

# Semantic Annotation for WSDL and XML SAWSDL

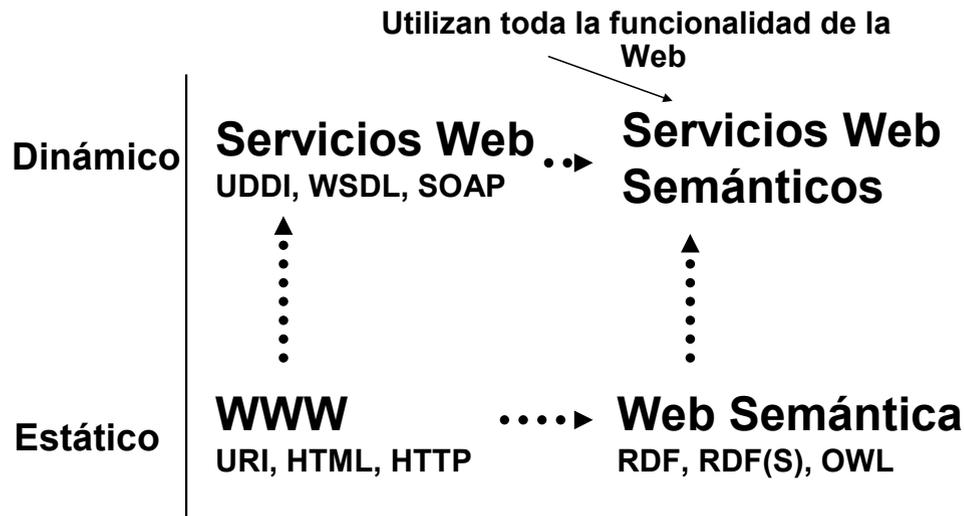
Presentación:  
Luis Miguel Serrano Cámara  
Recuperación de la Información

- 1.- Introducción**
- 2.- SAWSDL en WSDL 2.0**
- 3.- SAWSDL en WSDL 1.1**
- 4.- Ejemplos**
- 5.- Páginas de Interés**
- 6.- Conclusiones**

# Servicios Web Semánticos

**Servicios Web Semánticos = Servicios Web + Web Semántica**

**Web Semántica: Web extendida, con información bien definida, permitiendo a computadores y personas interoperar entre sí**



## Concepto núcleo de la Web Semántica

**Las ontologías representan de manera formal y consensuada especificaciones de conceptos, que proveen un conocimiento compartido y común del dominio como información semántica procesable por las máquinas e interoperable a través de agentes (organizaciones, individuos y software).**

**Lenguajes de modelado de ontologías: OWL y WSML**

# SAWSDL (Semantic Annotations for WSDL)

**Desarrollado por un grupo de trabajo del W3C**

**Añade extensiones al lenguaje de descripción de servicios Web (WSDL)**

**Se aplica a:**

**WSDL 1.1 (Estándar actual)**

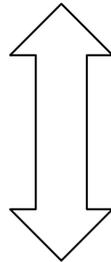
**WSDL 2.0 (A Propose recommendation, paso oficial a su “oficialidad”)**

**Similar a WSDL-S. Se diferencia en que permite especificar el comportamiento de los servicios -> Permite Coreografía de servicios**

# WSDL 2.0 Introducción

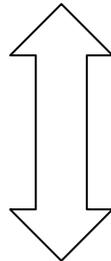
Conceptualmente, WSDL 2.0 tiene los siguientes componentes para la representación descriptiva de servicios:

**Type Definition,  
Interface,  
Interface Operation,  
Interface Fault,**



**Definición abstracta de servicios**

**Binding,  
Service  
Endpoint.**



**Implementación de servicios**

La especificación de SAWSDL se centra en la anotación semántica de la definición abstracta de un servicio

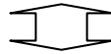
para permitir el descubrimiento, la composición y la invocación dinámicos de servicios.

A modo de resumen, los atributos extendidos que SAWSDL proporciona son los siguientes:

- **modelReference**, especifica la asociación entre componente de un esquema de **WSDL** o de **XML** y un concepto en un cierto modelo semántico.
- Dos atributos, llamados **liftingSchemaMapping** y **loweringSchemaMapping**, que son añadidos a declaraciones de elementos de **XML**, definiciones de tipos complejos, simples ..para mappings entre datos y XML.

