

Invocación de Aplicaciones desde el Workflow con Servicios Web considerando sus requerimientos no funcionales

Paola Martellotto¹, Marcela Daniele¹, Daniel Riesco²

¹Universidad Nacional de Río Cuarto - Facultad de Ciencias Exactas, Fco-Qcas y Naturales
Dpto. de Computación – Río Cuarto, Córdoba, Argentina
{paola, marcela}@dc.exa.unrc.edu.ar

²Universidad Nacional de San Luis - Facultad de Ciencias Físico Matemáticas y Naturales
Departamento de Informática – San Luis, Argentina
driesco@unsl.edu.ar

Resumen. El Modelo de Referencia de Workflow estandariza el desarrollo de aplicaciones workflow, y logra su interoperabilidad. Aplicando la tecnología de Servicios Web es posible optimizar la invocación de aplicaciones desde el motor Workflow, logrando que el mismo no necesite conocer la ubicación de la aplicación a invocar, y que cualquier aplicación pueda cambiar su ubicación sin que implique un cambio en su invocación. Pero UDDI, presenta inconvenientes para establecer una correspondencia entre los requerimientos del solicitante y la especificación de los servicios que poseen el mismo comportamiento semántico. Además, para la selección de un servicio, no tiene en cuenta sus atributos de calidad (QoS). Usando Reglas de Transformación de Grafos se logra obtener una especificación precisa del servicio web, y en este trabajo se propone mejorar la selección del servicio web más adecuado, incorporando la especificación de sus atributos de calidad (QoS).

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

1 Introducción

Los sistemas de gestión de workflow son sistemas de software que definen, crean y manejan la ejecución de flujos de trabajo. La WfMC (Workflow Management Coalition) [1] estandariza la automatización de los flujos de trabajo, y define un marco genérico para la construcción de sistemas de gestión de workflow. Este modelo de referencia define cinco interfaces, que permiten a las aplicaciones del workflow la comunicación a distintos niveles, y especifica un conjunto de WAPIs (Workflow Application Programming Interfaces) que definen las funciones de dichas interfaces. En particular, para la invocación de las aplicaciones externas define la Interfaz de las Aplicaciones Invocadas.

La tecnología implícita en los servicios web permite dar una solución al inconveniente que presentan las WAPIs, al obligar a conocer la información acerca de la aplicación y su invocación en tiempo de desarrollo.

Actualmente, los servicios web son registrados a través del protocolo UDDI [5] para ser localizados y usados por las aplicaciones. Este protocolo requiere la descripción sintáctica de la aplicación concreta 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. Para salvar este inconveniente, en [4] se propuso la utilización de Reglas de Transformación de Grafos para que el motor workflow pueda seleccionar la aplicación más adecuada, entre varias que ofrecen el mismo servicio, en tiempo de ejecución.

Sin embargo, la descripción sintáctica de un servicio web en UDDI focaliza en sus aspectos funcionales, sin contener ninguna descripción de los atributos de calidad (QoS). Con el incremento del número de servicios web que proveen similares funcionalidades, es fundamental encontrar el mejor servicio web que satisface las necesidades del solicitante, incluyendo tanto los requerimientos funcionales como los no funcionales.

Este trabajo propone, optimizar la selección del servicio web más adecuado que satisfaga tanto la funcionalidad como los atributos de calidad requeridos por el solicitante, siempre en el marco de optimizar la invocación de aplicaciones desde un sistema de gestión de Workflow, usando Reglas de Transformación de Grafos y describiendo tanto la funcionalidad como los atributos de calidad del servicio web.

La sección 2 describe el estado del arte, la sección 3 se presenta la especificación funcional de la interfaz de las aplicaciones invocadas, y se describen las características de calidad deseadas para los servicios web, la sección 4 muestra la aplicación de reglas de transformación de grafos para la especificación de servicios web y por último, se presentan las conclusiones de esta propuesta.

2 Estado del Arte

En propuestas anteriores de los autores de este trabajo [3] [4], se presenta una optimización de la comunicación del motor del Workflow con las aplicaciones de clientes y externas, y una propuesta relacionada al descubrimiento y selección de los servicios web que tiene en cuenta su semántica, y permite encontrar todos los servicios que resuelven el mismo requerimiento, aún con diferentes descripciones.

