

XML y derivados: Estándares para un paradigma tecnológico orientado a servicios

Francisco Ruiz González
Grupo Alarcos
Departamento de Informática
Universidad de Castilla-La Mancha

octubre-2003

XML y derivados

Estructura de la Presentación

- Orígenes del XML
 - SGML
 - Problemas de HTML y SGML
 - Objetivos iniciales
- Características del XML
- Tipos de normas XML
- Extensiones
 - Estructuración de documentos
 - Enlaces y direccionamiento
 - Transformación y presentación
 - Consultas
 - Programación
 - Otras
- Lenguajes horizontales
 - Seguridad
 - Servicios web
 - Metadatos y conocimiento
 - Formularios
 - Otros
- Lenguajes verticales
- Opciones de futuro
- Conclusiones

Orígenes del XML - SGML

- XML (eXtensible Markup Language)
 - se plantea a mediados de los 90 por la necesidad de poder definir y manejar estructura interna en los documentos HTML.
 - pero está basado en un estándar bastante anterior: SGML (*Standard Generalized Markup Language*)
 - con orígenes en los años 60 (IBM, GML).
 - estándar ISO 8879 en 1986.
 - XML vs SGML
 - regla del 80/20
 - 80% de la funcionalidad
 - 20% de la complejidad.
 - XML ≈ SGML light



Orígenes del XML Problemas de HTML y SGML

HTML

- Problemas
 - Extensibilidad
 - Estructura lógica
 - Intercambio de datos
 - Reutilización de datos
 - No es orientado a objetos
 - Gestión de enlaces rotos
- Beneficios
 - Simple
 - No requiere herramientas especiales
 - Extendido
 - Papel importante en auge de Internet.

SGML

- Problemas
 - Complejidad
- Beneficios
 - Reutilización de datos
 - Longevidad de la información
 - Integridad y mayor control sobre los datos.
 - Permite compartir información
 - Portable
 - Flexible

Orígenes del XML

Objetivos iniciales

- Cuando se comienza a trabajar en 1996, los objetivos de diseño de XML eran:
 - debe ser utilizable directamente sobre Internet (navegadores).
 - debe soportar una amplia variedad de aplicaciones.
 - debe ser compatible con SGML.
 - debe ser fácil procesar documentos XML.
 - el número de características opcionales debe ser mínimo, y a ser posible cero.
 - los documentos XML deben ser legibles por un humano y razonablemente claros.
 - la especificación de XML debe ser formal y concisa.
 - los documentos XML deben ser fáciles de crear.
 - la brevedad en la marcación tiene poca importancia.

Características del XML (i)

- XML es una idea simple, no nueva, pero tremendamente útil porque ...
 - ha llegado en un momento adecuado para sacar partido de diversas tecnologías (Internet, Web, ...)
 - y poder abordar de nuevas maneras, más adecuadas y potentes, problemas importantes:
 - Integración de datos estructurados (tablas relacionales) y poco estructurados (documentos).
 - Aportar significado a la web (web semántica).
 - Integración de sistemas de información basados en tecnologías diferentes.
 - Hacer una web orientada a las aplicaciones (*web services*) en vez de una web orientada sólo a la interacción con personas (HTML).

Características del XML (ii)

- XML es
 - Un lenguaje de marcas (etiquetas delimitadas)
 - para definir nuevos lenguajes (un **metalenguaje**)
- Con las siguientes características principales:
 - Versátil: separa contenido, estructura y presentación
 - Extensible: se pueden definir nuevas etiquetas
 - Estructurado: se pueden modelar datos a cualquier nivel de complejidad
 - Validable: cada documento se puede validar frente a un DTD/Schema, o en su defecto, se puede declarar bien formado.
 - Abierto: independiente de plataformas, empresas, lenguajes de programación o entornos de desarrollo.
 - Sencillo: fácil de aprender y de usar.

Características del XML (iii)

- Otras características adicionales de XML son:
 - Independencia del medio, pudiendo publicar contenidos en múltiples formatos.
 - Los documentos XML son fácilmente procesables y compatibles en Internet.
 - Permite validación de los documentos.
 - Permite composición de los documentos.
 - Puede ser un contenedor de datos. Utilizando DTD o XML Schemas se puede representar eficientemente cualquier dato de forma que puede ser leído por personas o aplicaciones.
 - Internacional: utiliza el conjunto de caracteres UNICODE.
 - Ayuda a descongestionar Internet, ya que gran parte del procesamiento se puede hacer en el cliente.
 - XML no es compatible con HTML, pero los documentos HTMLv4.0 son fácilmente convertibles a XML.