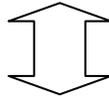
## La anotación **modelReference**:

wSDL:interface



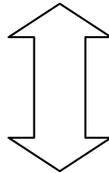
proporciona una clasificación u otras descripciones semánticas del interfaz

wSDL:operation  
wSDL:fault



proporciona información semántica acerca de la operación

xs:element  
xs:complexType  
xs:simpleType  
xs:attribute



definen la semántica de entrada o salida de datos en operaciones WSDL.

Anotación **modelReference** en `wsdl:interface` :

```
...  
<wsdl:interface name="Order"  
sawSDL:modelReference="http://example.org/categorization/products/electr  
onics">  
...  
</wsdl:interface>  
...
```

## Anotación **modelReference** en wsdl:operation :

```
...
<wsdl:operation name="order" pattern="http://www.w3.org/2006/01/wsdl/in-out"
  sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#RequestPurchaseOrder">
  <wsdl:input element="OrderRequest"/>
  <wsdl:output element="OrderResponse"/>
</wsdl:operation>...
```

<http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder>

```
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
-<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#"
xml:base="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#">
  <owl:Ontology />
  ....
  ....
  <owl:ObjectProperty rdf:ID="hasIdentifier">
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
    <rdfs:range rdf:resource="#Identifier" />
  </owl:ObjectProperty>
  <owl:Class rdf:ID="RequestPurchaseOrder" />
</rdf:RDF>
```

Anotación **modelReference** en `wSDL:faults` :

```
...  
<wSDL:interface name="Order">  
  <wSDL:fault name="ItemUnavailable" element="AvailabilityInformation"  
    sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/p  
urchaseorder#ItemUnavailable"/>  
...  
</wSDL:interface
```

Anotación **modelReference** en tipos simples `xs:simpleType` :

```
...  
<xs:simpleType name="Confirmation"  
  sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purch  
  aseorder#OrderConfirmation">  
...  
</xs:simpleType>  
...
```

En este ejemplo, cualquier elemento o atributo que sea tipo `Confirmation` es descrito por el concepto `OrderConfirmation` en el modelo semántico referenciado, en esta caso por la ontología orden de compra (`purchaseorder`).

Anotación **modelReference** en **xs:complexType** (en los elementos internos):

```
...
<xs:complexType>
  <xs:sequence minOccurs="1" maxOccurs="unbounded">
    <xs:element name="quantity" type="xs:integer"
      sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#Quantity"/>
    <xs:element name="UPC" type="xs:string"
      sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#Product
      Code"/>
  </xs:sequence>
</xs:complexType>
...
```

En este extracto del código, cada elemento interno tiene una anotación a conceptos en el modelo semántica, como "Quantity" y "ProductCode" que describen cada uno de los componentes del tipo complejo.

```
.....
<owl:FunctionalProperty rdf:ID="hasLexicalRepresentation">
  <rdfs:domain rdf:resource="#Identifier" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
</owl:FunctionalProperty>
<owl:Class rdf:ID="Quantity" />
= <owl:DatatypeProperty rdf:ID="hasAmount">
  <rdfs:domain rdf:resource="#Quantity" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
</owl:DatatypeProperty>
```

Anotación `modelReference` en `xs:complexType` (directamente en el tipo compuesto):

```
...
<xs:complexType
  sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchase
  order#OrderRequest">
  <xs:sequence minOccurs="1" maxOccurs="unbounded">
    <xs:element name="quantity" type="xs:integer"
    <xs:element name="UPC" type="xs:string"
  </xs:sequence>
</xs:complexType>...
```

Aquí, el tipo complejo en su totalidad es anotado con referencia al concepto “OrderRequest”, el cual describe los elementos “Quantity” y “ProductCode” que conforman el tipo complejo.

**Si conceptos múltiples describen el tipo complejo, todos sus URIs se puede incluir en el valor de la propiedad `modelReference`.**

Un tipo compuesto puede ser anotado en su totalidad y elemento a elemento de forma simultánea, siendo estas anotaciones independientes la una de las otras.

## Anotación `modelReference` en `xs:element`

```
...  
<xs:element name="OrderRequest"  
  sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#OrderRequest">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="customerNo" type="xs:integer" />  
      <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded" />  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
...
```

En este ejemplo, la anotación indica que el elemento "OrderRequest" es descrito por el concepto "OrderRequest" en el modelo semántico referido.

