

Selección de aplicaciones invocadas por el motor Workflow basada en Servicios Web y reglas de Transformación de Grafos

Paola Martellotto¹ Marcela Daniele¹ Daniel Riesco²

¹ Universidad Nacional de Río Cuarto - Facultad de Ciencias Exactas, Fco-Qcas y Naturales -
Dpto. de Computación - Te/Fax: + 54 (0) 358 – 4676235 - {paola, marcela}@dc.exa.unrc.edu.ar

² Universidad Nacional de San Luis - Dpto de Informática - Te: + 54 (0) 2652 – 424027 driesco@unsl.edu.ar

Resumen

Los Sistemas de Administración de Workflow permiten automatizar la ejecución de los procesos de negocio sobre uno o más motores Workflow. Para lograr su comunicación con las aplicaciones externas, la WfMC definió la Interfaz de Aplicaciones Invocadas, la cual requiere que la información acerca de la aplicación y su invocación sea codificada en tiempo de desarrollo, y no sea posible en tiempo de ejecución acceder a otra aplicación diferente que ofrezca el mismo servicio.

En trabajos anteriores se propuso una alternativa para optimizar la comunicación del motor workflow con las aplicaciones externas, permitiendo invocar una aplicación a partir de los servicios web disponibles que satisfacen su requerimiento. Pero UDDI, protocolo utilizado para registrar y localizar los servicios web, presenta algunos inconvenientes al momento de seleccionar un servicio, ya que el registro del mismo no tiene en cuenta su semántica, dificultando así el descubrimiento de más de un servicio que satisface un mismo requerimiento.

Este trabajo propone utilizar reglas de transformación de grafos para especificar los servicios web y optimizar la selección de la aplicación más adecuada, según lo determine el motor Workflow cuando realiza la invocación. Las reglas de transformación de grafos permiten obtener una especificación semántica precisa de un servicio web, y establecer una correspondencia entre los requerimientos del usuario y los servicios web disponibles.

Palabras Clave: Workflow, Modelo de Referencia de Workflow, Interfaz de Aplicaciones Invocadas, Servicios Web, Reglas de Transformación de Grafos

1. Introducción

Los Sistemas de Administración de Workflow describen y automatizan los procesos de negocio de una organización. La WfMC (Workflow Management Coalition) define el Modelo de Referencia de Workflow [3] especificando cinco interfaces, que permiten a las aplicaciones del Workflow la comunicación a distintos niveles. En particular, la Interfaz de las Aplicaciones Invocadas [4] se define para la invocación de las aplicaciones externas, y requiere conocer la información acerca de la aplicación y su invocación en tiempo de desarrollo. Por esta razón, se hace imposible en tiempo de ejecución acceder a otra aplicación diferente que ofrezca el mismo servicio.

Los Servicios Web son aplicaciones auto-contenidas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web, sin la necesidad de conocer la ubicación exacta de los mismos [1]. Con el propósito de mejorar la selección de aplicaciones en tiempo de ejecución, en [5] se plantea una especificación de la Interfaz de Aplicaciones de Clientes utilizando Servicios Web, 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.

En la actualidad, se utiliza el protocolo UDDI (Universal Description, Discovery and Integration) [2] para registrar y localizar los servicios web. Este protocolo requiere la descripción de la aplicación o servicio concreto que se quiere invocar y si hay más de una aplicación que brinda el mismo servicio, es necesario definir en tiempo de desarrollo cual se invocará.

En este trabajo, el estudio está centrado en permitir que el motor Workflow pueda seleccionar la aplicación más adecuada a su requerimiento, de entre varias que ofrecen el mismo servicio. Se propone optimizar esta selección de servicios usando reglas de Transformación de Grafos [12].

Las reglas de transformación de grafos permiten proveer una especificación semántica precisa de un servicio web, y establecer una correspondencia entre los requerimientos del usuario y los servicios web disponibles, facilitando así su descubrimiento automático, en tiempo de ejecución.

