

Automatización de la etapa de análisis para la aplicación de la técnica de prueba metamórfica a composiciones de servicios WS-BPEL

M Carmen de Castro-Cabrera, Kevin J Valle-Gómez, Inmaculada Medina-Bulo

Departamento de Ingeniería Informática, Universidad de Cádiz
Avda. Universidad de Cádiz, 10, 11519, Puerto Real, España
kevin.vallegomez@alum.uca.es,
{maricarmen.decastro,inmaculada.medina}@uca.es

Resumen La importancia de los servicios web en el mundo actual está en auge, y el impacto social que están teniendo gracias a las nuevas tecnologías desarrolladas, hace necesario el diseño de mejores técnicas de prueba para el software específico de composición de estos servicios web. Una de las técnicas propuestas para las composiciones WS-BPEL es la técnica de prueba metamórfica. En trabajos anteriores se ha presentado una arquitectura para su aplicación y se han aportado algunas ideas para la automatización de la etapa inicial de la misma, que se corresponde con el análisis y obtención de propiedades, pero no se llegaron a desarrollar ni implementar. En este trabajo se presenta el diseño de una solución para automatizar ciertos aspectos de la etapa de análisis y obtención de propiedades, la cual ha sido probada en diferentes casos de prueba obteniéndose buenos resultados.

Keywords: pruebas metamórficas, relaciones metamórficas, WS-BPEL, composiciones de servicios web, solución, análisis

1. Introducción

El lenguaje *WS-BPEL 2.0* [5] permite, a partir de servicios web (WS) ya existentes, crear procesos de negocio complejos. A pesar de la rápida y continua mejora de este tipo de servicios, las técnicas de prueba empleadas para probarlos no han avanzado al mismo ritmo. Por ello, es de máximo interés investigar en esta línea e implementar nuevos métodos de pruebas en este campo, así como, mejorar los existentes.

La *técnica de prueba metamórfica* (PM) [3], que ha sido aplicada con éxito en lenguajes de programación tradicionales [7], permite verificar el software obteniendo nuevos casos de prueba a partir de otros iniciales mediante la aplicación de las denominadas relaciones metamórficas (RM). Las RM son propiedades esperadas o existentes en el software, definidas sobre un conjunto de entradas y salidas.

El *análisis y obtención de propiedades* de las composiciones WS-BPEL 2.0 es uno de los aspectos más críticos en la arquitectura propuesta en [2]. Debido a la

complejidad de estas composiciones y las reglas de negocio existentes, más allá de los procesos puramente matemáticos, esta tarea se realiza normalmente de manera manual, creando una clara necesidad de automatización con el objetivo de agilizar el proceso y los costes.

En este trabajo se describe un proceso automatizado para la fase de análisis para su posterior integración en la arquitectura propuesta en [2]. La información obtenida en el proceso de análisis será utilizada para la implementación de RMs. Este trabajo se estructura de la siguiente forma: En la sección 2 se expondrán los fundamentos necesarios para abordar el resto del trabajo. En la sección 3 se describe proceso seguido para el análisis de las composiciones WS-BPEL, mientras que en la sección 4 se muestra un caso de estudio. Finalmente, se detallan las conclusiones y el trabajo futuro en la sección 5.

2. Fundamentos

A continuación se describen los fundamentos, en este orden: el lenguaje WS-BPEL, la técnica prueba metamórfica y las herramientas utilizadas.

WS-BPEL [5] está basado en XML y permite especificar el comportamiento de un proceso de negocio basado en sus interacciones con servicios web (SW). Se utiliza para generar nuevos procesos de negocios a partir de servicios ya existentes. Los bloques principales que forman las composiciones WS-BPEL son las *actividades*.

