

Estructuración, Difusión y Procesamiento de Información Usando XML y SOAP. *

M. Calle¹, J. M. García², G. Villa³, J. L. Andrade⁴

¹Ingeniero de Organización Industrial

Dpto. Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros. Universidad de Sevilla.
Camino de los Descubrimientos, s.n., 41092 - Sevilla
mcalles@esi.us.es

²Ingeniero Informático

Dpto. Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros. Universidad de Sevilla.
Camino de los Descubrimientos, s.n., 41092 - Sevilla
jmgs@esi.us.es

³Ingeniero Industrial

Dpto. Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros. Universidad de Sevilla.
Camino de los Descubrimientos, s.n., 41092 - Sevilla
gvilla@esi.us.es

⁴Ingeniero de Telecomunicación

Dpto. Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros. Universidad de Sevilla.
Camino de los Descubrimientos, s.n., 41092 - Sevilla
andrade@cartuja.us.es

RESUMEN

El intercambio de información a través de Internet ha estado limitado tradicionalmente por dos motivos: el formato de la información y los protocolos de comunicaciones. Con la aparición de XML y SOAP se permite la transferencia de información con estructuras complejas, dejando atrás los anticuados pares parámetro/valor. Estas nuevas tecnologías posibilitan el desarrollo de formatos universales que normalicen la información para la difusión a través de la comunidad científica. En este trabajo se expone una descripción de estas técnicas y las aplicaciones que se están desarrollando en campos diversos como la optimización, modelado, realidad virtual, bioinformática, etc.

1. Introducción de XML y SOAP.

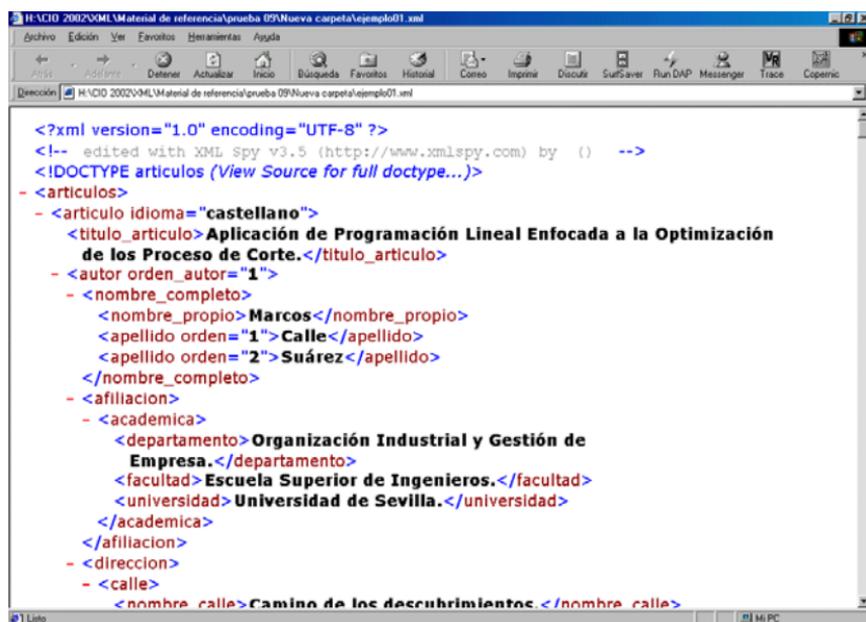
En este apartado se presenta una breve descripción de XML y SOAP.

1.1 XML.

XML corresponde a las siglas de “eXtensible Mark-up Language” o “lenguaje de marcado extensible”. Este lenguaje surge como respuesta a la necesidad de creación de documentos estructurados que posibiliten el intercambio de información entre aplicaciones distantes e independientes. El carácter de independencia entre las aplicaciones se refiere al punto de vista de la implementación, tanto del lenguaje utilizado para dichas aplicaciones (java, C++, VB, etc), como de la plataforma en la que se implanta dicha aplicación (Windows, Unix, etc). XML se acoge a la definición de un metalenguaje, debido a que se trata de un lenguaje usado para la creación de otros lenguajes.

* Este trabajo se deriva de la participación de sus autores en un proyecto de investigación financiado por MCYT con referencia DPI2000-0567, titulado “Diseño y desarrollo de un sistema integrado de producción y distribución en un entorno metropolitano”.