Tipos de normas XML

- Extensiones del propio XML
 - Amplían las capacidades del XML original, no son un lenguaje XML en sentido estricto.
 - Se definen de forma similar al XML original, como un subconjunto de SGML.
 - Todas ellas deben ser compatibles entre sí.
- Lenguajes XML (Aplicaciones XML)
 - Lenguajes definidos en base a XML y sus extensiones.
 - Cada lenguaje se define mediante una gramática que consiste en un tipo de documento XML (DTD o XSD).
 - Pueden ser:
 - Horizontales: resuelven cierta funcionalidad que es útil en general.
 - SOAP (*Simple Object Access Protocol*)
 - Verticales: dirigidos a un sector o utilidad particular.
 - CML (*Chemical Markup Language*)

Extensiones

- El éxito de XML ha originado que se demanden nuevas funcionalidades, que se abordan definiendo extensiones adicionales para:
 - Estructurar documentos (*XML Schema*).
 - Enlaces y direccionamiento (*XPath, XLink, XPointer*).
 - Transformación y presentación (familia *XSL, CSS2*).
 - Consultas (*XQuery*).
 - Programación (*DOM, SAX*).
 - Otros (*Namespaces, XInclude, XBase, ...*).

Extensiones

Estructuración de documentos (i)

- La gramática de los lenguajes XML, es decir, la estructura y elementos permitidos en los documentos XML, se define mediante
 - DTD (Document Type Definition)
 - Documento ASCII plano que especifica tanto los elementos que forman un tipo de documento dado, como las relaciones que se dan entre ellos.
 - XSD (XML Schema Definition)
 - Mejoran los DTD's porque están escritos en XML y permiten nuevas características:
 - definir tipos de datos,
 - utilizar espacios de nombre
 - definir intervalos de valores para los atributos y elementos.
 - características OO ...

Extensiones

Estructuración de documentos (ii)

- Un lenguaje XML (DTD/XSD) representa un modelo de datos jerárquico
 - Estructura los datos de acuerdo a un determinado "esquema semántico".
- Estos lenguajes se definen especificando los elementos y atributos permitidos.
 - Esta especificación se realiza mediante reglas gramaticales.
 - Un conjunto concreto y bien formado de tales reglas forman un esquema XML (representado por un DTD o un XSD).
 - Un esquema XML define un conjunto coherente de documentos, es decir un tipo de documentos.
- Ejemplo:
 - XHTML es el lenguaje HTML reformulado como aplicación XML.

Extensiones

Estructuración de documentos (iii)

- Ejemplo de XML Schema

```
<schema targetNamespace="http://www.bd.es/schema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:bd="http://www.bd.es/schema" >
  <element name="Articulo" type="bd:tArticulo" />
  <complexType name="tArticulo">
    <element name="Cabecera" type="bd:tCabecera"/>
    <element name="Cuerpo" type="bd:tCuerpo"/>
    <element name="Final" type="bd:tFinal"/>
  </complexType>
  <complexType name="tCabecera">
    <element name="Titulo" type="string"/>
    <element name="Autor" type="string"/>
  </complexType>
  .....
```

Extensiones

Enlaces y direccionamiento – XPath (i)

- XML Path Language (XPath) es un lenguaje declarativo para localizar nodos y fragmentos (texto, elementos, atributos ...) en el árbol de un documento XML.
- Es utilizado por otras normas para
 - Direccionamiento (XLink, XPointer y XSLT)
 - "Pattern matching" (XSLT y XQuery)
- Se basa en el XPath Data Model:
 - Un documento XML se representa como un árbol jerárquico con siete tipos de nodos (raíz, elemento, texto, atributo, espacio de nombres, instrucción de procesamiento y comentario).
- Conceptos importantes:
 - Caminos de localización / libro / capítulo / párrafo
 - Predicados

Extensiones

Enlaces y direccionamiento – XPath (ii)

- Ejemplos XPath:
 - Seleccionar nombres de ingredientes de receta que se utiliza media taza:
`//ingrediente[@cantidad='0.5' and @unidad=taza]/@nombre`
 - Seleccionar todos los capítulos públicos que tengan algún párrafo que contenga algún elemento con atributo href:
`//capitulo[parrafo/*[@href]][@public='si']`
 - Seleccionar todos los capítulos públicos que tengan algún párrafo importante o un apéndice:
`//capitulo[parrafo/[@importante='si']]//apendice`

Extensiones

Enlaces y direccionamiento – XPointer

- XPointer describe cómo se puede apuntar a un lugar específico de un determinado documento XML.
- Es una extensión de XPath que permite asociar a una dirección URI con una expresión XPath con algunas propiedades extras.

`http://www.sitio.es/doc.xml#xpointer\(/libro/capitulo\[@public\]\)`

- Nuevos conceptos:
 - Puntos (para trabajar a nivel de caracteres).
 - Rangos (para trabajar a nivel de palabras, subcampos de fechas, ..).

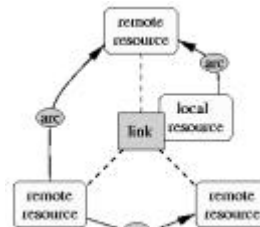
Extensiones

Enlaces y direccionamiento – XLink (i)

- XLink (*XML Linking Language*) define la forma en la que los documentos XML se pueden relacionar entre sí definiendo nuevos tipos de elementos XML que representan enlaces (links).
 - Utiliza XPointer para localizar recursos.
 - Características especiales:
 - Uso de "alias".
 - Asociaciones entre más de 2 recursos (enlaces multidireccionales).
 - Un origen y varios destinos.
 - Enlaces agregados (varios orígenes, un sólo destino)
 - Asociar metadatos a un enlace.
 - Expresar enlaces que residen fuera de los recursos enlazados.