Este ejemplo es muy similar a una anotación a nivel superior del `complexType` en que el elemento que es anotado está definido en términos de un `complexType` y la anotación describe el "OrderRequest" en su totalidad.

### Anotación `modelReference` en `xs:attribute`

Si el elemento “quantity” del ejemplo anterior, hubiese sido definido como un atributo, la anotación `modelReference` en este caso sería como la mostrada a continuación:

```
...  
<xs:attribute name="quantity" type="xs:integer"  
    sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#Quantity"/>  
...
```

### Las anotaciones **liftingSchemaMapping** y **loweringSchemaMapping**

El valor de los atributos **liftingSchemaMapping** y **loweringSchemaMapping** son un conjunto de cero o más URIs que referencia a definiciones de mapping.

**liftingSchemaMapping**

Define cómo un documento XML se conforma con un cierto modelo semántico, es decir la salida del proceso de la transformación será datos semánticos. La entrada a la transformación es el elemento de XML en donde se encuentra hecha la referencia a los mapping .

**loweringSchemaMapping**

Define cómo los datos en un modelo semántico se transforman a los datos XML.  
La entrada serán unos ciertos datos semánticos. La salida del proceso será el elemento de XML en que declaración del mapeo está situado; o un elemento válido acorde con el tipo del elemento donde el mapeo esta localizado

### Las anotaciones **liftingSchemaMapping** y **loweringSchemaMapping**

El mapeado entre esquemas soluciona la tarea de post-descubrimiento cuando se utiliza un Web service, solucionando los encajes de información entre inputs y outputs.

Es posible especificar la reducción o aumento (lowering o lifting) de información, así como ambas simultáneamente en un elemento.

**Los múltiples valores tanto en lifting como en lowering se interpretan como alternativas.**

El procesador del cliente debe elegir uno de ellos a aplicar, y esta elección esta a discreción total del cliente.

Por ejemplo, la elección del mapeo puede ser realizada en función del idioma en que estos estén realizados (ya que existe la posibilidad de diferentes idiomas), o por la eficiencia del mapeo o por otras referencias.

Se pueden utilizar como lenguajes de mapeado **xlst** u otros lenguajes de mapeado tales como **SPARQL** o **XQuery**.

## SAWSDL (Semantic Annotations for WSDL 2.0)

El siguiente extracto de código, muestra como XSLT puede ser usado mapping en un elemento XSD a un concepto del modelo semántico.

```
...  
<xs:element name="OrderRequest"  
    sawsdl:modelReference=http://www.w3.org/2002/ws/sawsdl/spec/  
    ontology/purchaseorder#OrderRequest  
    sawsdl:liftingSchemaMapping="http://www.w3.org/2002/ws/sawsdl/  
    spec/mapping/Response2Ont.xslt">  
    <xs:complexType>  
        <xs:sequence>  
            <xs:element name="customerNo" type="xs:integer" />  
            <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded"  
        />  
    </xs:sequence>  
    </xs:complexType>  
</xs:element>...
```

### Anidamiento de liftingSchemaMapping o loweringSchemaMapping

```
...  
<xs:element name="orderItem" type="itemType"  
  sawsdl:liftingSchemaMapping="http://example.org/mapping/OrderItem2Ont.xslt"/>  
  <xs:complexType name="itemType"  
    sawsdl:liftingSchemaMapping="http://example.org/mapping/OrderType2Ont.xslt">  
    <xs:sequence>  
      <xs:element ref="partDesc" />  
    </xs:sequence>  
    <xs:attribute name="ItemID" type="xs:string"/>  
  </xs:complexType>  
...
```

El tipo complejo orderItem se define más adelante en el esquema y el tipo complejo en sí mismo tiene un mapping.

En tal caso, el liftingSchemaMapping especificado en el elemento sobrescribe el especificado en el tipo complejo.

La razón de especificar esa regla de sobre escritura es permitir que a un elemento que no se le palique el mapping.

## ¿WSDL 1.1?

El mecanismo para la anotación semántica descrita en esta especificación se puede aplicar también a las descripciones de servicios Web en WSDL 1.1.

Todos los atributos XML definidos en esta especificación se aplican sin la modificación a las descripciones de WSDL 1.1.

