

Un servicio Web para investigación operativa

Marcos Calle Suárez, Gabriel Villa Caro, José Manuel García Sánchez, Jesús Racero Moreno

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

Resumen

Este trabajo presenta un servicio Web para investigación operativa. El objetivo de este servicio Web consiste en apoyar la propuesta de un lenguaje de modelado basado en XML, enfocado a problemas de investigación operativa. Con el citado lenguaje denominado ORML (Operations Research Markup Language) se pretende estandarizar el modelado matemático. Este trabajo se justifica debido a la existencia de múltiples lenguajes de modelado algebraico y paquetes de resolución, siendo cada uno de estos lenguajes compatible únicamente con algunos paquetes de resolución. Los servicios ofertados actualmente son los siguientes: en primer lugar, conversión de problemas codificados en ORML hacia otros lenguajes de modelado (GAMS, AMPL MPL, LPL y AIMMS), y el segundo lugar, resolución de problemas de flujo en redes a coste mínimo, mediante el algoritmo de resolución RelaxIV. Entre los servicios en desarrollo se encuentran los siguientes: en primer lugar, la conversión de problemas codificados en alguno de los lenguajes de modelado existente (GAMS, AMPL MPL, LPL y AIMMS) hacia ORML, y en segundo lugar, la resolución de otros tipos de problemas. La sección 2 del presente trabajo presenta brevemente la estructura de este servicio Web, así como una descripción detallada de los servicios ofrecidos.

Palabras clave: Servicio Web, investigación operativa, modelado

1. Introducción

En la actualidad existen múltiples lenguajes de modelado algebraico y paquetes de resolución. Un paquete de resolución consiste en un software dedicado a la obtención de soluciones óptimas de problemas de programación matemática. Usualmente el tamaño de los problemas a resolver por el software de resolución suele ser extenso. Debido a ello, se propusieron múltiples lenguajes compactos de modelado algebraico, facilitando de este modo la creación de los modelos de los problemas. Existen a su vez ciertas herramientas de conversión de modelos compactos hacia los formatos de entrada usados por los paquetes de resolución. Cada lenguaje de modelado previamente citado es compatible únicamente con algunos de los paquetes de resolución existentes.

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

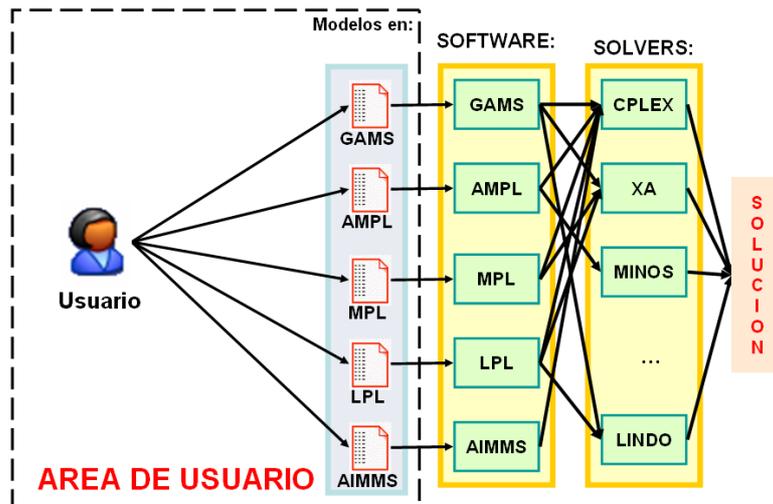


Figura 1. Ámbito de los lenguajes compactos de modelado algebraico

ORML (Operations Research Markup Language), Calle *et al.*, (2003), intenta estandarizar el modelado matemático, ofreciendo varias interfaces que permiten la conversión de los modelos codificados en éste lenguaje hacia los lenguajes compactos de modelado algebraico existentes (GAMS, Brooke *et al.*, (1988), AMPL, Fourer *et al.*, (1993), MPL, Maximal Software, (2003), LPL, Hürlimann, (2002) y AIMMS, Bisschop y Roelofs, (2001)). Gracias a ello, ORML permite a sus usuarios suplir la necesidad de aprender varios de estos lenguajes de modelado, al obtener automáticamente el modelo transformado a otros lenguajes, posibilitando de este modo la resolución de los modelos mediante cualquier paquete de resolución existente. Adicionalmente, se están desarrollando varias interfaces que permitirán a los usuarios importar modelos codificados en cualquiera de los lenguajes compactos de modelado algebraico existentes (GAMS, AMPL, etc.) y transformarlos a ORML.