El organismo encargado del desarrollo de XML es el World Wide Web Consortium [1], o W3C, que define a XML como “un lenguaje sencillo y flexible basado en SGML”. SGML (Standard Generalized Markup Language) es un estándar ISO.



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by () -->
<!DOCTYPE articulos (View Source for full doctype...)>
- <articulos>
- <articulo idioma="castellano">
  <titulo_articulo>Aplicación de Programación Lineal Enfocada a la Optimización
  de los Proceso de Corte.</titulo_articulo>
- <autor orden_autor="1">
  - <nombre_completo>
    <nombre_propio>Marcos</nombre_propio>
    <apellido orden="1">Calle</apellido>
    <apellido orden="2">Suárez</apellido>
  </nombre_completo>
- <afiliacion>
  - <academica>
    <departamento>Organización Industrial y Gestión de
    Empresa.</departamento>
    <facultad>Escuela Superior de Ingenieros.</facultad>
    <universidad>Universidad de Sevilla.</universidad>
  </academica>
</afiliacion>
- <direccion>
  - <calle>
    <nombre_calle>Camino de los descubrimientos.</nombre_calle>
```

Figura 1. Ejemplo de documento XML.

1.1.1 Lenguajes asociados con XML.

XML no existe de modo independiente, sino que se apoya en otros aportes del W3C, entre los que citaremos algunos como por ejemplo:

- Esquemas XML. El objetivo de estos esquemas es establecer las estructuras de tipos de documentos XML. Estos esquemas cumplen con unos cometidos similares a los DTD de SGML. Aunque XML soporta los DTD, los esquemas son preferibles debido a que usan la misma sintaxis que XML.
- Espacios de nombres. El origen de estos espacios de nombre es dar mayor autonomía a los documentos XML, de modo que un mismo documento XML se pueda basar en varios esquemas al mismo tiempo. Los espacios de nombres también resuelven las posibles colisiones entre los nombres de los elementos de los diferentes esquemas.
- XLink y Xpointer. Estas dos metodologías guardan una estrecha relación en sus objetivos, debido a que son las encargadas de crear los hipervínculos y de referenciarlos respectivamente.
- XSL (eXtensible Stylesheet Language). Este lenguaje es el responsable de crear las hojas de estilo. Dichas hojas de estilo son las encargadas de dar un formato de presentación al documento XML. Otras de las atribuciones de XSL es la de

transformar un documento XML en otro tipo de documento, por ejemplo, HTML, PDF, etc. En la figura 2 se muestra un ejemplo de presentación de un documento XML en un navegador usando XSL.



Figura 2. Ejemplo de documento XML.

- Xpath. Establece el modo de referenciar a los elementos de un documento de XML. Es usado por XSL.
- DOM (Document Object Model). Se trata de un modelo de objeto del documento, es decir, el documento XML se comporta como un objeto, proporcionando un conjunto de propiedades y métodos que posibilitan la consulta y modificación del citado documento XML.
- SAX (Simple API for XML). Se trata de una alternativa al DOM destacando entre otras características la velocidad de proceso.

1.2 SOAP (Simple Object Access Protocol).

La traducción del nombre es Protocolo Sencillo de Acceso a Objeto. Se trata de un protocolo sencillo que posibilita el intercambio de información estructurada y que permite distintos tipos de datos (cadenas de caracteres, expresiones numéricas, matrices, etc). Otra de las características diferenciadoras es el entorno en el que se utiliza. Dicho entorno es de tipo distribuido, en el cual existen múltiples aplicaciones independientes que intercambian información con formato XML. Otra de las posibilidades de SOAP es la llamada a procedimientos remotos (RPC).

