

Métodos y Técnicas para Cálculo del Grado de Entendimiento de Servicios Web Basados en WSDL

Hernán Bernardis⁽¹⁾, Edgardo Bernardis⁽¹⁾, Mario M. Berón⁽¹⁾, Daniel E. Riesco⁽¹⁾, Pedro Rangel Henriques⁽²⁾, Maria Joao V. Pereira⁽³⁾

⁽¹⁾Departamento de Informática / Facultad Ciencias Físico Matemáticas y Naturales/ Universidad Nacional de San Luis
Ejército de los Andes 950 – San Luis – Argentina
{hbernardis, ebernardis, mberon, [driesco](mailto:driesco@unsl.edu.ar)}@unsl.edu.ar

⁽²⁾ Departamento de Informática/Universidade do Minho
Braga – Portugal
pedrorangelhenriques@gmail.com

⁽³⁾ Departamento de Informática e Comunicações/ Instituto Politécnico de Bragança
Bragança - Portugal
mjoao@ipb.pt

Resumen

La transformación del desarrollo de software es cada vez mayor migrándose la mayoría de los sistemas a arquitecturas orientadas a servicios en la nube. Esta relativamente nueva modalidad de desarrollo de sistemas de software se hace difícil comprender los sistemas debido a que la complejidad de un sistema ahora radica en la complejidad de cada uno de sus módulos sumada a la complejidad de los servicios web con los que el mismo interactúa.

En el caso particular de los servicios web que utilizan un WSDL (Web Services Description Language) [1] para ser descriptos, la complejidad de entendimiento para un ingeniero de software se incrementa debido a que los mismos son diseñados para ser interpretados por agentes de software y no por personas. Esta alta formalidad permite incluso que estos servicios web posean herramientas de generación automática. Esta automatización resulta extremadamente útil y eficiente al reducir los tiempos de desarrollo pero carece de semántica que permita comprenderlo luego de creado.

En este artículo se describe una línea de investigación centrada el cálculo de múltiples métricas de los Servicios Web mediante el análisis de sus especificaciones WSDL. Estas métricas sirven como base para el cálculo del grado global de entendimiento del WSDL.

Palabras clave: Web Services, métricas, comprensión, WSDL.

CONTEXTO

La línea de investigación descrita en este artículo se desarrolla en el Laboratorio de Calidad e Ingeniería de Software (LaCIS) de la Universidad Nacional de San Luis; y se encuentra enmarcada dentro del proyecto (P-031516): “*Ingeniería del Software: Conceptos, Prácticas y Herramientas para el Desarrollo de Software de Calidad*”, perteneciente a la universidad antes mencionada. Dicho proyecto es reconocido por el programa de incentivos, y es la continuación de diferentes proyectos de investigación de gran éxito a nivel nacional e internacional. También dentro del proyecto bilateral con la Universidade do Minho

(Portugal): “*Fortalecimiento de la Seguridad de los Sistemas de Software mediante el uso de Métodos, Técnicas y Herramientas de Ingeniería Reversa*”, aprobado por el Ministerio de Ciencia Tecnología e Innovación Productiva (Mincyt) con código PO/16/93.

1. INTRODUCCIÓN

El incremento en el uso de los servicios web dentro del desarrollo de sistemas es cada vez mayor. Los sistemas migran hacia arquitecturas orientadas a servicios [2,3] en lugar de seguir con la arquitectura centralizada tradicional. La idea detrás de esto es similar a la modularización tradicional en el desarrollo de sistemas. En lugar de aislar código en clases, paquetes o métodos, para ser reutilizados mediante su llamada, en múltiples partes del sistema, se lo hace mediante la creación de servicios web. Esto posee dos grandes ventajas: i) su alcance es mucho más global debido a que los servicios web se publican en internet y son accesibles desde cualquier lado y ii) su construcción es independiente del lenguaje utilizado por quien lo invoque. Estas características permiten también que se puedan utilizar servicios web de terceros, es decir, sin necesidad que los mismos sean desarrollados por quienes lo utilizan.

