

Optimizando la comunicación entre las Aplicaciones de Clientes y el motor del Modelo de Referencia de Workflow con Servicios Web

Daniel Riesco

Departamento de Informática, Universidad Nacional de San Luis
San Luis, C.P. 5700, Argentina
driesco@unsl.edu.ar

Paola Martellotto, Marcela Daniele

Departamento de Computación, Universidad Nacional de Río Cuarto
Río Cuarto, C.P. 5800, Argentina
paola@dc.exa.unrc.edu.ar, marcela@dc.exa.unrc.edu.ar

Abstract

This work proposes to optimize the communication between the Workflow Client Application Interface and the Workflow Engine of the Workflow Reference Model, defining a specification with Web Services. The Workflow Reference Model, developed by the Workflow Management Coalition (WfMC), shows an architecture that standardizes the development of Workflow applications, in order to allowing interoperability among them. It defines five interfaces and specifies a set of WAPIs (Workflow Application Programming Interfaces) that define the functions of these interfaces like calls to APIs in a third-generation language, forcing to know specifically where an application is located to be able to invoke it. The Web Services are autonomous and independent services that are offered by means of the Web. Its main benefit is that they allow that the applications more modular, facilitating the reusability in different platforms and programming languages. This specification with Web Services improves the communication of Client's Applications with the Workflow Engine, favoring the user since he doesn't need to know the exact location of the application to invoke, and to the applications because they can vary its location in the web without implying any change in its invocation.

Keywords: Workflow Reference Model, Workflow Client Application Interface, Web Service.

Resumen

Este trabajo propone optimizar la comunicación entre la Interfaz de Aplicaciones de Cliente y el motor Workflow del Modelo de Referencia de Workflow, definiendo una especificación con Servicios Web. El Modelo de Referencia de Workflow, desarrollado por la Workflow Management Coalition (WfMC), muestra una arquitectura que estandariza el desarrollo de aplicaciones Workflow, con el objeto de permitir la interoperabilidad entre los mismos. Define cinco interfaces y especifica un conjunto de WAPIs (Workflow Application Programming Interfaces) que definen las funciones de dichas interfaces como llamadas a APIs en un lenguaje de tercera generación, obligando a conocer específicamente dónde está ubicada una aplicación para poder invocarla. Los Servicios Web son servicios autónomos e independientes que se ofrecen mediante la web. Su principal beneficio es que permiten que las aplicaciones sean más modulares y desacopladas, facilitando su reutilización en distintas plataformas o lenguajes de programación. Esta especificación con Servicios Web mejora la comunicación de las Aplicaciones de Cliente con el motor Workflow, favoreciendo al usuario dado que no necesita conocer la ubicación exacta de la aplicación a invocar, y a las aplicaciones porque pueden variar su ubicación en la red sin implicar ningún cambio en su invocación.

Palabras Clave: Modelo de Referencia de Workflow, Interfaz de Aplicaciones de Cliente, Servicio Web.

1. Introducción

La globalización y los cambios de paradigmas empresariales, la evolución de las tecnologías de la información y la política liberal imperante hoy en el mundo, ubican a las organizaciones en el juego de la competitividad internacional. Las organizaciones se orientan a ser más horizontales hacia el enfoque de redes de procesos, los que deben ser diseñados de principio a fin empleando nuevas tecnologías. Un proceso de negocio es un conjunto de tareas relacionadas lógicamente que se ejecuta con la intención de obtener un resultado de negocio particular, el cual incluye recursos humanos así como los recursos materiales con el objetivo de producir un beneficio para la organización. El modelado de procesos de negocio permite visualizar las tareas, actividades y flujos, así como las unidades organizacionales diferentes que son afectadas por el proceso.

Los sistemas de Administración de Workflow se están desarrollando en las empresas como una nueva forma de organizar y administrar la información, ayudando a automatizar los procesos de negocio. Tales sistemas pueden predefinir un procedimiento de trabajo con la información relevante, el rol de los participantes en cada paso de trabajo, y la aplicación de software requerida para procesar cada paso. Cada nuevo caso se asigna automáticamente a los participantes en una secuencia concreta, la información se entrega a la gente que la necesita y las aplicaciones se ejecutan cuando es necesario.

Para poder tener cierto nivel de interoperabilidad entre los diversos productos de Workflow, es necesario definir un conjunto de interfaces y formatos para el intercambio de datos entre dichos componentes.