En este trabajo, se estudiaron propuestas relacionadas al descubrimiento de servicios web considerando sus atributos de calidad. En [6] se propone incorporar un Certificador QoS para verificar los atributos de calidad ofrecidos por los proveedores del servicio, donde el registro en UDDI se realiza después que el proveedor verifica los requerimientos QoS. En [7], se presenta un framework basado en los componentes básicos de un servicio web con la capacidad adicional de almacenar características QoS usando la estructura de datos tModel. En [8] se presenta un framework de descubrimiento que considera los atributos de calidad QoS, y la información del contexto de ejecución del servicio. En [9], proponen extender la ontología OWL-S con un modelo flexible QoS, basado en un conjunto de restricciones globales del

usuario. En [10], extienden WSDL para incorporar la descripción de las características de calidad QoS. En [11], se presenta un framework para mejorar el descubrimiento de los servicios web, que define tres participantes y la interacción entre ellos: *Personal Agent*, *Matching Agent* y los *Service Mediators*.

La mayoría de las propuestas mencionadas reconocen la importancia de la evaluación de los atributos de calidad de un servicio web para lograr una selección más adecuada.

3 Especificación de la Interfaz de Aplicaciones Invocadas

WSDL (Web Service Description Language) [13] [14] es el lenguaje usado para describir los aspectos funcionales de un servicio web, pero no contiene ninguna descripción de las características de los servicios web (QoS). La descripción no funcional de un servicio se concentra en las características de calidad del mismo, tales como disponibilidad, tiempo de respuesta, fiabilidad, etc. Las características de calidad de un servicio se refieren a la habilidad del mismo de responder a las invocaciones y llevarlas a cabo en consonancia con las expectativas del proveedor y de los clientes [5].

Las características de calidad consideradas en este trabajo son *performance* (velocidad para completar un requerimiento, medida en términos de tiempo de respuesta, throughput y latencia), *disponibilidad* (probabilidad de que el servicio web este disponible), *fiabilidad* (habilidad de un servicio para funcionar correcta y consistentemente), *precisión* (radio de error producido por un servicio), *robustez* (si el servicio funciona correctamente en presencia de entradas inválidas o incompletas) y *reputación* (reputación otorgada al servicio por el usuario final).

A la especificación funcional de la operación que permite invocar una aplicación externa (InvokeApp), definida en [4], se le incorpora la descripción de los requerimientos no funcionales como muestra la Figura 1. Los mismos son definidos como un elemento nuevo en la sección types del documento WSDL.

```
<xs:element name="QoSReq" type="xs:tQoSReq"/>
<xs:complexType name="tQoSReq">
  <xs:sequence>
    <xs:element name="QoSname" type="xs:string"/>
    <xs:element name="QoSunitType" type="xs:tType"/>
    <xs:element name="QoSminValue" type="xs:float"/>
    <xs:element name="QoSmaxValue" type="xs:float"/>
  </xs:sequence>
</xs:complexType>
```

Figura 1. Descripción del elemento QoSReq de la operación InvokeApp

4 Selección de servicios con Reglas de Transformación de Grafos

La especificación de *invokeApp* contiene el tipo de aplicación, los parámetros y los requerimientos adicionales. A esta especificación, son incorporados los

requerimientos no funcionales del solicitante. El solicitante detalla los atributos de calidad requeridos en el documento WSDL de *invokeApp*. Los atributos de calidad de los servicios ofrecidos por los proveedores son monitoreados por el motor workflow, y registrados internamente para realizar la correspondencia. Como resultado, el motor workflow obtiene una lista de los servicios web candidates, y los analiza, teniendo en cuenta la historia de las ejecuciones previas y el registro de características de calidad del motor workflow. La historia de las ejecuciones es almacenada por el propio motor workflow, a medida que se van invocando las aplicaciones. Además, evalúa las características de calidad de las aplicaciones y las registra en el historial, y posee también un registro de características de calidad del propio motor workflow, que son las características de calidad mínimas que acepta al invocar una aplicación. Su objetivo es permitir el análisis de las características que no fueron explícitamente requeridas por el solicitante, en base a los requerimientos mínimos que el mismo motor workflow tiene preestablecidos. Finalmente, el motor elige uno de los servicios web candidatos, lo invoca y lo registra en la historia de las ejecuciones previas.