Extensiones

Enlaces y direccionamiento – XLink (ii)



- Existen 2 tipos de enlaces:
 - Simples

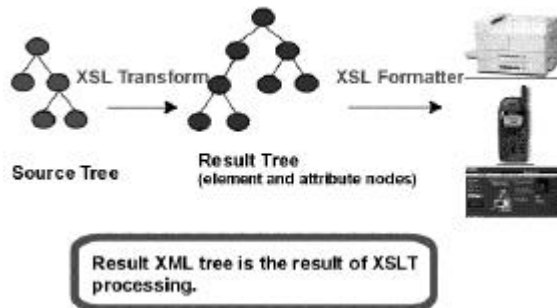
```
<AUTOR xlink:href="autores.xml#juan" xlink:show="new">  
<NOMBRE>Juan Primero Segundo</NOMBRE>  
</AUTOR>
```
 - Extendidos

```
<EDITOR_AUTOR xlink:extended>  
<xlink:locator href="#ana" id="editor"/>  
<xlink:locator href="autores.xml#juan" id="autor"/>  
<xlink:arc from="editor" to="autor" show="replace"/>  
</EDITOR_AUTOR xlink:extended>
```

Extensiones

Transformación y Presentación – XSL (i)

- XSL (e**X**tensible **S**tylesheet **L**anguage) no sólo permite definir el estilo a aplicar a cada elemento XML. También es un lenguaje de programación para transformar documentos XML.



Extensiones

Transformación y Presentación – XSL (ii)

- El resultado puede ser un documento HTML, WML (para WAP), texto plano, RTF, PDF, o incluso otro documento XML.
- Utiliza XPath para referir partes de documentos XML.
- Anteriormente XSL eran 2 estándares separados:
 - XSL Transformations (XSLT): lenguaje de programación para transformar documentos XML.
 - XSL Formatting Objects (XSL-FO): vocabulario para definir cómo presentar un documento XML.
- El vocabulario de objetos de formato (elementos de tipo "fo:") representa el conjunto de abstracciones tipográficas disponibles.

Extensiones

Transformación y Presentación – XSL (iii)

- Una hoja de estilo XSL es una serie de reglas que determinan como va a ocurrir la transformación.
- Cada regla se compone de un patrón de localización (*pattern*) y una plantilla (*template*).

```
<xsl:template match="/">
<HTML>
<BODY>
<xsl:for-each select="/LIBROS/LIBRO">
Título:
<xsl:value-of select="TITULO"/><BR/>
Autor:
<xsl:value-of select="AUTOR"/><BR/>
Precio:
<xsl:value-of select="PRECIO"/> pesetas<BR/>
</xsl:for-each>
</BODY>
</HTML>
</xsl:template>
```

Extensiones

Transformación y Presentación – XSL (iv)

- Ventajas de usar hojas de estilo XML:
 - Centralizar la forma de presentación (formato)
 - Separar estructura ↔ contenido
 - Reutilización de datos
 - Diferentes formatos de salida para los mismos datos
 - Unificar el estilo de presentación



Extensiones

Transformación y Presentación – CSS2

- CSS2 (*Cascading Style Sheets level 2*) es una nueva versión del lenguaje de hoja de estilos de HTML para poder ser usado con XML.
- Más sencillo que XSL, es conveniente usarlo siempre que no sea necesaria una transformación.
- Permite describir el formato en el que se desea que aparezcan las entidades definidas en un documento.

```
P {font-family:Verdana; font-size:10 pt}
```

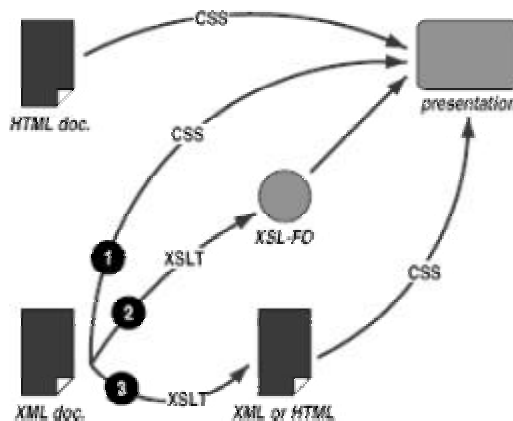
```
TABLE {border:2; font-family:Tahoma; font-size:9 pt}
```

```
H3 {font-family:Comic Sans MS; font-size:12 pt; color:blue}
```

Extensiones

Transformación y Presentación – vs

- XSL vs CSS2:



Extensiones

Consultas – XQuery (i)

- **XQuery** proporciona un modo flexible de consulta para extraer datos de los documentos XML.
 - Los archivos XML pueden ser reales o virtuales, es decir, otras fuentes (hojas de cálculo, ASCII, bases de datos, ...) vistas como datos XML.
 - Se pretende que desempeñe un papel similar al SQL en las BD relacionales: las colecciones de documentos XML podrán ser accedidas como si fueran una base de datos.
 - Esta basado en varias propuestas de lenguajes previas (XML-QL, YATL, Lorel, Quilt).
 - Se ha integrado con XPath (versión 2.0).
 - Existe un cierto solape con XSLT.
 - Una consulta puede referir a más de un documento.