La Workflow Management Coalition (WfMC) ha desarrollado un Modelo de Referencia de Workflow [7] identificando las interfaces con las estructuras genéricas de las aplicaciones de Workflow, para permitir a los productos comunicarse a distintos niveles. En particular, para permitir la interacción de los usuarios con el motor de Workflow utiliza una Worklist, que es manejada por un administrador. La Interfaz de las Aplicaciones de Cliente es la encargada de manejar la interacción entre el motor de Workflow y el administrador de la Worklist. La WfMC ha especificado un conjunto de APIs (Application Programming Interfaces) para la administración de Workflows [4], las cuales están soportadas por los productos Workflow y se denominan Workflow Application Programming Interfaces (WAPIs).

El Modelo de Referencia de Workflow [7] fue desarrollado desde estructuras genéricas de aplicaciones de Workflow, identificando las interfaces con estas estructuras, para permitir a los productos comunicarse a distintos niveles. Todos los sistemas de Workflow contienen componentes genéricos que interactúan de forma predefinida.

Por otro lado, los Servicios Web son aplicaciones auto-contenidas, auto-descriptas que pueden ser publicadas, localizadas e invocadas a través de la Web, sin la necesidad de conocer la ubicación exacta de los mismos [1]. Para la descripción de un Servicio Web se utiliza WSDL (Web Service Description Language) [4] [5] basado en XML [2]. Además, es necesario describir los mensajes entre las aplicaciones y el servicio web, y la forma en que los mismos serán transportados a través de la web. SOAP (Simple Object Access Protocol) [3] es el protocolo más conocido basado en mensajes, que es utilizado para describir la interacción de las aplicaciones con los web services. Por su parte, el protocolo de transporte más usado es HTTP (Hiper Text Transport Protocol). Y por último, es necesario registrar y localizar el servicio web, para lo cual se define un directorio de Servicios Web distribuido y basado en Web que permite que se listen, busquen y descubran. Por lo general este directorio es definido UDDI (Universal Description, Discovery and Integration) [6]. El uso de estos protocolos estándares permite lograr la interoperabilidad en ambientes heterogéneos, con independencia del Sistema Operativo, lenguaje de programación, etc.

Las implementaciones conocidas de la Interfaz de las Aplicaciones de Cliente [8] definen las funciones de las interfaces y su especificación, como llamadas a APIs en el lenguaje "C", y para la invocación de una aplicación se requiere conocer específicamente su ubicación. En este trabajo se propone una

especificación de dicha Interfaz utilizando Web Services, con el objetivo de que el usuario del Workflow no necesite conocer la ubicación de la aplicación que desea invocar, y que cualquier aplicación pueda cambiar su ubicación en la red sin que esto implique ningún cambio en su invocación.

2. Estado del Arte

En [9] la WfMC presenta una propuesta de especificación de las Interfaces 2 y 4 basada en el uso de IDL y bindings con OLE como alternativas a las especificaciones existentes C y MIME. Los primeros trabajos de la WfMC (sobre WAPIs) se concentran en la definición de las funciones de las interfaces y su especificación como llamadas a APIs en el lenguaje "C". La especificación de la interoperabilidad se desarrolla subsecuentemente usando IDL para la especificación abstracta y bindings concretos basados en MIME, para usar vía Internet. En esta propuesta el manejo de las consultas WAPI se reemplaza por el uso de una colección de objetos OLE y se define un filtro para reemplazar el filtro WAPI. El IDL binding de los estándares de la OMG define un modelo de objetos que combina las interfaces 2, 4 y 5, y también direcciona el área de aplicaciones invocadas, donde las aplicaciones se asumen como objetos de negocio. Se define una especificación jFLOW usando la notación UML.

En [10] se presenta la especificación JointFlow, adoptada por 19 compañías en respuesta a OMG's Workflow Management Facility RFP. Esta especificación se basa en el trabajo de la Workflow Management Coalition y define las interfaces que soportan la interacción en tiempo de ejecución entre los componentes de Workflow, permite la interoperabilidad de los componentes de negocio a través de los dominios de negocio y permite el monitoreo de los procesos. La especificación JointFlow focaliza en definir un conjunto de interfaces que permiten la interoperabilidad de los componentes de Workflow, soporta el monitoreo y la asignación de recursos. La especificación JointFlow se utiliza para realizar aplicaciones Workflows distribuidas.

