidades de los formatos de ficheros y la venta conjunta de su software con los nuevos ordenadores.

Pero aunque Microsoft alcanzó el dominio del mercado de los paquetes de oficina, diferentes fuerzas estaban trabajando para impedir su predominio. Uno de estos factores era el auge de una red basada en estándares abiertos como son Internet y las redes ubicuas. Otro factor importante era la creciente crisis de complejidad: "En las últimas cuatro décadas, las arquitecturas de software han intentado lidiar con los crecientes niveles de complejidad del software. Pero el nivel de complejidad no para de crecer y las arquitecturas tradicionales parecen estar llegando al límite de sus posibilidades de solucionar el problema. Al mismo tiempo, persisten las tradicionales necesidades de las organizaciones de TI: la necesidad de responder de forma rápida a los nuevos requerimientos de negocio, la necesidad de reducir continuamente el coste de las TI de la organización, y la capacidad de absorber e integrar nuevos socios de negocio y nuevos grupos de clientes, por citar algunos. Como industria, hemos pasado por muchas arquitecturas de computación diseñadas para permitir un proceso totalmente distribuido, lenguajes de programación diseñados para ser utilizados en cualquier plataforma, tiempos de implementación muy reducidos y una mirada de productos de conectividad diseñados para permitir una mejor y más rápida integración de las aplicaciones. Sin embargo, la solución integral sigue escapándonosos.

Ahora se pueden encontrar entornos más complejos. Los sistemas heredados deben ser reutilizados en lugar de substituidos, porque con unos presupuestos cada vez más ajustados la substitución tiene un coste prohibitivo. Vemos que el acceso barato y ubicuo a Internet ha creado las posibilidades de modelos de negocio completamente nuevos, que deben por lo menos ser evaluados porque ya lo está haciendo la competencia. El crecimiento por fusión o adquisición se ha convertido en algo habitual, de modo que organizaciones, aplicaciones e infraestructuras de TI deben ser integradas y absorbidas. En un entorno de semejante complejidad, las soluciones puntuales sólo acrecientan el problema y nunca nos harán salir del agujero. Deben desarrollarse sistemas en los que la heterogeneidad sea algo fundamental para el entorno, porque se deberán permitir

el ajuste entre una gran variedad de hardware, sistemas operativos, middleware, lenguajes y almacenamiento de datos. El efecto acumulativo de décadas de crecimiento y evolución ha producido una enorme complejidad. Con todos esos retos de negocio para las TI, no es de extrañar que la integración de aplicaciones sea una de las primeras prioridades de muchos CIOs" (Ver "Migrating to a service-oriented architecture" de Kishore Channabasavaiah et al., <<http://www-128.ibm.com/developerworks/library/ws-migratesoa/>>).

Para enfrentarse al desafío planteado por esa crisis de complejidad, empezó a surgir una nueva *Arquitectura Orientada a Servicios* (SOA). La SOA está en gran parte construida en torno a XML, un lenguaje extensible de marcado abierto y legible, usado como bloque fundamental para futuras expansiones. Al igual que ocurre con los datos almacenados en importantes formatos de fichero heredados, la SOA requiere infinitas repeticiones del siguiente flujo de trabajo: [i] identificar la posición y forma de los datos especificados por la aplicación solicitante; [ii] convertir los datos desde formatos de fichero heredados a un formato XML válido; [iii] extraer de forma programada aquellas porciones de datos que han de ser readaptadas; [iv] introducir los datos en un proceso de transformación XML; [v] extraer los datos en el formato XML requerido; y [vi] serializar esos datos a la aplicación especificada en el flujo de trabajo para su posterior proceso.

Obsérvese que en ese supersimplificado flujo de trabajo las aplicaciones son puntos de paso, es decir enrutadores de información más que puntos de llegada. Aplicaciones diseñadas en los días de la "red pedestre" (**N. del T.:** *sneaker net* o red construida mediante el intercambio manual de soportes digitales), como es el caso de Microsoft Office, tienen pronunciadas desventajas. La migración sin fallos de datos entre formatos es un elemento integral de un proceso SOA. Los flujos de trabajo que incorporan estos pasos son conocidos comúnmente como *procesos de negocio*.

Microsoft es capaz de predecir las tendencias de la industria tan bien como cualquier otro proveedor. Por ello no es sorprendente que Microsoft esté desarrollando su propio paquete de software propietario de procesos de negocio. Ese paquete, la infame cadena de proceso de la información, usa EOOXML no sólo como un formato de fichero, sino también como un protocolo de comunicaciones entre las diversas aplicaciones. Por lo tanto, EOOXML es bastante más que solamente un formato de fichero para un paquete de aplicaciones de oficina. Al igual que OpenDocument, está siendo diseñado como una herramienta de interoperabilidad para

procesos de negocio dentro de soluciones tales como SOA. Como se verá a continuación, la diferencia es que EOOXML es una especificación cerrada y propietaria. Por el contrario, ODF es completamente abierta.