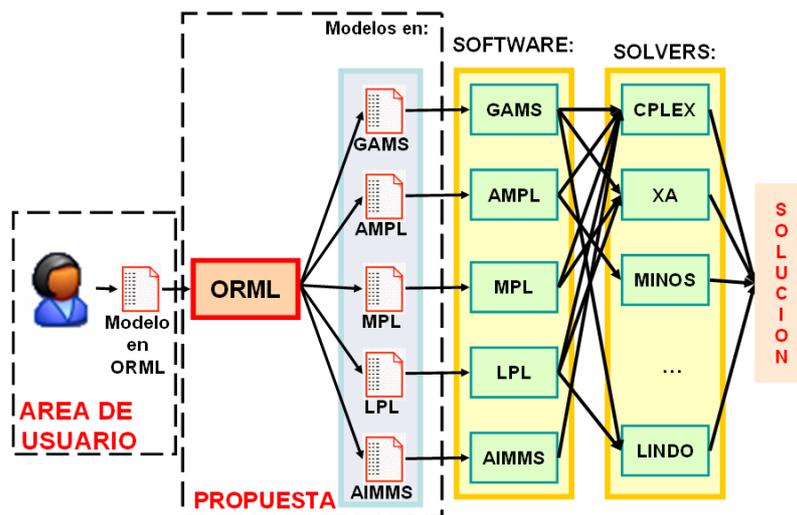


Figura 2. Ámbito de ORML

ORML consiste en un lenguaje basado en XML, dedicado al modelado de problemas de investigación operativa. Un problema codificado en ORML consiste en dos archivos (denominados respectivamente modelo paramétrico y datos del problema), permitiendo de este modo la reutilización de los modelos mediante la modificación de los datos del problema. Los componentes de ORML son: conjuntos, parámetros, variables, macros, restricciones y

funciones objetivo. Las definiciones de la versión 1.0 del esquema de XML de ORML están disponibles en la siguiente dirección:

"http://mcs.us.es/ORML/Aportaciones/ns/orml/1.0/ORML_XML_SCHEMA_1_0.zip".

Además de ORML, se ha desarrollado un lenguaje auxiliar denominado ORRML (Operations Research Results Markup Language), el cual intenta estandarizar la representación de los resultados de los problemas de programación matemática. Debido a que cada paquete de resolución utiliza su propio formato de representación de resultados, se hace necesaria la estandarización de un lenguaje para la representación de los resultados de los diferentes paquetes de resolución. ORML y ORRML facilitan la compatibilidad con los paquetes de resolución existentes. ORRML tiene el mismo objetivo que Optimization Reporting Markup Language, Ezechukwu *et al.*, (2003).

2. Servicio Web ORML

Un servicio Web consiste en una aplicación dedicada a ofrecer un conjunto de servicios a otras aplicaciones a través de Internet. El presente servicio Web ORML ha sido desarrollado con independencia de la plataforma (Windows, UNIX, etc.), utilizando para ello Java. Los paquetes empleados son enumerados a continuación:

- Java 2
- Apache Tomcat
- MySQL
- Apache SOAP
- Apache Xalan
- Apache Xerces
- Apache James

Las comunicaciones empleadas en este servicio Web se basan en los protocolos de comunicaciones HTTP y SOAP, Box *et al.*, (2000), utilizadas entre el navegador Web de los usuarios y el maestro del servicio Web, así como entre los clientes, esclavos y maestro del presente servicio Web. Las comunicaciones basadas en SOAP permiten desarrollar clientes en cualquiera de los lenguajes de programación existentes: Java, C++, Perl, etc. Las comunicaciones contenidas en el presente servicio Web son descritas mediante documentos WSDL.

2.1. SOAP

SOAP (Simple Object Access Protocol) consiste en un protocolo de comunicación mediante el cual las aplicaciones distribuidas intercambian información a través de Internet. SOAP puede ser usado sobre múltiples protocolos, como por ejemplo, HTTP y SMTP. La información intercambiada es estructurada y tipificada debido a que dicha información consiste en mensajes basados en XML. Una de las características más importante consiste en la capacidad de intercambiar información entre aplicaciones desarrolladas para distintas plataformas en distintos lenguajes de programación. Los mensajes SOAP se dividen en dos partes: cabecera y cuerpo. La cabecera es opcional, consistiendo en información complementaria, como por ejemplo, autenticación. El cuerpo es obligatorio, consistiendo en la información principal a intercambiar, pudiendo ser dicha información una petición o respuesta.