La gran migración hacia arquitecturas orientadas a servicios ha generado múltiples herramientas y técnicas para automatizar la creación de las descripciones de los mismos. En el caso de los servicios web basados en WSDL, la formalidad de los WSDLs permite automatizar su construcción. Si un WSDL es creado con la formalidad necesaria para ser interpretado por un agente de software automáticamente, entonces suena lógico que pueda ser generado por otro software de manera automática.

Esto es altamente provechoso por el ahorro de tiempo que produce construir un sistema que realiza una determinada funcionalidad y, de manera automática, crear su interfaz y descripción para su uso como servicio web.

Disminución de tiempos, evitarse la construcción de formalidades innecesarias, facilidad de construcción son algunas de las ventajas que proveen estas herramientas de generación automática. Pero, como todo, también hay desventajas en este proceso y las mismas aparecen a la hora de analizar los WSDLs generados.

Si se supone la situación particular de un sistema de una multinacional con múltiples invocaciones a servicios web que se debe migrar hacia una tecnología nueva. Este proceso lleva un planeamiento de migración en donde se deben especificar por cada módulo o funcionalidad los periodos en tiempo y los recursos necesarios para la migración. Para los sistemas tradicionales existen múltiples formas de estimar la complejidad [4,5] pero para los servicios web y los módulos afectados, que tan difícil será?, ¿Cuánto tiempo llevará?. Otra situación que puede ocurrir es el caso donde un banco posee múltiples servicios web en su sistema con arquitectura orientada a servicios y desea determinar qué tan vulnerables son a posibles ataques informáticos. ¿Cómo puede analizar las vulnerabilidades del mismo? ¿Cómo determinar que tan comprensibles son a personas externas que deseen encontrar vulnerabilidades en sus descripciones para atacar? Ambos escenarios desnudan la necesidad de técnicas de comprensión de servicios web, pilares funcionales de todo sistema.

Toda comprensión siempre nacerá con una extracción de información. Si no obtengo información de algo, ¿Cómo puedo llegar a comprenderlo? Es bien sabido que la comprensión dentro del proceso de desarrollo de software es la parte más costosa en tiempo y recursos, justamente por la dificultad que implica extraer la información adecuada y necesaria para realizarla [4,5]. Poca información no alcanzará, mucha información entorpecerá. Entonces saber qué información extraer es casi tan importante como la información en sí. En el caso particular de los

servicios web nace otro inconveniente: de dónde sacar la información de una funcionalidad encapsulada en algún lugar de internet que sólo nos dice su forma de usarla [6,7]. Es aquí donde la atención se desplaza, obviamente, a la mayor fuente de información que se posee: la descripción WSDL del servicio web. Orientada hacia los agentes de software que la interpretarán y utilizarán, esta descripción no siempre es simple de comprender. Es por este motivo que se construyen técnicas que agilicen y faciliten este proceso extrayendo la información adecuada para su comprensión [8,9,10].

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Este trabajo de investigación centra sus esfuerzos en la comprensión de servicios web cuyas descripciones son WSDLs. Esta comprensión se divide en tres grandes etapas, las cuales se describen sintéticamente en las subsecciones siguientes.

Extracción de información

La comprensión de los servicios web que se desea alcanzar hace necesaria la extracción de información de los mismos. En el caso particular de los servicios web, la información es bastante escasa. Se puede leer algo en los portales donde se publican los mismos como UDDI u alguna otra página de este tipo pero no siempre alcanza para comprenderlo. La mayor fuente de información de los mismos son sus descripciones WSDL. Por ser estas descripciones un dialecto XML, es fácil utilizar un parser DOM [11] que permita recorrerlos y generar árboles de sintaxis abstractas. Luego, desde estas representaciones en árboles se realizan recorridos que permiten extraer la información que se necesita. En esta investigación particular, se extraen los nombres de los identificadores de las etiquetas más importantes (*types*, *messages*, *portTypes*, *operation*, *bindings*, *services*), los comentarios y la documentación (etiqueta *documentation*)

pero el árbol ya formado permite continuar extrayendo más información.