En [11] los autores presentan una especificación utilizando Grid Computing. En los WFMS (Workflow Management Systems) tradicionales los procesos son diseñados por herramientas de definición de procesos y ejecutados por el motor Workflow, en el mismo WFMS. Las tareas son desarrolladas por los usuarios finales o las aplicaciones. Los procesos solamente pueden ser ejecutados por motores que son accedidos por usuarios específicos y aplicaciones específicas. Por lo tanto, utilizar WFMS es restrictivo de las localizaciones. En esta propuesta las tareas son desarrolladas por los servicios Grid (los cuales se basan en un conjunto de interfaces estándar). Los servicios pueden ser accedidos por cualquier aplicación de acuerdo a estos estándares.

En [12] se introduce un agente de composición de Workflow, que es capaz de componer Workflow de Servicios Web, usar descripciones semánticas de los Servicios Web y encontrar Servicios Web para un Workflow. Se muestra cómo un Servicios Web se puede componer utilizando ontologías de Servicios Web semánticos, que se refiere a definir la semántica de un Servicios Web, es decir, su significado, más que sus parámetros de entrada y salida. Con esta tecnología se presenta un modelo para componer Workflow de Servicios Web. El modelo para la composición de Workflow describe qué ocurre cuando se crea una nueva instancia de Servicios Web. La instancia se publica en un directorio.

En [13] se propone encapsular la funcionalidad de las organizaciones en interfaces apropiadas y publicarlas como Servicios Web. Se espera que los Servicios Web puedan integrarse como parte de los procesos web. Pero esta integración es dificultosa dada la alta heterogeneidad, autonomía y distribución de la web. Una solución es el uso de ontologías. También es esencial para los Servicios Web soportar todas las fases del ciclo de vida de un proceso web. Se describe cómo el aplicar semántica a cada paso del ciclo de vida de un Proceso Web Semántico puede ayudar al rehúso, la integración y escalabilidad.

En [14] se presenta un lenguaje de Workflow que usa los Servicios Web como componentes, una arquitectura para un ambiente en tiempo de ejecución para este lenguaje, y aprovecha las ventajas de la utilización de esta clase de tecnología. La idea es desarrollar procesos de negocio formados por una composición de servicios que están disponibles en una red de computadoras. La tecnología Workflow se usa para coordinar las interacciones entre los Servicios Web. Los Servicios Web representan los pasos lógicos que componen un Workflow. El lenguaje propuesto AELCWS no soporta la interacción directa con las personas durante la ejecución de un proceso, de modo que la Interfaz de Aplicaciones de Cliente no está definida ni implementada. La Interfaz de Invocación de las Aplicaciones sí es soportada por este lenguaje, a través de la invocación de los servicios que conforman la definición del proceso.

3. WorkFlows

Workflow es la automatización de los procesos de negocio donde los documentos, la información y tareas se pasan entre los participantes del sistema de acuerdo a un conjunto de reglas previamente establecidas [7].

Por su parte, un WFMS es un sistema que define, crea y administra la ejecución de Workflow a través del uso de software que se ejecuta sobre uno o más motores Workflow, los que interpretan la definición de los procesos, interactúan con los otros participantes del Workflow e invocan herramientas y aplicaciones [7].

Las tendencias imperantes en el mundo de los negocios obligan a las empresas a repensarse a sí mismas y rediseñar sus soportes tecnológicos, con el fin de: aumentar la productividad, mejorar la calidad, mejorar el servicio al cliente, reducir los costos y adaptarse a un entorno cambiante. En un WFMS:

- El trabajo no se extravía o detiene,
- Los gerentes pueden enfocarse en el personal y problemas de negocio,
- Los procedimientos son formalmente documentados,
- La mejor persona (o máquina) es asignada para hacer cada caso,
- El procesamiento paralelo es práctico.

El trabajo es realizado por el mejor participante, es decir, se distribuye el trabajo, y existe un supervisor quien puede influir en esta asignación automática. Además, el sistema necesita conocer qué trabajo espera ser asignado.

4. El Modelo de Referencia de la WfMC

El Modelo de Referencia de Workflow [7] identifica las interfaces con estructuras genéricas de aplicaciones, para permitir a los productos comunicarse a distintos niveles. Todos los sistemas de Workflow contienen componentes genéricos que interactúan de forma predefinida. La figura 1 muestra el Modelo de Referencia de Workflow propuesto por la WfMC.