Extensiones

Consultas – XQuery (ii)

- Una consulta es una expresión que:
 - Lee una secuencia de fragmentos XML o valores atómicos y
 - Devuelve una secuencia de fragmentos XML o valores atómicos.
- Los principales tipos de expresiones son:
 - Expresiones XPath, para navegar por los documentos.
 - Constructores de elementos.
 - FLWR (FOR, LET, WHERE, RETURN) para iterar por los elementos de una colección.
 - Condicionales (IF, THEN ELSE) para construir el resultado en base a alguna condición.
 - Con cuantificadores (SOME, ANY) para chequear la existencia de algún elemento que cumpla una condición.
 - Listas a las que se pueden aplicar operadores (UNION, ...) y funciones (AVG,...).

Extensiones

Consultas – XQuery (iii)

- Ejemplo de consulta:
 - Obtener el año y título de todos los libros publicados por Addison-Wesley después de 1991:

```
<bib>
{
  for $b in doc("http://www.bn.com/bib.xml")/bib/book
  where $b/publisher = "Addison-Wesley" and $b/@year > 1991
  return
  <book year="{ $b/@year }">
  { $b/title }
  </book>
}
</bib>
```

Extensiones

Programación

- API's (*Application Program Interface*) que facilitan un conjunto estándar de llamadas a funciones para manipular documentos XML desde programas:
 - **DOM** (*Document Object Model*), del W3C.
 - Orientada a objetos.
 - Facilita el acceso al documento XML completo.
 - Incluye una representación estándar, en memoria, de la estructura de un documento XML y un API para acceder (acceso aleatorio), modificar, eliminar o insertar los elementos y atributos.
 - **SAX** (*Simple API for XML*).
 - Orientado a eventos.
 - El documento se procesa de manera secuencial.
 - Cada elemento XML dispara un evento.
 - Pensado para leer con rapidez documentos XML y reaccionar en función de su contenido.

Extensiones Otros (i)

- **Namespaces** in XML

- Método para cualificar elementos y nombres de atributos de documentos XML, asociándolos con espacios de nombres (*namespaces*) identificados por referencias URI.

```
<x xmlns:edi='http://ecommerce.org/schema'>  
</x>
```

- Sirve para
 - Evitar las colisiones en los nombres de los elementos y atributos.
 - Hacer públicos DTD's, XML Schemas o partes de ellos con fines de reutilización.
 - Ayuda para combinar lenguajes XML.

Extensiones Otros (ii)

- **XBase** establece un mecanismo para utilizar URI's relativos.

```
<...xml:base="http://www.sitio.es/"> <...href="~/yo/dir/index.html" .../>  
equivale a  
http://www.sitio.es/~yo/dir/index.html
```

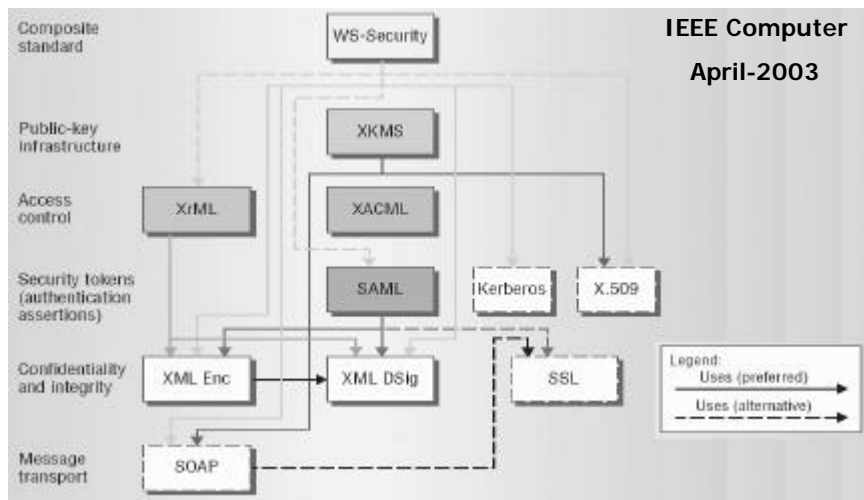
- **XInclude** (*XML Inclusions*) provee un modelo de proceso y una sintaxis para hacer inclusiones.
 - Facilita la reutilización y modularidad.
 - Permite combinar documentos XML, o construir nuevos documentos XML a partir de otros previos.

```
<foo xmlns:xi="http://www.w3.org/2001/XInclude">  
<xi:include href="partedoc.xml"/>  
</foo>
```

Lenguajes Horizontales

- Lenguajes XML que tienen una utilidad de interés general:
 - Seguridad: XML DSig, XML Enc, SAML, XACML, XKMS.
 - Servicios Web: SOAP, WSDL, UDDI.
 - Metadatos y conocimiento: RDF, XMI.
 - Formularios Web: XForms.
 - Otros: SVG, DC.