2.2. WSDL

WSDL (Web Service Description Language), Christensen *et al.*, (2001), es una descripción basada en XML del conjunto de servicios ofrecidos a través de Internet. Esta descripción detalla el formato de los mensajes que componen los procedimientos seguidos por los programas clientes para comunicarse con este servicio Web. Estos mensajes pueden ser enviados a través del SOAP, HTTP (GET/POST) o MIME.

2.3. Arquitectura del servicio Web ORML

La siguiente figura muestra las comunicaciones existentes en este servicio Web.

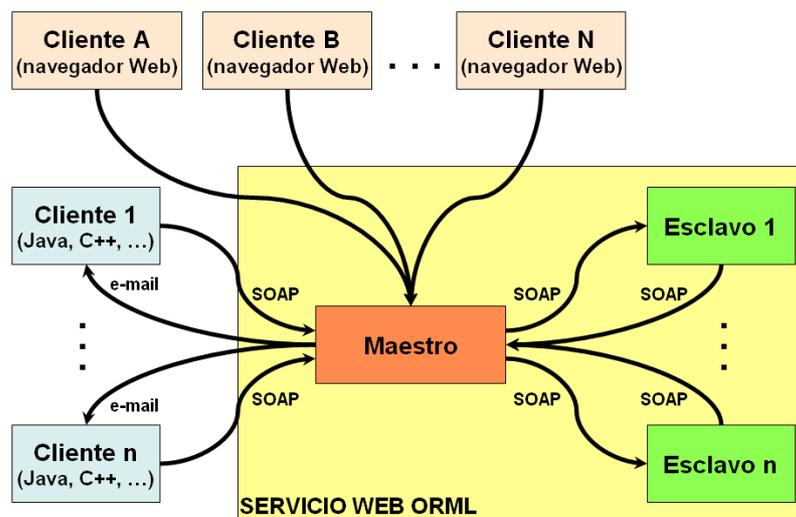


Figura 3. Arquitectura del servicio Web ORML

El servicio Web ORML se componen en los siguientes cuatro tipos de agentes:

- Maestro.
- Esclavos.
- Clientes (aplicaciones independientes).
- Clientes (navegador Web).

El maestro del servicio Web ORML se encargan de las siguientes tareas:

- Recibir las peticiones de resolución y conversión de problemas enviados por los clientes (navegador Web y aplicaciones independientes).
- Recibir las peticiones de registro de nuevos esclavos.
- Administrar las peticiones enviadas por los clientes.
- Enviar trabajos a los clientes.
- Recibir las soluciones de los trabajos enviados a los clientes.
- Enviar soluciones de los trabajos a los usuarios.

Los esclavos son los encargados de resolver las peticiones enviadas por los usuarios. En este servicio Web se permite la adición en tiempo de ejecución de nuevos esclavos, incrementando

de este modo la capacidad de servicio. Esta característica intenta usar las capacidades de computación ociosas de los equipos, independientemente de la plataforma.

Los usuarios pueden crear nuevos clientes, implementados en cualquier lenguaje de programación (Java, C++, Perl, etc.), siendo el único requisito el establecimiento de las comunicaciones SOAP con el maestro de acuerdo con una estructura preestablecida. Dicha estructura de los mensajes SOAP son descritas mediante WSDL. Debido a todo ello, los usuarios pueden incluir los servicios ofrecidos en otras aplicaciones. Todas las comunicaciones están basadas en SOAP, exceptuando el envío de las soluciones, las cuales son remitidas a los usuarios mediante correo electrónico. Se encuentra disponible un ejemplo de cliente desarrollado en Java en la siguiente dirección URL:

“<http://mcs.us.es:8080/ORML/servlet/ORMLMasterHTTPAccessingServlet>”

Además de la posibilidad de enviar nuevas peticiones de resolución y conversión de problemas a través de aplicaciones independientes existe la posibilidad de enviarlo a través de un navegador Web. Antes de poder usar el presente servicio Web es necesario registrarse mediante un navegador Web a través de la dirección URL siguiente:

“<http://mcs.us.es:8080/ORML/servlet/ORMLMasterHTTPRegisteringServlet>”

Una vez registrado en el servicio Web se puede acceder a través de navegador Web a las siguientes opciones:

- Consulta y modificación de los datos del usuario.
- Cancelación de peticiones enviadas.
- Ejecución de peticiones canceladas o resuelta previamente.
- Eliminación de peticiones enviadas previamente.
- Consulta del tiempo empleado en la resolución de las peticiones.
- Consulta del estado de las peticiones.
- Envío de nuevas peticiones de resolución y conversión.
- Consulta de gráficas estadísticas de resolución de las peticiones.