2. Metodología de intercambio de información.

A continuación se presentan las dos metodologías intercambio de información disponibles. En

primer lugar la tradicional, caracterizada por tratarse de información desestructurada. En segundo lugar, la aportada por XML y SOAP, que posibilitan una información estructurada.

2.1 Intercambio de información desestructurada.

La arquitectura de los sistemas tradicionales, con sus correspondientes protocolos de comunicación y formatos de intercambio, han permitido únicamente un flujo de información en el que se omite la estructura de sus bases de datos originarias. Debido a ello era necesario un posterior proceso de tratamiento y análisis de la información, siendo necesaria la intervención del usuario. De todo esto se desprende el nivel de complejidad del proceso a realizar por el usuario una vez recepcionada la información. Dicha complejidad depende de la estructura de las bases de datos empleadas. En la figura 3 se muestra un diagrama con el intercambio de información desestructurada.

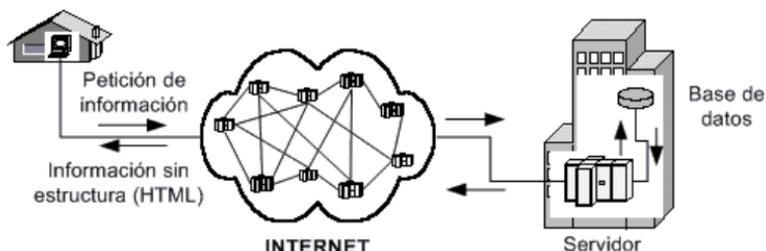


Figura 3. Intercambio de información desestructurada.

2.2 Intercambio de información estructurada.

Gracias a XML y SOAP ha aparecido una nueva arquitectura de los sistemas, en la cuál es posible transmitir información con una estructura incorporada. Esta metodología de intercambio de información estructurada se basa en el uso de XML para estructurar la información y el protocolo SOAP, sobre el que viaja la citada información. A grandes rasgos el uso de esta metodología pasa por varias etapas. En primer lugar se produce una petición de información desde el cliente al servidor. A continuación, el servidor recopilará la información necesaria y le proporciona un formato de XML. Una vez preparada la información se procede a transmitirla mediante el citado protocolo. El siguiente paso es la recepción, tratamiento y análisis de la información, todo ello realizado de modo automático por la aplicación oportuna. Gracias a todo ello, se consigue una mayor eficiencia en el intercambio de información, y una mayor simplicidad de las tareas del usuario.

Otro punto importante a destacar es el aumento de la interoperabilidad de los distintos sistemas. Esta característica consiste en que los sistemas de información que se comunican mediante la metodología expuesta podrán estar implementados mediante el lenguaje de programación deseado, independientemente del uso de XML y SOAP. De este modo podemos implantar un sistema con Java, el cual se comunicará con otros implementados respectivamente en C++, Visual Basic, Perl, etc. En la figura 4 se muestra un sistema en el que se produce un intercambio de información estructurada.

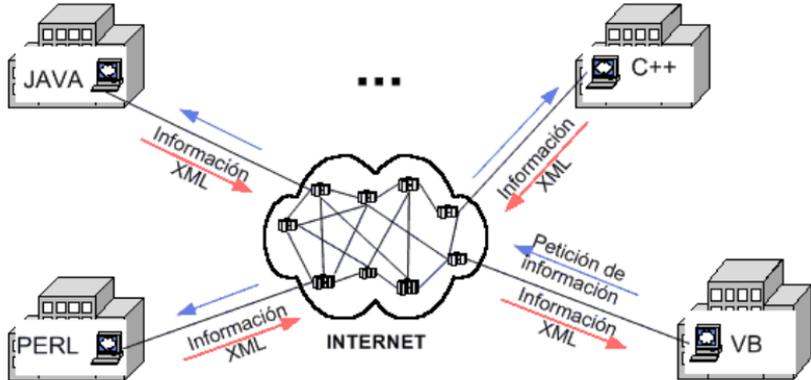


Figura 4. Intercambio de información estructurada.

3. Desarrollos actuales basados en XML y SOAP.

A continuación se presentan algunos ejemplos de aplicaciones basadas en XML y SOAP que se encuentran en fase de desarrollo, además de otras aplicaciones existentes en Internet.

3.1 Aplicación en el campo de la optimización

Entre los objetivos destacables de esta aplicación se encuentran los siguientes:

- Creación de un lenguaje matemático común para el modelado de problemas de optimización como por ejemplo programación lineal, etc.
- Conversión del formato común del modelo de problemas específicos a otros formatos de herramientas comerciales.