En este modelo adoptado hay una separación entre los procesos y el control de la lógica de las actividades. Esta lógica está dentro del Servicio de Representación de Workflow (Workflow Enactment Service). Esta separación permite la integración de las diversas herramientas con una aplicación particular.

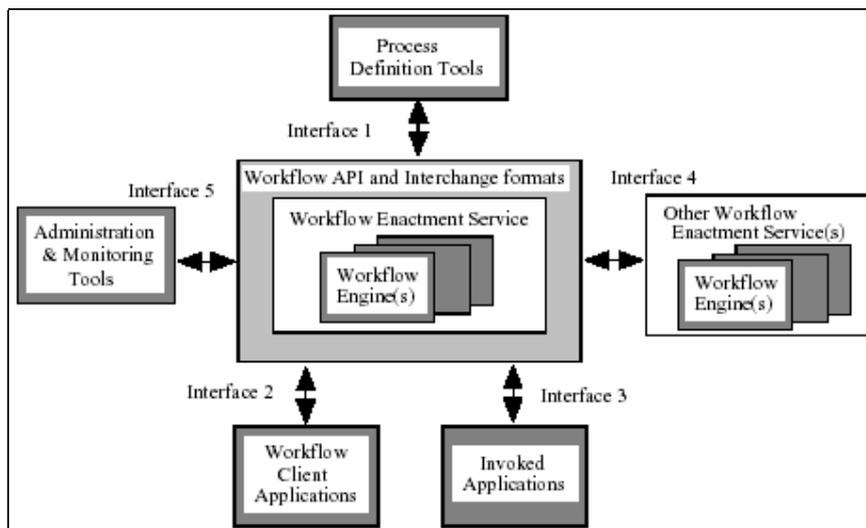


Figura 1. Modelo de Referencia de Workflow. Componentes e Interfaces.

El Servicio de Representación de Workflow interpreta la descripción de procesos y controla las diferentes instancias de los procesos, secuencia las actividades, adiciona ítems a la Lista de Trabajo (Worklist) de los usuarios, e invoca aplicaciones necesarias. La interacción del Servicio de Representación de Workflow con los recursos externos se da por una de las dos interfaces siguientes:

- La interfaz de las Aplicaciones de los Clientes, a través de la cual el motor de Workflow interactúa con el administrador de la Lista de Trabajo, responsable de organizar el trabajo por intermedio de un recurso de usuario.
- La interfaz de las Aplicaciones Invocadas, la cual le permite al motor de Workflow activar una herramienta para realizar una actividad particular. Esta interfaz podría estar basada en un servidor, es decir no existe la interacción con el usuario.

La Lista de Trabajo permite controlar la interacción con los usuarios. El motor deposita en la Lista de Trabajo los ítems a ser ejecutados por cada usuario.

El Administrador de la Lista de Trabajo maneja la interacción entre los participantes del Workflow y el Servicio de Representación de Workflow, vía la Lista de Trabajo. Soporta un amplio rango de interacción con otras aplicaciones clientes.

5. Servicios Web

Los Servicios Web son aplicaciones auto-contenidas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web. Una vez desarrolladas, otras aplicaciones (y otros Servicios Web) pueden descubrirlas e invocar el servicio dado [1].

El desarrollo y la programación de sistemas orientados a objetos o componentes han llevado a lo largo del tiempo a tener la necesidad de reutilizarlos en diferentes proyectos. Ya sean componentes desarrollados por uno mismo o componentes desarrollados por terceros. Hasta la existencia de los Servicios Web esta reutilización se limitaba a un lenguaje de programación o una plataforma en particular. Por lo tanto, el uso de los Servicios Web facilita la reutilización de una aplicación en distintas plataformas o lenguajes ya sea para uso personal en distintos proyectos, para comercializarlos o adquirir prestaciones de terceros.

De la misma forma que anteriormente se incluía en las aplicaciones referencias a otras librerías como ser DDLs, ahora se podrán referenciar funciones que se estarán ejecutando en otra computadora o servidor sin importar en qué están programadas ni en qué plataforma están corriendo.