2. Antecedentes

Existen diversos trabajos que presentan alternativas para implementar la comunicación del Workflow con las aplicaciones de clientes. En [5] los autores proponen aprovechar los beneficios de los Servicios Web, utilizándolos para especificar las funciones de la Interfaz de las Aplicaciones de Cliente, facilitando la comunicación del Workflow con las aplicaciones. Pero actualmente el descubrimiento y selección de los servicios web presenta algunos inconvenientes dados por el hecho de que el registro de los mismos no tiene en cuenta su semántica, lo cual dificulta encontrar todos los servicios que resuelven el mismo requerimiento, aunque su descripción no sea exactamente la misma. Se han propuesto distintas alternativas para solucionar estos inconvenientes. En [6] y [8] se presenta un enfoque formal para la selección automática de servicios web, basado en la correspondencia de los requerimientos del solicitante del servicio con la descripción de los servicios ofrecidos por el proveedor. Especifican las precondiciones y los efectos de las operaciones requeridas y las ofrecidas. Utilizan reglas de transformación de grafos con condiciones positivas de aplicación como una notación visual y formal para definir los contratos, y poder determinar la compatibilidad semántica y sintáctica entre los requerimientos del solicitante y las ofertas del proveedor y se concentra en chequear la compatibilidad del comportamiento. A partir de la relación semántica encontrada entre el solicitante y el proveedor, avanzan en probar la correctitud y completitud de esta relación. En [7], [9] y [11] se propone solucionar el problema de la selección dinámica y en tiempo de ejecución de los servicios web combinando ontologías y reglas de transformación de grafos. Las ontologías se usan para especificar la semántica del servicio web, representándolas de manera visual a través de diagramas de clases UML. Las reglas proveen una especificación semántica precisa, necesaria para automatizar el descubrimiento de los servicios de una manera visual e intuitiva. Presentan también una implementación usando estándares y herramientas disponibles en la actualidad.

En [10] los autores plantean incorporar la prueba automática para validar los servicios web antes de permitir su registro. Así, los clientes obligan a los servicios web ofrecidos a proveer una signatura compatible, un comportamiento adecuado, y una implementación de alta calidad.

3. Las Reglas de Transformación de Grafos

La transformación de grafos [12] es una técnica gráfica, formal, declarativa y de alto nivel muy usada para transformación de modelos, simulación de modelos, chequeos de consistencia entre modelos o vistas y optimización de diversos contextos.

La transformación de grafos es un mecanismo formal para la manipulación de grafos basado en reglas. En analogía a las gramáticas de Chomski sobre cadenas de caracteres, las reglas de grafos están formadas por una parte izquierda y una parte derecha que contienen grafos. Intuitivamente, para poder aplicar una regla a un grafo (llamado grafo anfitrión) se debe encontrar un morfismo de correspondencia entre la parte derecha de la regla y el grafo. Si dicho morfismo es encontrado, por un proceso de derivación, la regla se aplica sustituyendo la imagen de morfismo encontrado en el grafo, por la parte derecha de la regla. Dicho de otro modo, una regla de transformación de grafos, también llamada producción, (L, K, R) consiste en tres grafos L , K y R . Una producción (L, K, R) es aplicable a un grafo G si G contiene un subgrafo que es una imagen de L . Una producción (L, K, R) es aplicable a un grafo G si L es un subgrafo de G . El grafo L constituye la parte izquierda de la producción, y formula las condiciones bajo las cuales resulta aplicable la producción, o sea, es un subconjunto del conjunto de objetos (nodos y arcos) del universo. El grafo de contacto K , que generalmente es un subgrafo de L y de R , describe los subobjetos de la parte izquierda que se deben preservar en el proceso de aplicación de la producción. Por lo tanto, la diferencia entre L y K : $L-K$, contiene a todos aquellos subobjetos que se deben eliminar al aplicar la producción. Análogamente, la diferencia $R-K$ contiene a todos aquellos subobjetos que se deberán agregar al aplicar la producción. Este grafo intermedio K describe el contexto en el cual se integran los subobjetos agregados.

4. Especificación de la Interfaz de las Aplicaciones Invocadas con Servicios Web

La Interfaz de las Aplicaciones Invocadas definida por la WfMC, permite que el motor workflow pueda invocar aplicaciones externas, como por ejemplo servicios de e-mail, fax, administración de documentos, o aplicaciones de usuario. Dicha interfaz define un conjunto de APIs, que son usadas por el Sistema de Workflow para controlar los dispositivos de aplicaciones especializadas. Estas herramientas son las que finalmente se encargan de comenzar y terminar las aplicaciones, pasar la información relevante del workflow y de la aplicación “a” y “desde” la aplicación y controlar el estado a nivel de ejecución de la aplicación.