Lenguajes Horizontales Seguridad (i)



Lenguajes Horizontales Seguridad (ii)

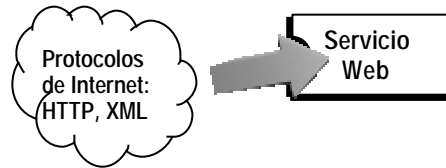
- XML DSig (*XML Signature*)
 - Describe una sintaxis XML para representar asociaciones entre firmas digitales y documentos XML u otros recursos electrónicos.
 - También incluye servicios para integridad, autenticación de mensajes y/o firmantes, para cualquier tipo de datos, localizados dentro del documento XML que incluye la firma o fuera.
- XML Enc (*XML Encryption*)
 - Define un vocabulario XML y reglas de procesamiento para proteger la confidencialidad (mediante encriptación) de documentos XML, partes de documentos o datos no XML.

Lenguajes Horizontales Seguridad (iii)

- SAML (*Security Assertion Markup Language*)
 - Marco de trabajo para intercambiar peticiones/respuestas de información de autenticación y autorización.
- XACML (*eXtensible Access Control Markup Language*)
 - Especificación XML para expresar políticas de control de acceso de grano fino.
- XrML (*eXtensible Rights Markup Language*)
 - Especificación XML para expresar derechos y condiciones (tiempos de expiración etc.) asociados con recursos y servicios digitales.
- XKMS (*XML Key Management Specification*)
 - Define una interfaz de Servicio Web para una infraestructura de clave pública para usar con los protocolos XML DSig y XML Enc.

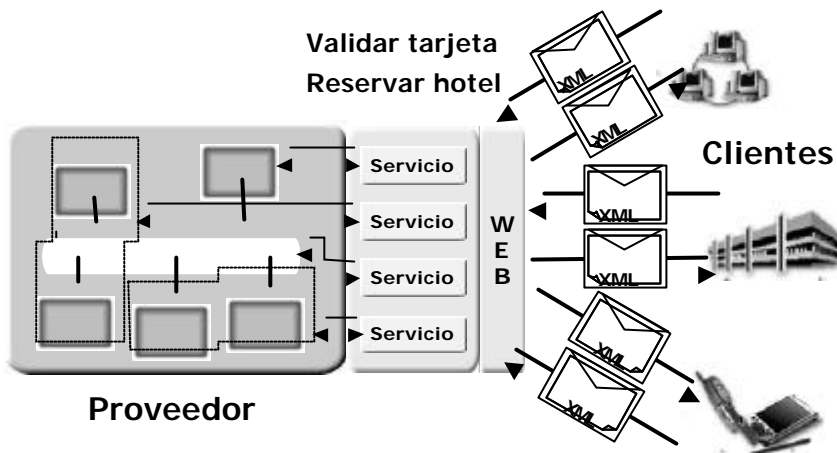
Lenguajes Horizontales Servicios Web (i)

- Tecnologías basadas en el intercambio de mensajes XML que permiten la interacción entre sistemas software mediante la metáfora de "servicio":
 - Un servicio se solicita a través de la web.
 - Los resultados también se obtienen a través de la web.
 - Se tramitan según un modelo, protocolos y formatos estandarizados.
 - Existe independencia del lenguaje de programación, plataforma o dispositivos de presentación.
- Son útiles para:
 - Integración de
 - negocios con terceros.
 - contenidos.
 - plataformas o sistemas.
 - Reutilización de código y servicios.



Lenguajes Horizontales Servicios Web (ii)

- Nuevo enfoque de la web orientada a los servicios



Lenguajes Horizontales

Servicios Web (iii)

- Arquitectura WSA: *Web Services Architecture*
 - Norma W3C que establece un modelo y un contexto para comprender los Servicios Web (WS) y las interrelaciones entre las diversas especificaciones y tecnologías utilizadas.
- Definición de WS:
 - Un WS es un sistema software diseñado para soportar interacciones interoperables máquina-a-máquina a través de una red.
 - Tiene un interfaz descrito en un formato computable (WSDL).
 - Otros sistemas pueden interactuar con el WS en una manera prescrita por su definición, usando mensajes (SOAP) transportados usando HTTP/XML en conjunción con otros estándares relacionados.

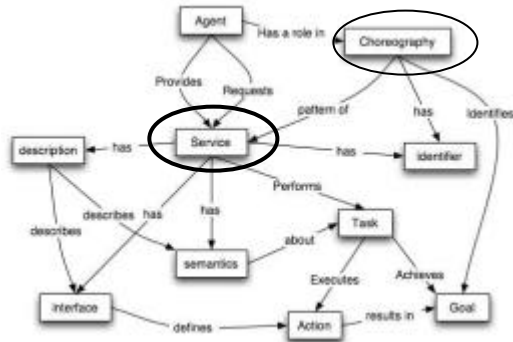
Lenguajes Horizontales

Servicios Web (iv)

- Algunos conceptos:
 - Agente vs Servicio:
 - Agente => Entidad concreta (pieza de software) que envía y recibe mensajes.
 - Servicio => Conjunto abstracto de funcionalidad que es provista.
 - Demandante vs Proveedor:
 - Proveedor => Persona u organización que provee un agente adecuado para implementar un servicio particular.
 - Demandante => Persona u organización que desea usar un determinado servicio de un cierto proveedor.
 - Descripción de Servicios: WSD.
 - Especificación computable del interfaz de un WS.
 - Define formatos de mensajes, tipos de datos, protocolos de transporte y formatos de serialización que deberán usar los agentes demandantes y proveedores.