Un Servicio Web se basa en las siguientes tecnologías:

- Para la descripción del servicio: un formato que describe la interfaz del componente (sus métodos y atributos) basado en XML [2]. Por lo general este formato es WSDL (Servicio Web Description Language) [4] [5]
- Para la representación de los mensajes: un protocolo de aplicación basado en mensajes que permite que una aplicación interactúe con el Servicios Web. Por lo general este protocolo es SOAP (Simple Object Access Protocol) [3]
- Para el transporte de los mensajes: un protocolo de transporte que se encargue de transportar los mensajes por Internet. Por lo general este protocolo de transporte es HTTP (Hiper Text Transport Protocol)
- Para el registro y la localización del servicio: un directorio de servicios web distribuido y basado en Web que permite que se listen, busquen y descubran este tipo de software. Por lo general este directorio es UDDI (Universal Description, Discovery and Integration) [6].

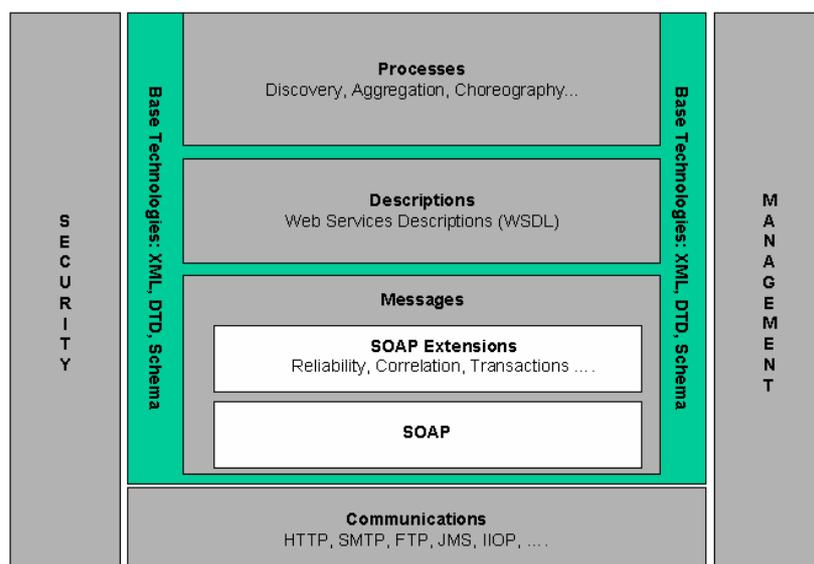


Figura 2. Arquitectura de los Servicios Web.

El uso de estos protocolos estándar es necesario para lograr la interoperabilidad en ambientes heterogéneos, con independencia del Sistema Operativo, el lenguaje de programación, etc.

WSDL [4] [5] es el lenguaje utilizado para describir un Servicio Web. WSDL describe un Servicio Web como un conjunto de puntos finales de comunicación (métodos) capaces de intercambiar mensajes. Es un archivo XML que describe el conjunto de métodos expuestos por un Servicio Web. Todo documento WSDL está compuesto por un elemento raíz llamado *definitions*, que a su vez está compuesto por los siguientes elementos:

- `types`: define qué tipos de datos serán transmitidos.
- `message`: define qué mensajes serán transmitidos.
- `portType`: define qué operaciones (funciones) serán soportadas.
- `binding`: establece cómo será transmitido el mensaje. Establece los detalles de la especificación SOAP.
- `service`: informa dónde se localiza el servicio.

WSDL se basa en el lenguaje XML (Extensible Markup Language) [2]. Este es el lenguaje utilizado para definir el formato de documentos o mensajes. XML comprende el uso de etiquetas denominadas tags que identifican los contenidos de un documento, y al hacerlo, los describen. Una etiqueta XML identifica información dentro de un documento, como así también la estructura de dicha información.

Los documentos XML poseen una estructura bien-formada y generalmente están asociados con un esquema (scheme) que especifica qué etiquetas están permitidas dentro de un documento, la estructura de esas etiquetas, y otras reglas relacionadas, tales como el tipo de dato que se espera dentro de una etiqueta.

Por otro lado, SOAP [3] es el protocolo utilizado en los Servicios Web para el transporte de los mensajes. Los mensajes SOAP están compuestos por un *tag* principal llamado *Envelope*, que está dividido en una cabecera o *Header* y un cuerpo o *Body*. Dentro del elemento *Body* estarán los elementos correspondientes al Web Method y además puede haber o no un elemento en común llamado *fault*, que indica que ha ocurrido un error y la razón de este. Por lo tanto, la definición de un Servicio Web implica, entonces, construir un documento WSDL que contenga los elementos mencionados anteriormente.