El concepto de PM está muy relacionado con las RM. Una definición para RM según [1] sería *propiedades existentes sobre el conjunto de entradas y sus correspondientes resultados para múltiples evaluaciones de una función*; es decir, siempre que la entrada sea correcta, con las RM esperamos una salida que cumpla determinadas propiedades necesarias. Una RM debe proveer una forma de generar nuevos casos de prueba a partir de otros ya existentes.

El desarrollo de la arquitectura propuesta en [2] requiere la consecución de varias etapas, cuya interacción permite la aplicación de PM. Para la etapa de análisis y obtención de propiedades, es necesario un analizador de composiciones capaz de obtener de forma automática información relevante que facilite la elección e implementación de las propiedades a aplicar a cada composición.

Para la implementación se hace uso de algunos módulos de la herramienta *test-generator-autoseed* [4], y de la utilidad *XMLStarlet*, que permite modificar y validar XML. Su uso será descrito en los siguientes apartados.

3. Análisis y extracción de información

Para realizar el proceso de análisis de composiciones de la arquitectura [2] con la mínima intervención humana posible, se propone la implementación de una solución compuesta por los siguientes pasos:

1. Desarrollo de una aplicación capaz de utilizar mediante herencia funciones predefinidas en la herramienta **test-generator-autoseed** [4] para obtener

- información detallada de los elementos presentes en la composición WS-BPEL. Así mismo, se dispondrá de otro módulo de análisis que mediante el uso de **XMLStarlet** y funciones XPath obtendrá información detallada de los elementos presentes en los casos de prueba.
2. Puesta en común de los resultados obtenidos en cada parte, con el objetivo de determinar qué elementos serán más propensos a conducir a la obtención de RM.
 3. Presentación de un informe en dos formatos, uno con formato técnico y otro escrito en lenguaje natural, destinados a la máquina y el analista, respectivamente.

La herramienta **test-generator-autoseed** provee de un conjunto de funciones *getters* y *setters* que serán reutilizados para la extracción de información de los elementos presentes en las composiciones WS-BPEL. Así mismo, mediante el uso de expresiones XPath junto a **XMLStarlet**, podemos explorar el contenido de sus casos de prueba.

4. Aplicación a un caso de estudio

Para mostrar el resultado, se utilizará la composición *LoanApproval* [6]. Esta composición consiste en un proceso de aprobación de préstamos a partir de una cantidad monetaria, cuyo límite fijado en 10.000 unidades monetarias, determinará la acción de un aprobador de préstamos y/o un asesor financiero, como puede observarse en la Figura 1.

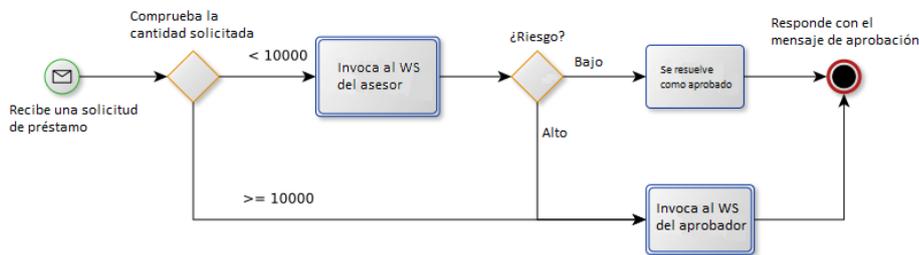


Figura 1. Lógica de la composición LoanApproval

La solución propuesta generará un informe disponible en dos formatos:

1. Una serie de ficheros CSV, cuyo formato genérico permitirá su gestión automática para la obtención de RM y otros usos que en su momento se crean convenientes. Un ejemplo del fichero correspondiente a las actividades IF de la composición *LoanApproval* puede verse en la Figura 2.
2. Un informe en lenguaje natural, que se utilizará de manera auxiliar para facilitar la tarea del análisis manual en el caso de que este fuese necesario.

type	name	condition	else
IF	If1	(number(string(\$processInput.input/ns0:amount)) <= 10000)	yes
IF	IfLowRisk	(\$risk.level = 'low')	yes