Tampoco es una casualidad que la guerra entre los partidarios de OpenDocument XML y el XML de Microsoft Office 2003 (un antecesor de EOOXML) apareciese primero ante el público durante el diseño de la SOA tanto a través del Informe Valoris de la Unión Europea como del Estado de Massachusetts con su Modelo de Referencia Técnico para Empresas de la División de Tecnologías de la Información (ITD). El proceso de diseño de esa arquitectura creó la necesidad de elegir qué formato de fichero XML sería el formato de destino para los formatos de ficheros binarios heredados de Microsoft. Por diversas razones. La ITD de Massachusetts eligió OpenDocument como ese formato de fichero. Naturalmente la SOA es sólo una parte de la historia. Un creciente número de webs y de aplicaciones distribuidas de web da soporte a OpenDocument, incluyendo el modelo SaaS (*Software as a Service*) y la nueva generación de la Web, la Web 2.0.

Aquí nos encontramos. Es hora de tomar una decisión. Microsoft está ofreciendo una cadena de proceso de la información basada en EOOXML muy convincente y con muchos elementos que supone un gran impulso para la base monopolista ya instalada de aplicaciones Microsoft Office, procesos de negocio vinculados y BoBs cerrados heredados. Se trata sin duda de un objetivo de negocio diseñado exactamente para extender el monopolio desde los ordenadores de escritorio a los servidores, dispositivos y demás. Asusta, por decirlo suavemente. No nos extraña en absoluto. Increíblemente audaz.

Por otra parte, ODF está diseñado y destinado a ser un formato universal de ficheros independiente de aplicaciones, plataformas, necesidades de archivo y avances en las TI aún desconocidos. Es un formato universal de ficheros al servicio de las necesidades de dominios de información tan diversos, y que sin embargo todavía están solicitando una conectividad y un intercambio interoperables, como son los entornos productivos de ordenadores de sobremesa, la publicación empresarial, los sistemas de gestión contenidos y de archivos, SaaS, SOA, la Web 2.0 y más.

EOOXML incumple decididamente lo que XML promete: la fácil transformación universal y la interoperabilidad intergeneracional. Pero no tema, la perfecta interoperabilidad entre ODF y los miles de millones de documentos binarios está ya disponible en ODF 1.2. Y luego está el prometedora potencial de la cadena de proceso de información ODF del *plugin* de la Funda-

ción Da Vinci (su *plugin* de ODF para Microsoft Office) y su API ODF InfoSet.

ODF está listo. Le lleva dos años de adelanto a EOOXML en el proceso de estandarización de ISO. ¡Que comience la batalla!

Agradecimientos

Marbux, un abogado retirado, miembro voluntario de la OpenDocument Fellowship, contribuyó realizando un análisis legal del artículo.

Referencias

¹ La interoperabilidad conlleva normalmente un sentido de ausencia total de barreras. Ver, por ejemplo, la definición que da ISO/IEC 2382-01, tal como se cita en la versión inglesa de Wikipedia ("*La capacidad de comunicar, ejecutar programas o transferir datos entre varias unidades funcionales, de forma que se requiera poco o nulo conocimiento por parte del usuario sobre las características específicas de estas unidades*").

² Para una crítica más detallada de EOOXML dentro del contexto de una SOA, ver el artículo "*IBM's potential MS-Office killer to roll out by year's end*" de Gary Edwards, disponible en <<http://talkback.zdnet.com/5208-10532-0.html?forumID=1&threadID=13561&messageID=273162&start=-4>>.

³ "*Is Open XML a one way specification for most people?*", disponible en <<http://sutor.com/newsite/blog-open/?p=1145>>. Rob Weir, de IBM, ha publicado después una replica satírica al artículo de Sutor (ver "*How to Write a Standard (If you Must)*"), en <<http://www.robweir.com/blog/2006/12/how-to-write-standard-if-you-must.html>>).

⁴ Caso Palmer contra BRG de Georgia, Inc., 498 U.S. 46, 50 (1990): "*tales acuerdos van contra la libre competencia independientemente de si las partes dividen un mercado dentro del cual ambas hacen negocio o de si se limitan a reservar un mercado para una y otro para la otra*". Ver también la Sección 1 del Código de Comercio de los EE.UU. en <http://caselaw.lp.findlaw.com/cascode/uscodes/15/chapters/1/sections/section_1.html>: "*Se declara ilegal todo contrato, o acuerdo en forma de compromiso o en cualquier otra forma, o conspiración que suponga una restricción del intercambio o comercio entre varios Estados o con naciones extranjeras. Quien firme un contrato o participen en cualquier acuerdo o conspiración que este código declare ilegal será considerado culpable de delito y si es condenado será castigado con una multa no superior a 10.000.000\$, si el implicado es una persona jurídica, y si es otro tipo de persona, de 350.000\$ o con pena de prisión no superior a tres años o con ambas penas, según estime el tribunal*". Los tribunales han impuesto la condición de que la restricción del comercio debe ser "no razonable". En aplicación de la Ley Sherman, para decidir si una restricción del comercio no es razonable hay que "*basarse (1) en la naturaleza o carácter del contrato o (2) en circunstancias que den lugar a la inferencia o presunción de que se pretende restringir el comercio o elevar los precios*", según la sentencia del Tribunal Supremo de los EE.UU. en el caso NCAA contra. Board of Regents of Univ. of Okla., 468 U.S. 85, 103 (1984), citando el caso National Society of Professional Engineers contra United States, 435 U.S. 679, 692 (1978); la sentencia está disponible en <<http://caselaw.lp.findlaw.com/scripts/getcase.pl?court=US&vol=468&invol=85>>.