6. Especificación de la Interfaz de las Aplicaciones de Clientes con Servicios Web

Como ya se mencionó, la Interfaz de las Aplicaciones de Clientes permite la interacción entre las aplicaciones clientes y el motor de Workflow. Para sostener dicha interacción se utiliza una Lista de Trabajo, que almacena la información de las aplicaciones que se deben invocar y posee un administrador de la misma. La Lista de Trabajo puede contener ítems relacionados con diferentes instancias de un proceso o ítems de diferentes procesos. El administrador puede interactuar con diferentes motores.

Las APIs asociadas a esta interfaz proveen un nivel básico de funcionalidad para soportar la invocación de aplicaciones. Dichas APIs se pueden consultar en [8].

Las funciones de la Lista de Trabajo proveen información a los participantes del Workflow sobre los trabajos que ellos tienen asignados. Como lo describe el Modelo de Referencia de la WfMC, un proceso consiste de un conjunto de actividades conectadas de forma tal de controlar el secuenciamiento de la invocación a aplicaciones. Una actividad está asociada a una o más aplicaciones a ser invocadas y, en tiempo de ejecución, se asocia con la/s persona/s que han sido asignadas para realizar el trabajo.

Dependiendo de la implementación del Workflow, un participante de Workflow puede tener asignadas una o más piezas de trabajo al mismo tiempo. Cada pieza de trabajo se denomina Item de Trabajo (work item) y la colección de todos los Items de Trabajo asignados a un participante se denomina Lista de Trabajo (worklist) del participante de Workflow.

Cuando se define un proceso, la actividad es el constructor usado para definir una pieza de trabajo a realizar. En tiempo de ejecución, cuando la actividad esta lista para ser ejecutada y uno o más candidatos se asignan para realizar el trabajo, se crea un Item de Trabajo y se ubica en la Lista de Trabajo del participante. Aunque ambos representan una pieza de trabajo, existen en diferentes puntos en el tiempo, y puede haber más de un Item de Trabajo para una misma actividad. Para especificar la Interfaz de Aplicaciones de Cliente del Modelo de Referencia de Workflow con Servicios Web se procede a especificar cada una de las funciones definidas con WAPIs en dicho modelo. El caso de estudio que se presenta en este trabajo desarrolla la especificación de la función *WMOpenWorkList*, la cual provee información a los participantes del Workflow sobre los trabajos que tienen asignados.

6.1. La función *WMOpenWorkList*

La función *WMOpenWorkList* especifica y ejecuta una consulta para producir la Worklist que cumple con el criterio de filtro de la consulta.

El comando provee la capacidad de retornar la lista de work ítems asignados a una participante de Workflow particular o a un grupo de trabajo. El solicitante puede realizar la consulta a nombre de sí mismo o quizá esperar para conocer qué trabajos le han sido asignados a él particular o a un grupo de trabajo.

La consulta retornará la lista de todos los work items que coinciden con los valores específicos del atributo. El comando también puede retornar, opcionalmente, la cuenta total de los work ítems disponibles. La WAPI de esta función se puede consultar en [8].

6.2. Definición del Servicio Web para la función *WMOpenWorkList*

Para la función *WMOpenWorkList*, uno de los elementos a definir son los tipos que usará el servicio. Estos tipos se declaran dentro del elemento <types>. Se utiliza un esquema XML para definir los tipos que son más complejos.

```

<types>
  <xs:schema
    ...
    <xs:element name="psession_handle" type="Tpsession_handle"/>
    <xs:complexType name="Tpsession_handle">
      <xs:sequence>
        <xs:element name="session_id" type="xs: String"/>
        <xs:element name="pprivate" type="xs: String"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="pworklist_filter" type="Tpworklist_filter"/>
    <xs:complexType name="Tpworklist_filter">
      <xs:sequence>
        <xs:element name="sqlString" type="xs: String"/>
        <xs:element name="attributeName" type="xs: String"/>
        <xs:element name="comparison" type="xs: int"/>
        <xs:element name="attributeValue" type="xs: boolean"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="count_flag" type="Tcount_flag"/>
    <xs:complexType name="Tcount_flag">
      <xs:sequence>
        <xs:element name="countFlag" type="xs: boolean"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="pquery_handle" type="Tpquery_handle"/>
    <xs:complexType name="Tpquery_handle">
      <xs:sequence>
        <xs:element name="ActivityDefinitionId " type="xs: String"/>
        <xs:element name="ActivityInstanceId" type="xs: String"/>
        <xs:element name="Id" type="xs: String"/>
        <xs:element name="Name" type="xs: String"/>
        <xs:element name="Participant" type="xs: String"/>
        <xs:element name="Priority" type="xs: int"/>
        <xs:element name="ProcessInstanceId" type="xs: int"/>
        <xs:element name="State" type="xs: String"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>