En [4] se presentó la especificación del servicio web *InvokeApp*, para invocar una aplicación externa independizándose de la ubicación exacta de la misma. Interesa ahora analizar la compatibilidad de los servicios web provistos en la red y los requeridos de un usuario del workflow, incorporando los atributos de calidad.

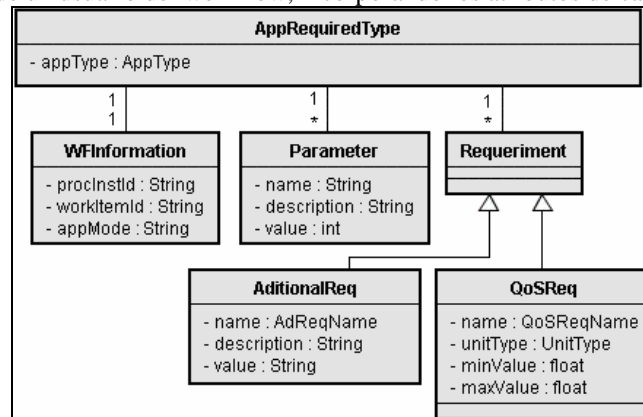


Figura 2. Grafos de la pre y poscondición del servicio solicitado

En la Fig. 2, el grafo representa la precondición del servicio solicitado, es decir, el estado del sistema antes de la ejecución del servicio. Este grafo tiene una relación directa con la especificación WSDL del elemento *InvokeApp*, en el elemento *types* del servicio. También aparece la clase **QoSReq**, que representa los atributos de calidad requeridos, que se agregaron como un nuevo elemento en la especificación de *InvokeApp*. Análogamente, se pueden especificar la poscondición del servicio solicitado, así como también las pre y poscondiciones de los servicios web ofrecidos por los proveedores. Estos grafos forman la parte izquierda y la parte derecha de las reglas de transformación de grafos.

Para decidir automáticamente si un servicio web ofrecido por un proveedor satisface la demanda del solicitante, es necesario comparar los grafos generados por ambos. Se formaliza usando relaciones de subgrafos. Si la precondición del proveedor es un subgrafo de la precondición del solicitante, entonces el proveedor brinda toda la

información necesaria para ejecutar el servicio. 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. En la Fig. 3 se puede observar que la precondition del proveedor contiene todas las clases de la precondition del solicitante, por lo tanto, es un subgrafo, y lo mismo sucede con las poscondiciones del solicitante y el proveedor, como se muestra en la Fig 4.