Figura 2. Fragmento CSV

Se han obtenido manualmente y probado con buenos resultados algunas RM de la composición *LoanApproval*. Se detalla a continuación una de ellas:

$$\mathbf{RM1:} \text{ req_amount}_1 \cdot 10 > 10,000 \wedge \text{req_amount}_2 = \text{req_amount}_1 \cdot 10 \wedge \text{ap_reply}_2 = \text{not}(\text{ap_reply}_1) \wedge \text{as_reply}_2 = \text{as_reply}_1 \implies \text{accepted}_2 = \text{not}(\text{accepted}_1) \wedge \text{ap_reply}_2 = \text{accepted}_2$$

Los elementos presentes en ella, se obtienen de la información contenida en la composición WS-BPEL y sus casos de prueba. Concretamente, podemos observar que la constante 10.000 es la cantidad límite que aparece en la lógica del programa, junto al operador relacional $>$, que determina qué rama debe seguirse en la ejecución de la composición, implica que debe haber un elemento condicional IF. Estos elementos aparecen en el ejemplo mostrado en la Sección 4, lo que implica que la información obtenida se considera de utilidad para la obtención de RM.

5. Conclusiones y trabajo futuro

Los procesos de negocio que se basan en WS-BPEL aumentan cada año, creando la necesidad de mejores técnicas de prueba para este tipo de software, obligando a adaptar técnicas tradicionales para verificar estas composiciones.

En este trabajo se ha presentado una aplicación capaz de automatizar parte de la etapa de análisis de composiciones de la arquitectura propuesta en [2], para su integración en la misma.

Como trabajo futuro, está completar la implementación de esta aplicación: Para ello, se está trabajando, en la obtención otros tipos de formatos de ficheros enfocados a la automatización del tratamiento de los resultados, haciendo la solución más fácilmente legible por un programa. Por otro lado, se trabajará en la mejora de la salida en lenguaje natural.

6. Agradecimientos

Este trabajo ha sido financiado por el Programa Nacional para la Investigación, Desarrollo e Innovación del MICINN, proyecto DArDOS (TIN2015-65845-C3-3-R).

Referencias

1. Andrews, J.H., Briand, L.C., Labiche, Y.: Is mutation an appropriate tool for testing experiments? In: Proceedings of the 27th International Conference on Software Engineering (ICSE 2005). pp. 402–411. ACM Press (2005)
2. Castro-Cabrera, M.d.C., Camacho-Magriñán, A., Medina-Bulo, I., Palomo-Duarte, M.: Una arquitectura basada en pruebas metamórficas para composiciones de servicios ws-bpel. In: Actas de las VII Jornadas de Ciencia e Ingeniería de Servicios. pp. 9–22. Servizo de publicacións da Universidade da Coruña, A Coruña, España (Sep 2011)
3. Chen, T.Y.: Metamorphic testing: A new approach for generating next test cases. HKUSTCS98-01 (1998)
4. Liñeiro Barea, V.: Herramienta para la generación automática de casos de prueba mediante siembra automática para WS-BPEL 2.0 (feb 2014), <http://rodin.uca.es:80/xmlui/handle/10498/16222>
5. OASIS: Web Services Business Process Execution Language 2.0. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> (2007), Organization for the Advancement of Structured Information Standards
6. UCASE Research Group: LoanApprovalDoc - WS-BPEL Composition Repository - Redmine ((accessed Jun 30, 2016)), <https://neptuno.uca.es/redmine/projects/wsbpel-comp-repo/wiki/LoanApprovalDoc>
7. Zhou, Z.Q., Huang, D.H., Tse, T.H., Yang, Z., Huang, H., Chen, T.Y.: Metamorphic testing and its applications. In: Proceedings of the 8th International Symposium on Future Software Technology (ISFST 2004). Software Engineers Association (2004)