Un objetivo más ambicioso a largo plazo es la creación de una comunidad entre distintos organismos investigadores, como por ejemplo, universidades. El objetivo de dicha comunidad sería la prestación recíproca de servicios de computación. Es decir, debido a que cada organismo posee herramientas propias de resolución de problemas, esta comunidad posibilitaría el intercambio de modelos matemáticos particulares desarrollados por unos organismos, para la petición de resolución de los citados modelos mediante herramientas de resolución propietarias de otros organismos. Estas peticiones de resolución no producirían una sobrecarga de dichos programas debido a la capacidad de los ordenadores actuales y a los tiempos ociosos de dichos equipos.

A continuación se muestra un ejemplo de uso de los modelos anteriores. En la figura 5 se muestra un ejemplo de petición de resolución de un problema. La totalidad del mensaje va incluida dentro del elemento “ENVELOPE”. Dicho elemento “ENVELOPE” está compuesto a su vez de los elementos “HEADER” y “BODY”. El elemento “HEADER” contiene la información de la aplicación destinataria del mensaje y otra información necesaria para el procesamiento de dicha información por parte del destinatario. Por otro lado el elemento “BODY” es el que contiene realmente la información que conforma el mensaje. En nuestro ejemplo se ha enviado un sencillo modelo matemático de programación lineal. El modelo contiene una serie de elementos que aportan una determinada información. Los elementos y su información se muestran a continuación:

- “nombreModelo”. Contiene el nombre del citado modelo, así como la fecha y versión del modelo.
- “funcionObjetivo”. Expresa que se maximizará “ $100x_1 + 100x_2$ ”.
- “restricciones”. Contiene una única restricción que es “ $2x_1 + 3x_2 \leq 40$ ”.
- “variables”. Ambas variables son enteras, estando la variable “ x_1 ” acotada entre $[0, \text{infinito})$ y la variable “ x_2 ” esta acotada entre $[7, \text{infinito})$.

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <httpEN:peticionSOAP
      xmlns:httpEN="http://ebiz.us.es/ftpNS"
      SOAP-ENV:mustUnderstand="1">
      <destinoRespuesta>
        < sitioFtp>ftp://ebiz.us.es/</ sitioFtp>
        < directorioFtp>respuestaSOAP/optimizacion</ directorioFtp>
        < nombreArchivoRespuestaFtp>modeloresuelto_soap</ nombreArchivoRespuestaFtp>
        < nombreUsuario>comunidad</ nombreUsuario>
        < contraseña>resultadocalculo</ contraseña>
      </ destinoRespuesta>
    </ httpEN:peticionSOAP>
  </ SOAP-ENV:Header>
  < SOAP-ENV:Body>
    < esquemaModelo.modelo
      xmlns:esquemaModelo="http://ebiz.us.es/esquemas/lineal_programming.xsd">
      < nombreModelo version="2002-04-05/00000001">ejemplo 1</ nombreModelo>
      < funcionObjetivo tipo="max">
        < sumando variable="x1">100</ sumando>
        < sumando variable="x2">100</ sumando>
      </ funcionObjetivo>
      < restricciones>
        < restricción id="rección01">
          < sumandos>
            < sumando variable="x1">2</ sumando>
            < sumando variable="x2">3</ sumando>
          </ sumandos>
          < signo>le</ signo>
          < terminoIndependiente>40</ terminoIndependiente>
        </ restricción>
      </ restricciones>
      < variables>
        < variable>
          < cotaInferior>0</ cotaInferior>
          < cotaSuperior>ilimitada</ cotaSuperior>
          < nombre>x1</ nombre>
          < tipo>entera</ tipo>
        </ variable>
        < variable>
          < cotaInferior>7</ cotaInferior>
          < cotaSuperior>ilimitada</ cotaSuperior>
          < nombre>x2</ nombre>
          < tipo>entera</ tipo>
        </ variable>
      </ variables>
    </ esquemaModelo.modelo>
  </ SOAP-ENV:Body>
</ SOAP-ENV:Envelope>