Como la Interfaz de las Aplicaciones Invocadas debe manejar requerimientos bi-direccionales, la interacción con los dispositivos de aplicaciones especializadas depende de la interfaz y arquitectura de la aplicación, lo cual restringe la selección dinámica de las aplicaciones. Para permitir el requerimiento y la actualización de datos de la aplicación y otras funcionalidades importantes en tiempo de ejecución, se propone especificar estas APIs con servicios web. En el cuadro 1 se especifican los elementos *types* e *Interface* del servicio web de la operación que permite invocar una aplicación externa.

```
<types>
  <xs:element name="InvokeApplication" type="tInvokeApplication"/>
  <xs:complexType name="tInvokeApplication">
    <xs:sequence>
      <xs:element name="appRequeriments" type="xs: tAppRequeriments"/>
      <xs:complexType name=" tAppRequeriments ">... </xs:complexType>
      <xs:element name="procInstId" type="xs: string"/>
      <xs:element name="workItemId" type="xs: string"/>
      <xs:element name="parameters" type="xs: tParameters"/>
      <xs:complexType name="tParameters"> ... </xs:complexType>
      <xs:element name="appMode" type="xs: integer"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ApplicationResult" type="tApplicationResult"/>
  <xs:complexType name="tApplicationResult">
    <xs:sequence>
      <xs:element name="appName" type="xs: string"/>
      <xs:element name="appLocation" type="xs: string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="WorkflowException" type="xs: tException"/>
  <xs:complexType name="tException">
    <xs:sequence>
      <xs:element name="error" type="xs: tError"/>
      <xs:element name="message" type="xs: String"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
</types>

<interface name = "InvokeApplicationInterface" >
  <fault name = "InvokeApplicationFault_WorkflowClientException" element = "xs: WorkflowException"/>

  <operation name="opInvokeApplication" pattern=http://www.w3.org/2004/03/wsdl/in-out>
    <input messageLabel="In" element="xs: InvokeApplication" />
    <output messageLabel="Out" element="xs: ApplicationResult" />
    <outfault ref="tns: InvokeApplicationFault_WorkflowException" messageLabel="Out"/>
  </operation>
</interface> ...
```

Cuadro 1: Descripción del servicio web que especifica la operación para invocar una aplicación

De esta manera quedan especificados los elementos *types* e *interface* para la definición de la operación InvokeApplication, de la Interfaz de Aplicaciones Invocadas 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.

5. Aplicación de las Reglas de Transformación de Grafos para la selección automática del Servicio Web *InvokeApplication*

Sobre la base de las propuestas de [6] y [7], se propone representar la interfaz del servicio web definido en el cuadro 1, con un diagrama de clases UML, y a través de la utilización de las reglas de transformación de grafos, se podrá comparar con los servicios web disponibles y accesibles desde el motor de workflow. La descripción del servicio web ofrecido y la formulación de un requerimiento se expresan como dos grafos con atributos (Figura 1). El grafo de la parte izquierda de la regla representa la precondición de un servicio, es decir, el estado del sistema antes de la ejecución del servicio web. El grafo de la parte derecha de la regla representa la poscondición, es decir, la situación después de satisfacerse la ejecución del servicio web.