Para acceder a las anteriores opciones se debe usar un navegador Web y solicitar la siguiente dirección URL:

“<http://mcs.us.es:8080/ORML/servlet/ORMLMasterHTTPAccessingServlet>”

La siguiente figura muestra el acceso Web a la consulta de las peticiones del usuario, etc.

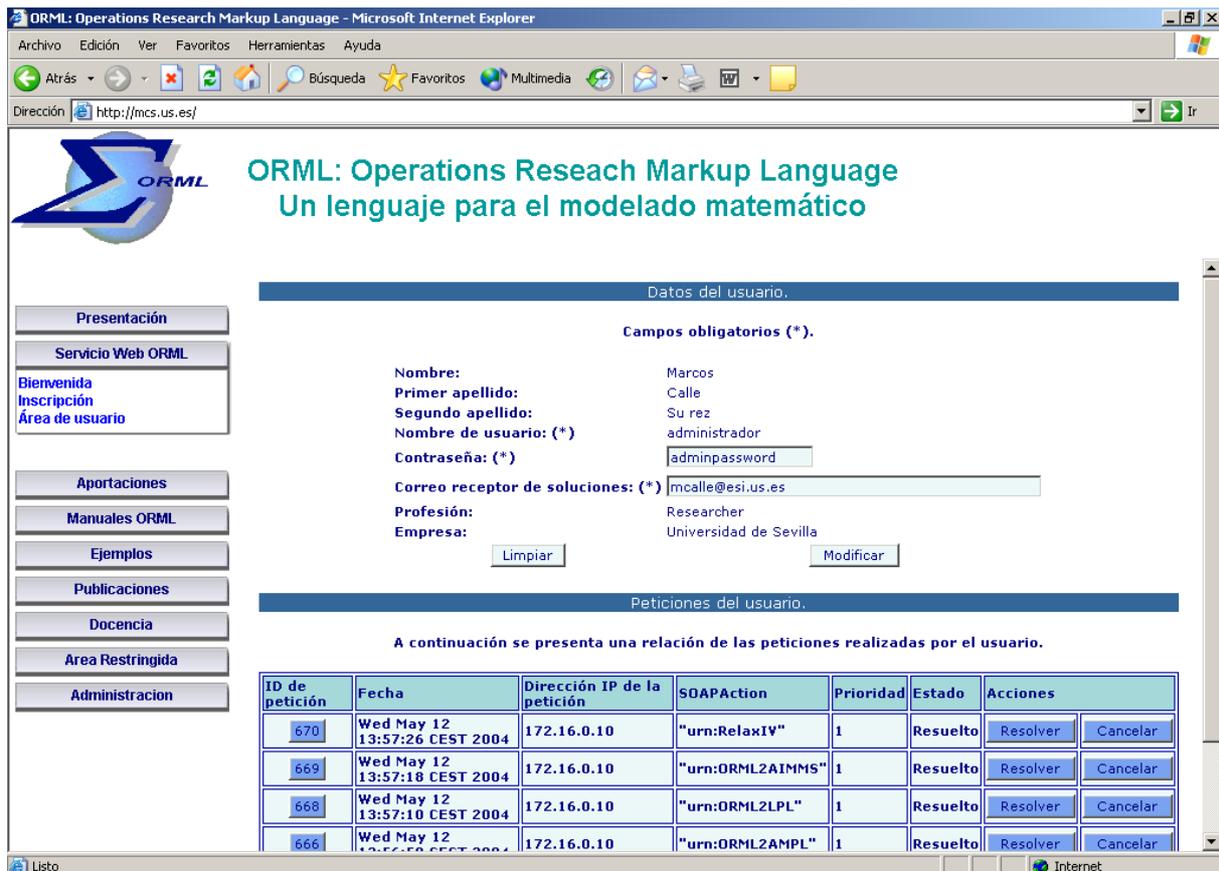


Figura 4. Arquitectura del servicio Web ORML

2.5. Descripción detallada de los servicios

Todos los servicios citados previamente están descritos usando WSDL. Dicha descripción se encuentra disponible en la siguiente dirección URL:

“http://mcs.us.es/ORML/Aportaciones/ws/orml/1.0/ORML_W_S_WSDL_1_0.zip”

Las siguientes secciones muestran y explican las comunicaciones del presente servicio Web.