```

```

<xs:element name="pcount" type="Tpcount"/>
<xs:complexType name="Tpcount">
  <xs:sequence>
    <xs:element name="count" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="WMInvalidSessionHandle" type="TWMLInvalidSessionHandle"/>
<xs:complexType name="TWMLInvalidSessionHandle">
  <xs:sequence>
    <xs:element name="excep" type="xs:String"/>
    <xs:element name="error" type="xs:String"/>
    <xs:element name="message" type="xs:String"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="WMInvalidFilter" type="TWMLInvalidFilter"/>
<xs:complexType name="TWMLInvalidFilter">
  <xs:sequence>
    <xs:element name="excep" type="xs:String"/>
    <xs:element name="error" type="xs:String"/>
    <xs:element name="message" type="xs:String"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</types>

```

WSDL permite separar la descripción de la funcionalidad abstracta de un Servicio Web de los detalles concretos de cómo y dónde la funcionalidad es ofrecida. Una interfaz WSDL define la interfaz abstracta de un Servicio Web como un conjunto de operaciones abstractas.

```

<interface name = "WMOpenWorkListInterface" >
  <fault name = "WMOpenWorkList_InvalidSessionHandle"
    element = "wfms: WMInvalidSessionHandle"/>
  <fault name = "WMOpenWorkList_InvalidFilter"
    element = "wfms: WMInvalidFilter"/>

  <operation name="opWMOpenWorkList"
    pattern=http://www.w3.org/2004/03/wsd/in-out>
    <input messageLabel="In" element="wfms: psession_handle" />
    <input messageLabel="In" element="wfms: pworklist_filter" />
    <input messageLabel="In" element="wfms: count_flag" />
    <output messageLabel="Out" element="wfms: pquery_handle" />
    <output messageLabel="Out" element="wfms: pcount" />
    <outfault ref="tns: WMOpenWorkList_InvalidSessionHandle" messageLabel="Out"/>
    <outfault ref="tns: WMOpenWorkList_InvalidFilter" messageLabel="Out"/>
  </operation>
</interface>

```

Otro elemento a especificar es el binding, que permite detallar cómo los mensajes pueden ser intercambiados. Especifica detalles del formato concreto de mensajes y del protocolo de transmisión para una interfase, y provee tales detalles para cualquier operación y falla en la interfaz.

```

<binding name="WMOpenWorkListSOAPBinding"
  interface="tns: WMOpenWorkListInterface"
  type=http://www.w3.org/2006/01/wsdl/soap
  wssoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP">

  <operation ref="tns: opWMOpenWorkList"
    wssoap:mep=.../>

  <fault ref="tns: WMOpenWorkList_InvalidSessionHandle"
    wssoap:code="soap:Sender"/>

  <fault ref="tns: WMOpenWorkList_InvalidFilter"
    wssoap:code="soap:Sender"/>
</binding>

```

Finalmente, resta definir el servicio WMOpenWorkList. Esta definición implica especificar dónde el servicio puede ser accedido, mediante el uso del elemento service. Un servicio WSDL especifica una interfaz simple que soportará el servicio, y una lista de ubicación de puntos extremos (endpoints) donde ese servicio puede ser accedido.

```

<service name="WMOpenWorkListService"
  interface="tns: WMOpenWorkListInterface">
  <endpoint name="WMOpenWorkListEndpoint"
    Binding = "tns: WMOpenWorkListSOAPBinding"
    address = .../>
</service>
</description>

```

De esta manera quedan especificados los elementos types, interfaz, binding y service para la definición de la operación WMOpenWorkList, de la Interfaz de Aplicaciones de Clientes del Modelo de Referencia de Workflow utilizando SOAP y WSDL. Esto permite definir una nueva clase de aplicaciones que utilizan los servicios web distribuidos por la red, aprovechando las ventajas de la interoperabilidad.