Lenguajes Horizontales Servicios Web (v)

- WSA define 5 modelos arquitecturales:
 - Orientado a los Mensajes.
 - Orientado a los Servicios.
 - Orientado a los Recursos.
 - De Políticas.
 - De Gestión.



Francisco Ruíz. XML y derivados

39

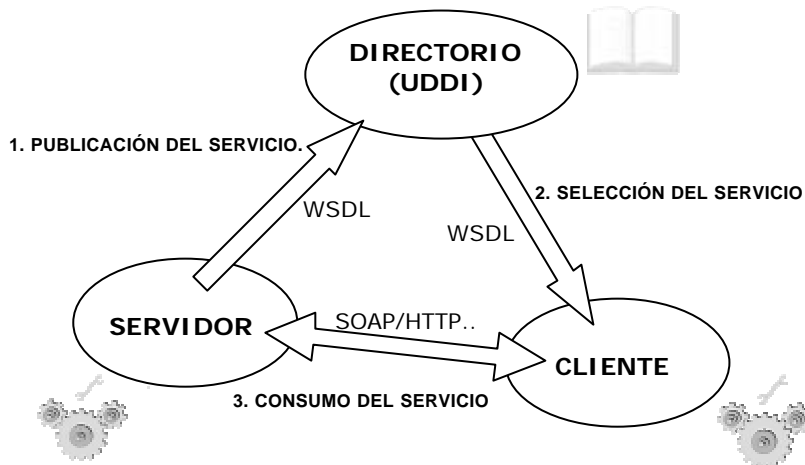
Lenguajes Horizontales Servicios Web (vi)

- Para lograr la comunicación estándar se han definido las especificaciones XML siguientes:
 - **SOAP** (*Simple Object Access Protocol*).
 - Formato de los mensajes intercambiados entre el cliente y el servicio web.
 - **WSDL** (*Web Service Definition Language*)
 - Formato del documento de descripción de un servicio web (contrato).
 - Identifica los métodos, funciones y parámetros necesarios para invocar un determinado servicio.
 - **UDDI** (*Universal Description, Discovery and Integration*)
 - Repositorio de registro y búsqueda de servicios web (directorio, "páginas amarillas").
 - Los servicios están registrados para que los posibles usuarios puedan encontrarlos.

Francisco Ruíz. XML y derivados

40

Lenguajes Horizontales Servicios Web (vii)



Lenguajes Horizontales Servicios Web (viii)

Ejemplo de mensajes SOAP:

- Petición libro

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <catalogo:buscalsbn
      xmlns:catalogo="http://catalogo.org/cat">
      <catalogo:isbn>
        84-9999-9999-99
      </catalogo:isbn>
    </catalogo:buscalsbn>
  </SOAP-ENV:Body >
</SOAP-ENV:Envelope>
```

- Respuesta libro

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body >
    <catalogo:buscalsbnResponse
      xmlns:catalogo="http://catalogo.org/cat">
      <catalogo:titulo>
        El Origen de las Especies
      </catalogo:titulo>
      <catalogo:autor>
        Darwin
      </catalogo:autor>
    </catalogo:buscalsbnResponse>
  </SOAP-ENV:Body >
</SOAP-ENV:Envelope>
```

Lenguajes Horizontales Metadatos (i)

- Los metadatos sirven para
 - Añadir semántica (describir el significado de los datos)
 - Gestión de conocimiento
 - Mayor genericidad en las soluciones (aplicaciones)
- Las 2 propuestas principales son:
 - **RDF** (*Resource Description Framework*), un marco de trabajo que permite la codificación, intercambio y reutilización de metadatos estructurados.
 - **MOF/XMI** (*Meta-Object-Facility, XML Metadata Interchange*) son dos normas OMG para
 - Definición, representación y gestión de metadatos con el paradigma orientado a objetos (MOF).
 - Intercambio de metadatos en entornos distribuidos heterogéneos (XMI).

Lenguajes Horizontales Metadatos (ii)

- Ejemplo RDF
 - Luis es el autor [creator] del recurso www.sitio.es/luis/index

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.sitio.es/luis/index">
    <s:Creator>Luis</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Lenguajes Horizontales Formularios Web

- **XForms** representa una nueva generación de formularios Web basados en XML.
 - Divide los formularios en tres partes:
 - Modelo XForms (qué hace)
 - Datos (qué muestra)
 - Interfaz de usuario (cómo lo muestra)
 - De esta manera permite:
 - Separar datos, lógica y presentación.
 - Reutilización
 - Tipado fuerte (reduciendo los accesos al servidor)
 - Independencia de la unidad física de presentación (pantalla, TV, impresora, scanner).
 - Reducir el uso de lenguajes de Script (lógica avanzada, múltiples formularios por página y viceversa, datos estructurados, ..)
 - Está diseñado para ser utilizado dentro de otros lenguajes XML.