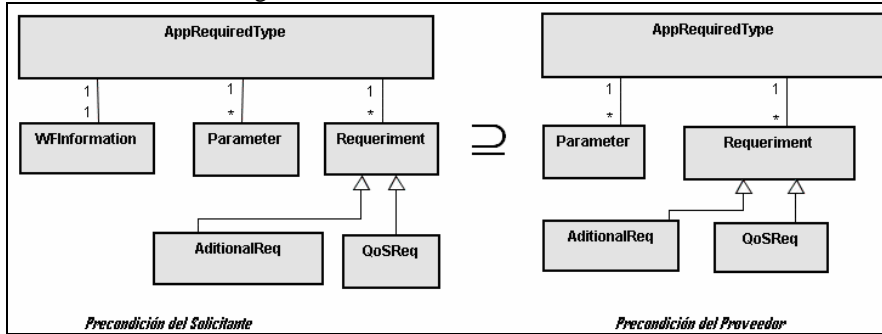


Figura 3. Grafos de las precondiciones del solicitante y el proveedor

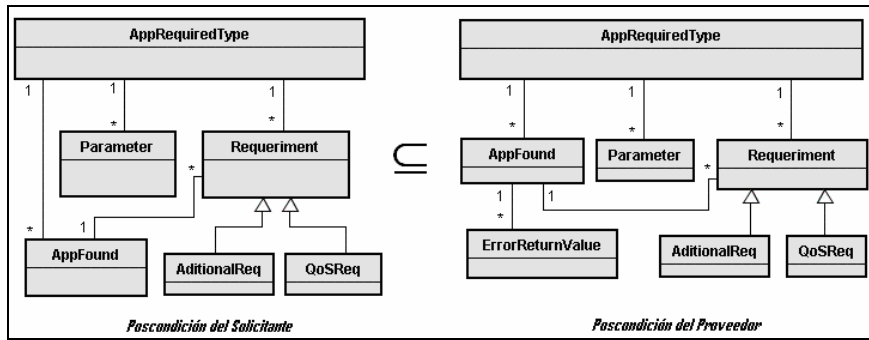


Figura 4. Grafos de las poscondiciones del solicitante y el proveedor

Si se cumple con el morfismo establecido, entonces es posible asegurar que el proveedor cumple los requerimientos del solicitante, tanto los funcionales como los no funcionales. Así, el solicitante puede ajustar sus requerimientos incorporando los factores de calidad que reflejan sus expectativas, y por otra parte, el motor workflow puede utilizar el mismo mecanismo de correspondencia para analizar tanto los requerimientos funcionales como los no funcionales.

5 Conclusiones

La formalización de la especificación como un grafo con atributos permite realizar una correspondencia con los servicios web disponibles en la red que cumplen con el morfismo establecido, logrando que el workflow se comporte internamente de forma

distribuida. Además, facilita la incorporación de servicios nuevos, y la invocación no está limitada a una aplicación específica, definida en tiempo de desarrollo.

Asimismo, con la incorporación de los atributos de calidad (QoS) a la descripción de los servicios web se hace posible evaluar la calidad de los mismos, filtrar la lista de todos los servicios descubiertos y obtener los más adecuados.

El entorno crecientemente competitivo en que se mueven las empresas hace que cada vez crezca más el apremio sobre los procesos de negocio operativos, entre los que se encuadra la provisión de servicios que permitan reducir tiempos, acelerar los procesos, mejorar y personalizar el trato con el cliente y, además, aumentar la eficiencia y reducir costes.

Esta propuesta no requiere modificar el modelo de servicios web actual y permite al motor workflow valerse de la información provista por el proveedor para mejorar la selección de servicios web.

Referencias Bibliográficas

1. Workflow Management Coalition. The Workflow Reference Model. WfMC-TC00-1003. <http://www.wfmc.org/standards/referencemodel.htm>. Último acceso Mayo 2008.
2. Workflow Management Coalition. Programming Interface 2&3 Specification. WfMC-TC-1009. V2.0. <http://www.wfmc.org/standards/publicdocuments.htm>. Ult. acceso Mayo/08.
3. 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. Ghaziabad. Dubai. March 2008.
4. Daniele Marcela, Martellotto Paola, Riesco Daniel. "Implementación de la comunicación entre el Workflow y las Aplicaciones Externas con Servicios Web y Reglas de Transformación de Grafos". Workshop de Ingeniería de Software y Base de Datos, CACIC 2008. Univ. Nacional de Chilecito. Chilecito, La Rioja, Argentina. Octubre/08.
5. Papazoglou M., Web Services: Principles and Technology. Ed. Pearson Education Limited, Prentice Hall, First Edition. 2008.
6. Ran, S., A Model for Web Services Discovery with QoS, ACM SIGecom Exchanges 4(1): pp. 1-10, 2003.
7. Al-Hunaity, M., El-Sheikh, A., Dudin, B., A New Web Service Discovery Model Based On QoS. Arab Academy for Banking and Financial Sciences, Jordan. ICTTA08. April/08.
8. Braun, I., Strunk, A., Stoyanova, G., Buder, B., ConQo – A Context- And QoS-Aware Service Discovery. Freiburg. IADIS. In proceedings of WWW/Internet. October 2008.
9. Zhang, W., Huang, Ch., Yang, Y., Jia, X.: QoS-driven Service Selection Optimization Model and Algorithms for Composite Web Services, IEEE proceedings of COMPSAC'07.
10. D'Ambrogio, A.: A Model-driven WSDL Extension for Describing the QoS of Web Services. ICWS. P 789-796. 2006.
11. Kokash N.: Web Service Discovery with Implicit QoS Filtering. In proceedings of the IBM PhD Student Symposium, colocated with the "International Conference on Service-Oriented Computing (ICSOC)". P 61-66. Amsterdam, December 2005.
12. 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 2008.
13. 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 2008.