Análisis de la información y cálculo de métricas

Toda la información extraída es analizada para poder obtener una mensura de la comprensión que puede tener dicho servicio web a partir de su WSDL.

En el caso particular de los identificadores, se aplican algoritmos de división de palabras para aislar cada una de las palabras que lo componen, en caso de que existan, para luego analizar qué tanta semántica aportan.

De los comentarios y la documentación se realizan análisis semánticos de sus palabras más significativas, quitando las stopwords (palabras sin significado semántico) y enfocando el análisis en aquellas que aportan significado.

Sobre toda esta información obtenida se calculan métricas que ponderan este entendimiento parcial. Estas métricas son de dos tipos: i) de tamaño y ii) de semántica. Las primeras permiten conocer la dificultad del WSDL mediante el cálculo en las cantidades de cada una de sus etiquetas. Cantidad de mensajes, de tipos, de tipos complejos, de bindings, de portTypes son algunas de las métricas calculadas a partir de las cuales se puede obtener una idea rápida de la complejidad del servicio web y de cada una de sus partes. Las segundas métricas se encargan de valorar, por cada etiqueta del WSDL, la carga semántica que posee apoyándose en su nombre identificador, sus comentarios y su documentación. Para el análisis de esta carga semántica se utiliza NLTK [12,13] y una API de spelling, ambas apoyadas en el diccionario global WordNet [14] para analizar por cada palabra la carga semántica que posee y proveer sugerencias de palabras en caso que exista algún error o la palabra sea poco clara.

Grado de entendimiento del WSDL

Esta última etapa utiliza todos los valores de las métricas calculados en la etapa anterior para

cada etiqueta para calcular un grado de entendimiento global del WSDL. Basado en LSP [15], se construyó una estructura de agregación que relaciona todas las métricas semánticas otorgándole un peso a cada una de ellas y con distintos operadores propios de LSP con pesos en los arcos de la estructura, se van combinando los valores de las métricas para ir construyendo el grado de entendimiento total del servicio web. Por cuestiones de tamaño de este artículo, no se puede colocar la estructura de agregación completa pero a modo ilustrativo se muestra en la Figura 1 la estructura de agregación para el entendimiento de los mensajes.

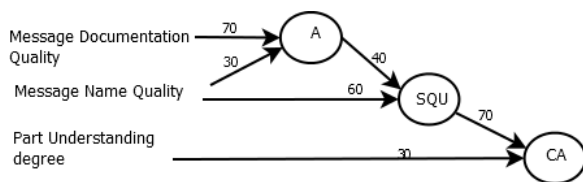


Figura 1. Estructura de agregación para medir el grado de entendimiento de los mensajes.

El lector interesado en profundizar más sobre estos conceptos puede leer [16].

Como conclusión a esta investigación se obtienen un conjunto de métricas que permiten calcular complejidades parciales por cada etiqueta principal del WSDL en tamaño y en comprensión semántica. Estas últimas sirven como base para la obtención del grado de entendimiento global del servicio web.

3. RESULTADOS

OBTENIDOS/ESPERADOS

De los resultados obtenidos en esta investigación se mencionan los más destacados:

- Se definieron las estrategias para la extracción de información desde las especificaciones WSDL de los WS.

- Se realizó un trabajo manual de análisis de múltiples WSDL para poder determinar las claves para facilitar el entendimiento de los WSDL.
- Se definieron y establecieron todas las métricas necesarias, tanto de tamaño como semánticas.
- Se construyó la estructura de agregación LSP que permite vincular las métricas y calcular el grado de entendimiento global del WSDL.
- Se construyó una herramienta que realiza la extracción de información, cálculo de métricas, cálculo del grado de entendimiento del WS usando LSP (WSDLUD) y la visualización de la información. Esta herramienta permite además comparar múltiples servicios webs de manera automática mediante el cálculo de las métricas para cada uno de ellos.