Lenguajes Horizontales Otros

- **SVG** (*Scalable Vector Graphics*)
 - Permite describir gráficos vectoriales bidimensionales, tanto estáticos como animados.
 - Tres tipos de objetos:
 - figuras (líneas rectas y curvas)
 - imágenes
 - texto.
- **DC** (*Dublin Core*).
 - Definir un conjunto de elementos que puedan ser utilizados por los autores para describir sus propios recursos Web.
 - Los 15 elementos definidos están relacionados con
 - El contenido: title, subject, description, source, language, relation, coverage.
 - La propiedad intelectual: creator, publisher, contributor, rights.
 - Instanciación: date, type, format, and identifier.

Lenguajes verticales

- OFX (*Open Financial eXchange*), intercambio de datos financieros.
- CML (*Chemical Markup Language*), descripción de compuestos químicos.
- MML (*Mathematical Markup Language*), descripción de fórmulas matemáticas.
- OSD (*Open Software Distribution*), descripción de paquetes de software.
- TEI (*Text Encoding Initiative*), representación electrónica de textos.
- ebXML, comercio electrónico B2B mediante mensajes XML.
- GML (*Geography Markup Language*), transmisión y almacenamiento de información geográfica (datos espaciales y no espaciales).
- VoiceXML (*Voice eXtensible Markup Language*), representación de voz en XML.
- SMIL (*Synchronized Multimedia Integration Language*), para presentaciones audiovisuales interactivas.

Opciones de futuro (i)

- Los principales riesgos detectados son:
 - Fragmentación: existe más de un lenguaje XML para lo mismo.
 - Ausencia de un modelo subyacente: las especificaciones están basadas en la sintaxis y no en un modelo de datos interoperable.
 - InfoSET.
 - Excesiva dificultad: gran cantidad de normas existentes.

XML es fácil

⌘ ⌘ ⌘

Ingeniero XML

Opciones de futuro (ii)

- Los escenarios posibles en el futuro son:
 - ~~Síndrome del esperanto~~
 - Parecía buena idea pero ... acabó siendo un montón gigantesco de especificaciones de diversos organismos incompatibles entre sí.
 - ~~Vuelta a empezar~~
 - La sencillez inicial se pierde por completo. El uso de las diversas normas requiere, cada vez más, la fuerza bruta (codificación).
 - **Un mundo XML-feliz**
 - Las tecnologías XML maduran, la vorágine actual desaparece y "el mar se calma". Se consigue definir una especificación global general de XML (XML, XPath, XLink, XSL, XQuery).

Opciones de futuro (iii)

- Nuevo paradigma de negocio:
 - Propietario (clásico) vs Libre (reciente)
 - Software como un Servicio
 - SaaS (Software as a Service) *IEEE Computer oct-2003*
 - SoC (Service-oriented Computing) *ACM Communic. oct-2003*
 - Posible gracias a la convergencia entre diversas tecnologías recientes
 - Servicios Web
 - Proceso Software (Flujos de Trabajo)
 - Ontologías y gestión de conocimiento (web semántica)
- Ejemplo:
 - Orquestación y Coreografía de Servicios Web =>
 - Modelado de procesos.
 - Gestión de flujos de trabajo.

Conclusiones

- XML es una idea sencilla pero tremendamente potente.
- Gran éxito actual.
- Nuevas perspectivas para resolver “viejos” problemas.
- Falta de claridad en la situación: solapes, inconsistencias, confusión.
- Rápido proceso de maduración tecnológica.
- Futuro optimista.
- Nuevo paradigma de negocio.

Sitios de consulta (i)

- W3C Architecture Domain. Extensible Markup Language (XML).
 - <http://www.w3.org/XML/>



Sitios de consulta (ii)

- XML-ES. Universidad Carlos III de Madrid.
 - <http://www.it.uc3m.es/~xml/enlaces.html#xml.db>



Francisco Ruiz. XML y derivados

53

Francisco Ruiz González

Contacto Personal:

Email: francisco.ruizg@uclm.es

Web: <http://alarcos.inf-cr.uclm.es/per/fruiz/cur/conf/xml/xml.htm>

Grupo **Alarcos**:

I+D en

- *Ingeniería del Software*
- *Bases de Datos*
- *Sistemas de Información*



alarcos.inf-cr.uclm.es

Francisco Ruiz. XML y derivados

54

Referencias (i)

- Canonical XML Version 1.0. W3C Recommendation, 15 March 2001.
 - www.w3.org/TR/xml-c14n
- Cascading Style Sheets, level 2 (CSS2 Specification). W3C Recommendation, 12-May-1998.
 - www.w3.org/TR/1998/REC-CSS2-19980512/
 - www.sidar.org/recur/desdi/traduc/es/css/cover.html
- Document Object Model (DOM).
 - www.w3.org/DOM/
- Dublin Core Metadata Element Set, Version 1.0: Reference Description. Sep-1998.
 - dublincore.org/documents/1998/09/dces/www.redirect.es/search/dces/
- eXtensible Access Control Markup Language (XACML) 1.0; OASIS XACML TC.
 - www.oasis-open.org/committees/xacml/
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation, 6 October 2000. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler.
 - www.w3.org/TR/REC-xmlwww.sidar.org/recur/desdi/traduc/es/xml/xml1/index.html
- eXtensible Rights Markup Language (XrML) 2.1; OASIS Rights Language TC.
 - www.oasis-open.org/committees/rights/