2.5.1. Archivo "orml_definitions_1_0.wsdl"

Este archivo contiene las descripciones de las distintas partes que componen los mensajes SOAP. Las siguientes líneas muestran un fragmento de este archivo.

```
<definitions xmlns=http://schemas.xmlsoap.org/wsdl/
  xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:orml="http://ebiz.us.es/ns/orml/1.0"
  xmlns:flow="http://ebiz.us.es/ws/orml/network_flow"
  xmlns:hdr="http://ebiz.us.es/ws/orml/1.0/headerRequest"
  xmlns:def="http://ebiz.us.es/ws/orml/1.0/definitions"
  targetNamespace="http://ebiz.us.es/ws/orml/1.0/definitions"
  name="ORMLWS">
  <import namespace=http://ebiz.us.es/ns/orml/1.0
    location="http://ebiz.us.es/ns/orml/1.0/ORML_Model_1_0.xsd"/>
  ...
  <import .../>
  <message name="ormlModel">
```

```

    <part name="body" element="orml:model" />
  </message>
  <message name="requestHeader">
    <part name="header" element="hdr:RequestORMLHeader" />
  </message>
  ...
  <message ...> ... </message>
  <portType name="conversionFromORMLPortType">
    <operation name="ormlModelConversionOperation">
      <input message="def:ormlModel" />
    </operation>
    ...
    <operation ...> ... </operation>
  </portType>
  ...
  <portType ...> ... </portType>
</definitions>

```

El elemento <definitions> contiene los atributos necesarios para la declaración de los prefijos asociados a los espacios de nombre. Los espacios de nombre usados en el presente código corresponden a los siguientes esquemas de XML referente a WSDL, extensiones de WSDL para SOAP, ORML, datos de los problemas de flujo en redes, contenido de la cabecera de los mensajes SOAP y definiciones de los mensajes. El elemento <import> establece la localización del esquema de XML para un espacio de nombre determinado. Cada elemento <message> contiene las definiciones de los tipos de datos. El elemento <message> cuyo atributo "name" coincide con "ormModel" establece el contenido del elemento <body> en el mensaje SOAP. Este contenido consiste en un elemento <model> definido en el archivo "ORML_Model_1_0.xsd". El elemento <portType> contiene un conjunto de operaciones llevadas a cabo en un terminal específico. El elemento <operation> se compone de los mensajes de petición y respuesta. En el anterior código fuente, el elemento <portType> denominado "ormlModelConversionOperation" se compone de un único mensaje recibido por este servicio Web. La estructura de este mensaje se detalla en el anterior elemento <message>, cuyo nombre es "ormlModel".

2.5.2. Archivo "orml_services_1_0.wsdl"

Este archivo contiene el formato de los mensajes, así como las operaciones asociadas a sus correspondientes terminales. Las siguientes líneas muestran un fragmento de este archivo.

```

<definitions xmlns=http://schemas.xmlsoap.org/wsdl/
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:def="http://ebiz.us.es/ws/orml/1.0/definitions"
  xmlns:srcv="http://ebiz.us.es/ws/orml/1.0/service"
  targetNamespace="http://ebiz.us.es/ws/orml/1.0/service"
  name="ORMLWS">
  <import namespace="http://ebiz.us.es/ws/orml/1.0/definitions"
    location="http://ebiz.us.es/ws/orml/1.0/orml_definitions_1_0.wsdl" />
  <binding name="conversionFromORMLBinding"
    type="def:conversionFromORMLPortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="ormlModelConversionOperation">
      <soap:operation soapAction="urn:ORMLModel2GAMS" />
      <input message="def:ormlModel">
        <soap:header message="requestHeader" part="header" use="literal" />
        <soap:body use="literal" />

```

```

    </input>
  </operation>
  ...
  <operation ...> ... </operation>
</binding>
...
<binding ...> ... </binding>
<service name="ORMLService">
  <documentation>Last modification: 18-11-2003</documentation>
  <port name="conversionFromORML" binding="srvc:conversionFromORMLBinding">
    <soap:address
location="http://193.147.160.117:8080/ORML/servlet/ORMLMasterReceivingRequestServlet"/>
    </port>
    ...
  <port ...> ... </port>
</service>
</definitions>

```

El elemento <binding> contiene el formato de mensaje de cada operación ejecutada en el mismo terminal. Esta definición incluye el valor de la cabecera HTTP denominada "SOAPAction", el contenido del elemento <Header> y el contenido del elemento <Body>. En el anterior código fuente se muestra como el mensaje de la operación denominada "ormlModelConversionOperation" contiene los siguientes componentes: valor "urn:ebiz.us.es:ORMLModel2GAMS" asociado a la cabecera HTTP "SOAPAction", elemento <Header> definido en el elemento <message> asociado a "requestHeader" en el archivo "orml_definitions_1_0.wsdl", y contenido del elemento <Body> definido en el elemento <message> asociado a "ormlModel" en el archivo "orml_definitions_1_0.wsdl". El elemento <service> contiene la dirección URL asociada a los servicios definidos en los anteriores elementos <binding>.