7. Conclusiones

Los Sistemas de Administración de Workflow se desarrollan en las empresas con el objetivo de automatizar los procesos de trabajo en las oficinas. Tales sistemas pueden predefinir un procedimiento de trabajo con la información relevante, el rol de los participantes en cada paso de trabajo, y la aplicación de software requerida para procesar cada paso. Los Sistemas de Administración de Workflow realizan esta administración automáticamente y así, el trabajo se completa de una forma más eficiente.

La Workflow Management Coalition ha desarrollado un Modelo de Referencia de Workflow para permitir cierto nivel de interoperabilidad entre los diversos productos de Workflow. Las WAPIs para la administración de Workflows desarrolladas por la WfMC definen las funciones de las interfaces y su especificación como llamadas a APIs en el lenguaje "C". Esta implementación requiere que la invocación de una aplicación implique conocer específicamente dónde está ubicada la misma. Los beneficios que otorgan los Servicios Web pueden ser aprovechados para mejorar esta implementación, facilitando la comunicación del Workflow con las aplicaciones. Con el uso de Servicios Web se logra

que el Workflow se comporte internamente de forma distribuida, es decir, que no necesite conocer dónde están las aplicaciones para invocarlas. Simplemente, requiere servicios y hay aplicaciones que le proveen dichos servicios.

Para especificar la Interfaz de Aplicaciones de Cliente del Modelo de Referencia de Workflow con Servicios Web se procede a especificar cada una de las funciones definidas con WAPIs en dicho modelo. En este trabajo se especificó la función *WMOpenWorkList* definiendo cada uno de los elementos que componen un Servicio Web. De manera análoga, las demás funciones de esta Interfaz pueden ser especificadas con Servicios Web, logrando una especificación independiente del lenguaje de programación, y de la plataforma subyacente.

8. Referencias Bibliográficas

- [1] World Wide Web Consortium. Web Service Architecture. <http://www.w3.org/TR/ws-arch/>. Último acceso Mayo 2007.
- [2] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Second Edition). <http://www.w3.org/TR/xml>. Último acceso Mayo 2007.
- [3] World Wide Web Consortium. SOAP Version 1.2 Part 1: Messaging Framework. <http://www.w3.org/TR/soap12-part1/>. Último acceso Mayo 2007.
- [4] World Wide Web Consortium. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. <http://www.w3.org/TR/wsdl20-primer>. Último acceso Mayo 2007.
- [5] World Wide Web Consortium. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. <http://www.w3.org/TR/wsdl20>. Último acceso Mayo 2007.
- [6] OASIS. UDDI Version 3.0.2. http://uddi.org/pubs/uddi_v3.htm. Último acceso Mayo 2007.
- [7] Workflow Management Coalition. The Workflow Reference Model. WfMC-TC00-1003. <http://www.wfmc.org/standards/referencemodel.htm>. Último acceso Mayo 2007.
- [8] Workflow Management Coalition. Programming Interface 2&3 Specification. WfMC-TC-1009. V2.0. <http://www.wfmc.org/standards/publicdocuments.htm>. Último acceso Mayo 2007.
- [9] Workflow Management Coalition. A Common Object Model Discussion Paper. WfMC-TC10-22. http://www.wfmc.org/standards/docs/TC-1022_common_Object%20Model_Paper.pdf. Último acceso Mayo 2007.
- [10] Schmidt M.T., Building Workflow Business Objects. IBM Software Group OOPSLA'98 Business Object Workshop IV.
- [11] Yang M., Liang H., Xu B., S-WFMS: A service-based WFMS in Grid Environment. Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05). IEEE. 2005.
- [12] Laukkanen M., Helin H.: Composing Workflows of Semantic Web Services. Workshop on Servicios Web and Agent-based Engineering. AAMAS'2003. Melbourne, Australia. 14/15 de Julio de 2003.
- [13] Cardozo J., Shelt A., Introduction to Semantic Servicios Web and Web Process Composition. Publication of LSDIS. Large Scale Distributed Information Systems. University of Georgia. Computer Science Department.
- [14] Hiane da S. Maciel L. A., Toshiro Yano E.: Uma Linguagem de WF Para Composicao de Web Services. XIX Simpósio Brasileiro de Engenharia de Software. Uberlandia, MG, Brasil. 2005.