```

Figura 5. Ejemplo de petición de resolución de modelo de programación lineal.

En la figura 6 se muestra un ejemplo de respuesta de solución del modelo anterior. En este caso solo existe el elemento “BODY”. La respuesta contiene la siguiente información:

- “nombreModelo”, contiene el nombre del citado modelo, así como la fecha y versión del modelo.
- “funcionObjetivo”. Expresa el valor alcanzado en la función objetivo igual a 1600.
- “variables”. Valores de las variables, tanto “x₁” como “x₂” igual a 8.

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <esquemaSolucionModelo:modelo
      xmlns:esquemaSolucionModelo="http://ebiz.us.es/esquemas/solucion.xsd">
      <nombreModelo version="2002-04-05/00000001">ejemplo 1</nombreModelo>
      <funcionObjetivo>1600</funcionObjetivo>
      <variables>
        <variable nombre="x1">8</variable>
        <variable nombre="x2">8</variable>
      </variables>
    </esquemaSolucionModelo:modelo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Figura 6. Ejemplo de respuesta de solución de modelo.

3.2 Integración de sistemas de gestión.

Esta aplicación trata de utilizar XML como soporte para la gestión de una organización en la cual existen varias delegaciones geográficas, encargadas de realizar las instalaciones, y un único centro de fabricación encargado de realizar la actividad productiva. Cada centro lleva a cabo su gestión de modo independiente, tanto la fábrica como las distintas delegaciones. Mediante XML y SOAP se mantienen intercomunicadas los distintos núcleos que conforman la organización. Cuando se realiza una petición de información se comunican todos los núcleos de modo que cada uno de ellos recupera la información estructurándola de modo independiente, y procediendo a enviarla posteriormente al solicitante. Por último, el solicitante recopila toda la información procedente de cada uno de los núcleos y la unifica. Esta es una característica fundamental que posibilita un resultado análogo al obtenido con una base de datos centralizada.

Entre las ventajas aportadas destacamos las siguientes:

- Eliminación del sistema de gestión central.
- Máximo aprovechamiento de las capacidades de los sistemas de gestión de cada núcleo.
- Independencia en el mantenimiento y ampliaciones de los sistemas de gestión de los distintos núcleos.

Entre las futuras ampliaciones se encuentra la utilización de agentes autónomos. Dichos agentes supervisan las actuaciones sobre los datos almacenados de forma local. A modo de ejemplo se expone a continuación una función aportada por dichos agentes. Una tarea cotidiana es la petición de un nuevo pedido en la cual se cumplimenta, vía Internet, una orden de pedido en el sistema de gestión del proveedor. A continuación se registra en nuestro sistema de gestión la anterior orden de pedido, para tener constancia de dicha orden. Gracias a XML, SOAP y los agentes autónomos se puede registrar en nuestro sistema de gestión dicha

orden de pedido, siendo el agente autónomo el encargado de registrar de modo automático dicha orden en el sistema de gestión del proveedor.

3.3 Publicaciones electrónicas.

En el campo de la difusión electrónica de la información, como por ejemplo artículos de revistas, XML proporciona una serie de ventajas derivadas de la estructuración de los datos. Además de todas las ventajas que aportan las bases de datos, el uso de XML proporciona una ventaja revolucionaria. Dicha ventaja consiste en la búsqueda de información basada en el significado contextual de los datos almacenados. Dicha búsqueda consiste en distinguir el significado de cualquier dato dependiendo del elemento en el cuál aparezca. De este modo una misma palabra, por ejemplo “Sevilla”, podrá tomar los significados de lugar de suceso, lugar de nacimiento, lugar de defunción, etc.

Además es posible dotar a todos los documentos XML de un formato estándar que se almacena de modo independiente de los datos que conforman la información de interés. En la figura 1 se presenta un ejemplo de un artículo con formato XML. El formato final de presentación de dicho artículo en un navegador se presenta en la figura 2.

4. Referencias.

- [1] World Wide Web Consortium [en línea], [Consulta: 10 marzo 2002]. <<http://www.w3.org>>
- [2] Adler, S., et al., (2001) “Extensible Stylesheet Language (XSL) Version 1.0” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/xsl/>>
- [3] Apparao, V., et al., (1998) “Document Object Model (DOM) Level 1” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/REC-DOM-Level-1>>
- [4] Birbeck, M., et al., (2001) “Professional XML”, 2nd Edition . Wrox Press Ltd.
- [5] Box, D., et al., (2000) “SOAP Version 1.2 Part 0: Primer” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/soap12-part0/>>
- [6] Bray, T., et al., (2000) “Extensible Markup Language (XML) 1.0 (Second Edition)” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/REC-xml>>
- [7] Bray, T., et al., (1999) “Namespaces in XML” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/REC-xml-names>>
- [8] Fallside, D.C., et al., (2001) “XML Schema Part 0: Primer” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/xmlschema-0/>>
- [9] Goldbarb, C.F., (2000) “The XML Handbook”. Prentice Hall PTR.
- [10] Lie, H.W., et al., (1999) “Cascading Style Sheets (CSS1) Level 1 Specification” [en línea], World Wide Web Consortium. [Consulta: 10 marzo 2002]. <<http://www.w3.org/TR/REC-CSS1>>
- [11] Seely, S., (2002) “SOAP: Cross Platform Web Service Development Using XML”. Prentice Hall PTR.