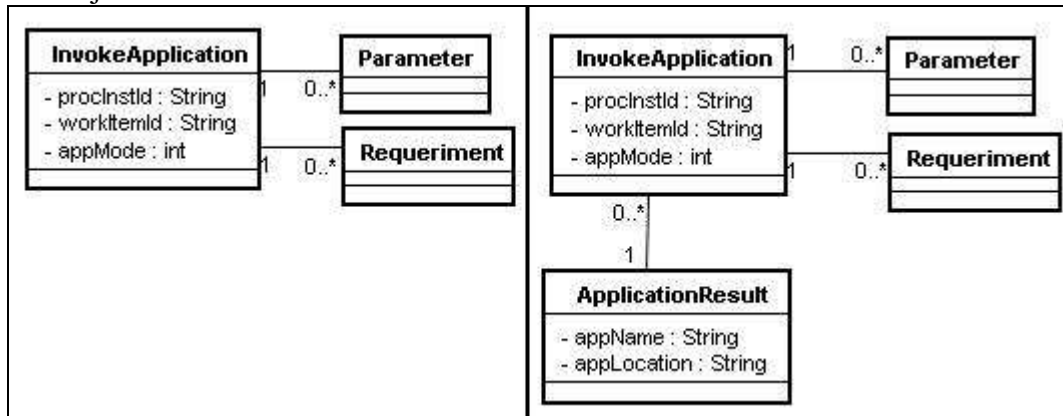


Figura 1: Grafos de la pre y poscondición del servicio solicitado

En los requerimientos modelados en la pre y la poscondición es posible almacenar la información relevante para el motor workflow sobre las características de la aplicación buscada por el usuario, por ejemplo, la categoría de la aplicación (cliente de correo, editor de texto, planilla de cálculo, etc.), el sistema operativo del usuario, su ubicación geográfica, etc.

Los parámetros almacenan información relevante para la aplicación, por ejemplo, el nombre de un archivo, el identificador de un registro, etc.

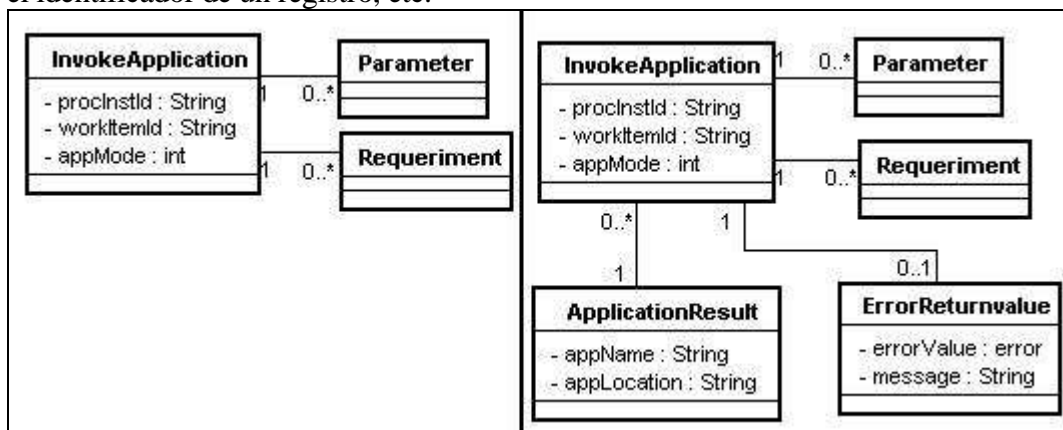


Figura 2: Grafos de la pre y poscondición del servicio del proveedor

Para decidir automáticamente si un servicio del motor workflow satisface la demanda del solicitante, es necesario comparar los grafos generados por el solicitante y el proveedor (Figura 2) (motor workflow). Se formalizará usando relaciones de subgrafos. Si la precondición del proveedor es un subgrafo de la precondición del solicitante, entonces el proveedor provee toda la información necesaria para ejecutar el servicio web. Si la poscondición del solicitante es un subgrafo de la poscondición del proveedor, entonces el servicio genera todos los efectos esperados por el solicitante, además de algunos efectos adicionales, como en este caso los errores posibles.

6. Conclusiones

Los Sistemas de Administración de Workflow se desarrollan en las empresas con el objetivo de automatizar los procesos de trabajo. La WfMC ha propuesto un Modelo de Referencia de Workflow para permitir la interoperabilidad entre estos sistemas. Dicho modelo presenta una Interfaz de Aplicaciones Invocadas para que el motor workflow pueda invocar aplicaciones externas, como por ejemplo servicios de e-mail, fax, administración de documentos, o aplicaciones de usuario.

En este trabajo se plantea la optimización de esta interfaz utilizando servicios web. A partir de dicha optimización se analizan los inconvenientes que existen actualmente al momento de elegir un servicio web, dados por las características de funcionamiento del protocolo UDDI (Universal Description, Discovery and Integration), y se estudian las diferentes líneas de investigación conducentes a solucionar estos inconvenientes. Los trabajos mencionados en la sección de antecedentes proponen un enfoque formal para la selección automática de servicios web, basado en la correspondencia de los requerimientos del solicitante del servicio con la descripción de los servicios ofrecidos por el proveedor, utilizando reglas de transformación de grafos.