2.5.3. Mensajes SOAP

Esta sección contiene un mensaje SOAP correspondiente al servicio previamente descrito denominado "conversionFromORMLBinding".

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Header>
    <RequestORMLHeader>
      <User>administrador</User>
      <Password>adminpassword</Password>
      <Priority>1</Priority>
      <ClientIP>192.168.0.102</ClientIP>
    </RequestORMLHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <model xmlns="http://ebiz.us.es/ns/orml/1.0"
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="http://ebiz.us.es/ns/orml/1.0 ../ORML-Model.xsd"
  name="Chinese Postman Problem" abbr="cpp01">
      <sets>
        <set name="I">
          <aliases>
            <alias name="i"/>

```

```

    <alias name="j"/>
  </aliases>
</set>
</sets>
...
</model>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Este mensaje SOAP contiene los elementos <Header> y <Body>. El elemento <Header> contiene la siguiente información relacionada con el usuario: nombre usuario, contraseña, prioridad de ejecución y dirección IP del equipo cliente. El elemento <Body> contiene el modelo ORML a convertir a GAMS.

3. Conclusiones y futuras ampliaciones

Este trabajo presenta un servicio Web que apoya la propuesta del lenguaje de modelado basado en XML denominado ORML (Operations Research Markup Language), dedicado a estandarizar el modelado matemático. Este servicio Web ofrece un servicio dedicado a la conversión de problemas codificados en ORML hacia otros lenguajes de modelado matemático existente (GAMS, AMPL, MPL, LPL y AIMMS). También se ofrece otro servicio consistente en la resolución de problemas, actualmente limitado a problemas de flujo en redes.

Acerca de las futuras ampliaciones del presente trabajo podemos mencionar las siguientes:

- ORML incluirá entre sus interfaces de conversión algunos lenguajes adicionales, como por ejemplo, OPL (Optimization Programming Language), Hentenryck, (1999).
- Conversión de problemas codificados en alguno de los lenguajes de modelado existentes (GAMS, AMPL, MPL, LPL y AIMMS) hacia ORML.
- Inclusión de nuevos elementos para representar otros tipos de problemas, como por ejemplo problemas combinatorios de optimización.
- Desarrollo de una interfaz gráfica que facilite la creación de modelos en ORML.

Referencias

- Bisschop, J., Roelofs, M., (2001). *AIMMS: The Language Reference*. Paragon Decision Technology B.V.
- Box D., Ehnebuske D., Kakivaya G., Layman A., Mendelsohn N., Frystyk Nielsen H. F., Thatte S., Winer D., (08 May 2000): Simple Object Access Protocol (SOAP) 1.1. World Wide Web Consortium, Fribourg, <http://www.w3.org/TR/SOAP> [23th November 2003]
- Brooke, A., Kendrick, D., Meeraus, A., (1988). *GAMS : a user's guide*. Scientific Press
- Calle M., Lozano S., Smith K., Villa G., (2003). An XML Schema Definition for an Operations Research Modeling Language. *Frontiers in Artificial Intelligence and Applications*, Vol. 104. IOS Press
- Christensen E., Curbera F., Meredith G., Weerawarana S., (15 March 2001): Web Services Description Language (WSDL) 1.1. World Wide Web Consortium, Fribourg, <http://www.w3.org/TR/wsdl> [23th November 2003]
- Ezechukwu O., Maros I., (April 2003). *OOF: Open Optimization Framework*. Department of Computing, Imperial College London

Fourer, R., Gay, D.M., Kernighan, B.W., (1993). *AMPL: a modeling language for mathematical programming*. Danvers, MA Boyd & Fraser
Hentenryck P.V., (1999). *The OPL Optimization Programming Language*. The MIT Press,
Hürlimann T., (2002). *Reference Manual for the LPL Modeling Language*. Version 4.43,
Departement for Informatics, Fribourg,
Maximal Software, Inc. *MPL Manual*.