Referencias (ii)

- Extensible Stylesheet Language (XSL) Version 1.0. W3C Recommendation, 15 October 2001.
 - www.w3.org/TR/xsl/
- HTML 4.01 Specification. W3C Recommendation, 24 December 1999.
 - www.w3.org/TR/html4/www.sidar.org/recur/desdi/traduc/es/html401-es/cover.html
- Meta Object Facility (MOF) Specification; version 1.4, April-2002. Object Management Group.
 - www.omg.org/technology/documents/formal/mof.htm
- Namespaces in XML. W3C Recommendation, 14-January-1999.
 - www.w3.org/TR/REC-xml-names/
- OASIS UDDI Version 3.0. UDDI Spec Technical Committee Specification, 19 July 2002.
 - www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3
- Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 22 February 1999.
 - www.w3.org/TR/1999/REC-rdf-syntax-19990222/
 - www.sidar.org/recur/desdi/traduc/es/rdf/rdfesp.htm

Referencias (iii)

- Scalable Vector Graphics (SVG) 1.1 Spec. W3C Recommendation, 14 January 2003.
 - www.w3.org/TR/SVG11/
- Security Assertion Markup Language (SAML) 1.0; OASIS Security Services TC.
 - www.oasis-open.org/committees/security/
- Simple API for XML (SAX).
 - www.saxproject.org/
- SOAP Version 1.2 Part 0: Primer. W3C Recommendation, 24 June 2003.
 - www.w3.org/TR/2003/REC-soap12-part0-20030624/
- Standard Generalized Markup Language (SGML).
 - xml.coverpages.org/sgml.html
 - sirio.deusto.es/abaitua/konzeptu/sgml/sgml0.htm
- Standards for XML and Web Services Security. IEEE Computer, 36(4) April-2003, pp. 96-98. Martin Naedele.
- Web Services Architecture. W3C Working Draft, 8 August 2003.
 - www.w3.org/TR/2003/WD-ws-arch-20030808/
- Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language. W3C Working Draft, 11 June 2003.
 - www.w3.org/TR/wsdl12/

Referencias (iv)

- XForms 1.0. W3C Proposed Recommendation, 01 August 2003.
 - www.w3.org/TR/2003/PR-xforms-20030801/
- XHTML 1.0 The Extensible HyperText Markup Language (2nd Edition), A Reformulation of HTML 4 in XML 1.0. W3C Recommendation, 1 August 2002.
 - www.w3.org/TR/xhtml1/
 - www.sidar.org/recur/desdi/traduc/es/xhtml1/xhtml11.htm
- XML Base. W3C Recommendation, 27 June 2001.
 - www.w3.org/TR/xmlbase/
 - www.sidar.org/recur/desdi/traduc/es/xml/xmlbase20010627es/index.htm
- XML Encryption Syntax and Processing. W3C Recommendation, 10 December 2002.
 - www.w3.org/TR/xmlenc-core/
- XML Inclusions (XInclude) Version 1.0. W3C Candidate Recommendation, 17 September 2002.
 - www.w3.org/TR/xinclude/
- XML Key Management Specification (XKMS) Version 2.0. W3C Working Draft, 18 April 2003.
 - www.w3.org/TR/xkms2/

Referencias (v)

- XML Linking Language (XLink) Version 1.0. W3C Recommendation, 27 June 2001.
 - www.w3.org/TR/xlink/
- XML Metadata Interchange (XMI), ver. 2.0. Object Management Group, May-2003.
 - www.omg.org/technology/documents/formal/xmi.htm
- XML Path Language (XPath) Version 1.0. W3C Recommendation, 16 Nov-1999.
 - www.w3.org/TR/xpath
- XML Pointer Language (XPointer). W3C Working Draft, 16 August 2002.
 - www.w3.org/TR/xptr/
 - geneura.ugr.es/~victor/cursillos/xml/XPointer/
- XML Schema Part 0: Primer. W3C Recommendation, 2 May 2001.
 - www.w3.org/TR/xmlschema-0/
 - www.sidar.org/recur/desdi/traduc/es/xml/xmlschema/xmlschema-0es.html
- XML-Signature Syntax and Processing. W3C Recommendation, 12 February 2002.
 - www.w3.org/TR/xmlsig-core/
- XQuery 1.0: An XML Query Language. W3C Working Draft, 22 August 2003.
 - www.w3.org/TR/xquery/
- XSL Transformations (XSLT) Version 1.0. W3C Recommendation, 16 Nov-1999.
 - www.w3.org/TR/xslt
 - www.sidar.org/recur/desdi/traduc/es/xml/xml-stylesheet-19990629-es.html