Sin embargo, en algunos casos se aplican a elementos diferentes en la estructura del documento de WSDL y un nuevo elemento se introduce para facilitar anotaciones de la operación.

### Elemento attrExtensions

WSDL 1.1 no permite extensión de atributos del elemento “**operación**”, por lo que es introducido un nuevo elemento extensión llamado attrExtension

## Elemento attrExtensions

```
...  
<wsdl11:operation name="order">  
  <wsdl11:input message="OrderRequestMessage"/>  
  <wsdl11:output message="OrderResponseMessage"/>  
  <awsdl:attrExtensions  
  
    awsdl:modelReference="http://www.w3.org/2002/ws/awsdl/spec/ontology/purchaseorder#RequestPurchaseOrder">  
</wsdl11:operation>  
...
```

Un ejemplo completo con notación SAWSDL sobre un WSDL 1.1 se puede ver en la siguiente dirección:

<http://www.w3.org/2002/ws/awsdl/spec/wsdl/order11>

## SAWSDL (Semantic Annotations for WSDL 1.1)

Cambios en la estructura de WSDL 1.1 con respecto a WSDL 2.0

WSDL 2.0	WSDL 1.1
interfaces	portType
Endpoints	Ports
--	“part” (situado bajo el elemento message)
fault	definido de modo idéntico a una input o output

**Mas ejemplos de W3C sobre WSDL 2.0:**

**Semantic Annotations for WSDL and XML Schema — Usage Guide:**

**<http://www.w3.org/2002/ws/sawSDL/spec/examples/>**

# SAWSDL (Semantic Annotations for WSDL 1.1)

## Mas ejemplos:

Se pueden encontrar hasta 10 ejemplos de WSDL 1.1 con y sin SAWSDL , para fijarse en la notación de SAWSDL



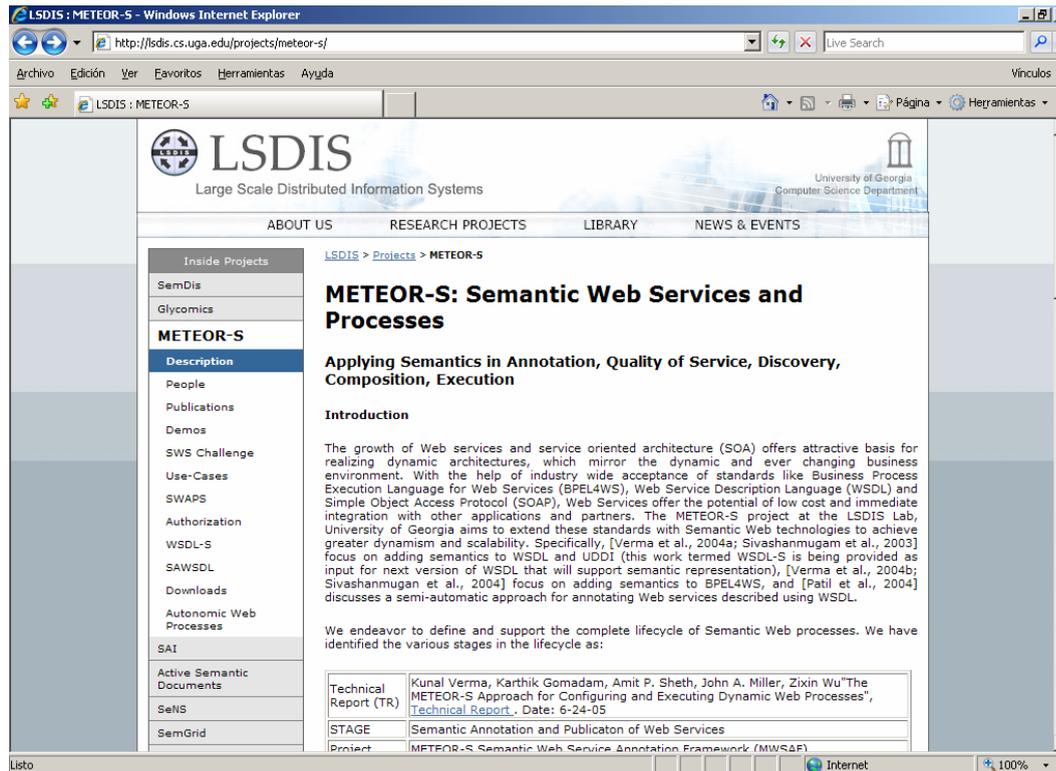
The screenshot shows the Stargate Glycomics Web Portal in a Windows Internet Explorer browser. The page features a navigation menu with links for HOME, SemBOWSER, Glyde, SAWSDL Use Case, Downloads, Mailing List, and FAQ. Below the menu, there is a section titled "Semantic Web process for Glycoproteomics data analysis" which describes the workflow for analyzing mass spectrometry data. A sub-section titled "Semantic Web Services: SAWSDL Files" provides a table comparing WSDL and SAWSDL files.