Para permitir que el motor Workflow pueda invocar aplicaciones externas seleccionando la más adecuada en tiempo de ejecución, se plantea modelar con un Diagrama de Clases UML la interfaz del servicio web *InvokeApplication*, de manera tal de formalizarla como un grafo con atributos, y poder realizar una correspondencia con los servicios web disponibles en la red que cumplen con el morfismo establecido. La lista de servicios obtenidos debe ser analizada por el motor Workflow para seleccionar el más adecuado según los requerimientos del usuario. Con esta especificación de las interfaces de Aplicaciones de Cliente y Aplicaciones Invocadas con servicios web y la optimización de la búsqueda y selección de los servicios en la red, se logra que el Workflow se comporte internamente de forma distribuida. Así, el workflow no necesita conocer dónde están las aplicaciones para invocarlas. Simplemente, requiere servicios y hay aplicaciones que le proveen dichos servicios. Además, se facilita la incorporación de servicios nuevos, y la invocación no está limitada a una aplicación específica, definida en tiempo de desarrollo.

7. Referencias Bibliográficas

- [1] World Wide Web Consortium. Web Service Architecture. <http://www.w3.org/TR/ws-arch/>. Último acceso 03/2008.
- [2] OASIS. UDDI Version 3.0.2. http://uddi.org/pubs/uddi_v3.htm. Último acceso Marzo de 2008.
- [3] Workflow Management Coalition. The Workflow Reference Model. WfMC-TC00-1003. <http://www.wfmc.org/standards/referencemodel.htm>. Último acceso Marzo de 2008.
- [4] Workflow Management Coalition. Programming Interface 2&3 Specification. WfMC-TC-1009. V2.0. <http://www.wfmc.org/standards/publicdocuments.htm>. Último acceso Marzo de 2008.
- [5] Debnath N., Martellotto P., Daniele M., Riesco D., Montejano G., Using Web Services To Optimize Communication Among The Workflow Engine And Its Client Application Interface. Second International Conference on Information Systems, Technology and Management. ICISTM 2008. Institute of Management Technology. Ghaziabad. Dubai. Marzo 2008.
- [6] Heckel R., Cherchago A., Application of Graph Transformation for Automating Web Service Discovery, Proc. Dagstuhl Seminar 04101 Language Engineering for Model-Driven Software Development, Dagstuhl, Germany, 2004.
- [7] Hausmann J.H., Heckel R., Lohmann M., Towards Automatic Selection of Web Services Using Graph Transformation Rules. Berliner XML Tage. XML-Clearinghouse, 2003 – 286-291.
- [8] Cherchago A., Heckel R., Specification Matching of Web Services Using Conditional Graph Transformation Rules, In G. Engels, H. Ehrig, F. Parisi-Presicce, and G. Rozenberg (Editors): Proc. 2nd International Conference on Graph Transformation (ICGT 04), Roma, Italy, Volume 3256 of Lecture Notes in Computer Science. Springer-Verlag, 2004
- [9] Hausmann J.H., Heckel R., Lohmann M., Model-based Discovery of Web Services, Proceedings of the IEEE International Conference on Web Services, 2004.
- [10] Heckel R., Mariani L., Automatic Conformance Testing of Web Services. Fundamental approaches to software engineering. International conference No8, Edinburgh , ROYAUME-UNI (04/04/2005) 1973, vol. 3442, pp. 34-48, [Note(s) : XIII, 371 p.,] (24 ref.) ISBN 3-540-25420-X.
- [11] Hausmann J.H., Heckel R., Lohmann M., Model-based development of Web services descriptions enabling a precise matching concept. International Journal of Web Service Research. Volume. 2. Issue: 2. p. 67 – 84. 2005.
- [12] Baresi L., Heckel R., Tutorial Introduction to Graph Transformation: A Software Engineering Perspective. First International Conference on Graph Transformation. Spain. v. 2505, pp. 402-429. (ICGT 2002).