Los objetivos planteados a corto y largo plazo son:

- Crear un módulo que permita recomendar automáticamente cambios en el WSDL para incrementar o decrementar el grado de entendimiento según sea necesario. Actualmente existe un trabajo de investigación avanzando con esta temática.
- Detectar y analizar automáticamente relaciones entre WSDLs mediante el estudio de sus espacios de nombres.
- Realizar un análisis más exhaustivo sobre los tipos complejos usados en cada WS.

4. FORMACIÓN DE RECURSOS

HUMANOS

Las tareas realizadas en el contexto de la presente línea de investigación están siendo desarrolladas como parte de trabajos para optar al grado de Magister en Ingeniería de Software. En el futuro se piensa generar diferentes tesis de maestría y doctorado usando como base parte de esta investigación.

5. REFERENCIAS

- [1] WSDL Specification for W3C <https://www.w3.org/TR/wsdl>.
- [2] Erl, Thomas. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR. 2004. ISBN: 0131428985.
- [3] Erl, Thomas. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR. 2005. ISBN: 0131858580.
- [4] Storey, M. "Theories, methods and tools in program comprehension: past, present and future," *13th International Workshop on Program Comprehension (IWPC'05)*, St. Louis, MO, USA, 2005, pp. 181-191. 2005.
- [5] Von Mayrhauser, A. and Vans, A. M. "Program comprehension during software maintenance and evolution," In *Computer*, vol. 28, no. 8, pp. 44-55, Aug. 1995.
- [6] N. Gold and K. Bennett. "Program comprehension for web services". In *Program Comprehension*, 2004. Proc. 12th IEEE International Workshop on. June 2004.
- [7] L. O'Brien Lero and D. Smith. "Working session: program comprehension strategies for web service and service oriented architectures". Proc. of 12th IEEE International Workshop on Program Comprehension. 2004.
- [8] H. El Bouhissi, M. Malki, and D. Bouchiha. "A reverse engineering approach for the web service modeling ontology specifications". In *Sensor Technologies and Applications 2008*. SENSORCOMM '08. Second International Conference on, pages 819–823, Aug 2008.
- [9] C. Mao. "Towards a data complexity metric set for web service composition". In *Computer and Information Technology (CIT)*, 2011 IEEE 11th International Conference on, pages 127–131, Aug 2011.
- [10] Fangfang Liu, Yuliang Shi, Jie Yu, Tianhong Wang, Jingzhe Wu. "Measuring Similarity of Web Services Based on WSDL". IEEE International Conference on Web Services ICWS. 2010.
- [11] Parser DOM specification for W3C. <https://www.w3.org/DOM>.
- [12] Natural Language Toolkit. <https://www.nltk.org/>.
- [13] Bird, Steven and Klein, Ewan and Loper, Edward. *Natural Language Processing with Python*. 2009. ISBN: 0596516495, 9780596516499. 1era Edición. O'Reilly Media, Inc.
- [14] Wordnet, A Lexical database for English. <https://wordnet.princeton.edu/>
- [15] Olsina, L., Rossi, G.: *Measuring Web Application Quality with WebQEM*. IEEE MultiMedia, 2002 09(4), 20–29 (2002).
- [16] Mario M. Berón, Hernán Bernardis, Enrique A. Miranda, Daniel E. Riesco, Maria João Pereira, Pedro Rangel Henriques. "WSDLUD: a Metric to Measure the Understanding Degree of WSDL Descriptions". *Proceedings of the 2015 Symposium on Languages, Applications and Technologies, SLATE'15*. Madrid, España 2015.