	WSDL	SAWSDL
1	<a href="#">100_CharFTClient.wsdl</a>	<a href="#">sa_100_CharFTClient.wsdl</a>
2	<a href="#">100_Sys.JavaRawmzXML.wsdl</a>	<a href="#">sa_100_Sys.JavaRawmzXML.wsdl</a>

# SAWSDL (Semantic Annotations for WSDL 1.1)

## Proyecto METEOR-S

Coordinador del proyecto: ([kunal.verma1@gmail.com](mailto:kunal.verma1@gmail.com)).



The screenshot shows a Windows Internet Explorer browser window displaying the LSDIS website. The URL is <http://lsdis.cs.uga.edu/projects/meteor-s/>. The page features the LSDIS logo and navigation tabs for 'ABOUT US', 'RESEARCH PROJECTS', 'LIBRARY', and 'NEWS & EVENTS'. A left sidebar lists various project categories, with 'METEOR-S' selected. The main content area is titled 'METEOR-S: Semantic Web Services and Processes' and includes a description, an introduction, and a table of technical reports.

**LSDIS**  
Large Scale Distributed Information Systems  
University of Georgia  
Computer Science Department

ABOUT US   RESEARCH PROJECTS   LIBRARY   NEWS & EVENTS

Inside Projects

- SemDis
- Glycomics
- METEOR-S**
- Description
- People
- Publications
- Demos
- SWS Challenge
- Use-Cases
- SWAPS
- Authorization
- WSDL-S
- SAWSDL
- Downloads
- Autonomic Web Processes
- SAI
- Active Semantic Documents
- SeNS
- SemGrid

LSDIS > Projects > METEOR-S

### METEOR-S: Semantic Web Services and Processes

#### Applying Semantics in Annotation, Quality of Service, Discovery, Composition, Execution

#### Introduction

The growth of Web services and service oriented architecture (SOA) offers attractive basis for realizing dynamic architectures, which mirror the dynamic and ever changing business environment. With the help of industry wide acceptance of standards like Business Process Execution Language for Web Services (BPEL4WS), Web Service Description Language (WSDL) and Simple Object Access Protocol (SOAP), Web Services offer the potential of low cost and immediate integration with other applications and partners. The METEOR-S project at the LSDIS Lab, University of Georgia aims to extend these standards with Semantic Web technologies to achieve greater dynamism and scalability. Specifically, [Verma et al., 2004a; Sivashanmugam et al., 2003] focus on adding semantics to WSDL and UDDI (this work termed WSDL-S is being provided as input for next version of WSDL that will support semantic representation), [Verma et al., 2004b; Sivashanmugam et al., 2004] focus on adding semantics to BPEL4WS, and [Patil et al., 2004] discusses a semi-automatic approach for annotating Web services described using WSDL.

We endeavor to define and support the complete lifecycle of Semantic Web processes. We have identified the various stages in the lifecycle as:

Technical Report (TR)	Kunal Verma, Karthik Gomadam, Amit P. Sheth, John A. Miller, Zixin Wu "The METEOR-S Approach for Configuring and Executing Dynamic Web Processes", <a href="#">Technical Report</a> . Date: 6-24-05
STAGE	Semantic Annotation and Publication of Web Services
Project	METEOR-S Semantic Web Service Annotation Framework (MWS&F)

Están implementando herramientas para SA-WSDL, con plugins de Eclipse.

# Conclusiones

---

**Estamos “comenzando” la investigación en este campo**

**Necesidad de una plataforma de ejecución. Hay muy poco desarrollado**

**Imprescindible el uso de ONTOLOGÍAS comunes.**