

Reuso, Composición y Refactorización de Servicios Heterogéneos

Andres Flores, Martin Garriga, Alejandra Cechich,
Marcelo Moyano, Alan De Renzis, Diego Anabalon, Franco Corgatelli
Grupo de Investigación en Ingeniería de Software del Comahue (GIISCO)
<http://giisco.uncoma.edu.ar>
Departamento de Ingeniería de Sistemas – Facultad de Informática
Universidad Nacional del Comahue
Buenos Aires 1400, (8300) Neuquén
Contacto: [andres.flores, martin.garriga]@fi.uncoma.edu.ar

Resumen

El paradigma de Computación Orientada a Servicios (SOC), promueve el desarrollo de aplicaciones distribuidas en ambientes heterogéneos, que son construidas ensamblando o componiendo servicios reusables, que se publican a través de una red y se acceden mediante protocolos específicos. SOC ha sido ampliamente adoptado con la tecnología de Servicios Web. Existen diferentes estilos de Servicios Web que amplían las oportunidades de selección de soluciones, pero generan un desafío de evaluación y ajuste de servicios heterogéneos. Entre los estilos se encuentran los servicios SOAP (con descripciones WSDL) y los servicios RESTful (con múltiples lenguajes de descripción tal como WADL, OpenAPI, etc.). Para afrontar estos desafíos se definió un Metamodelo de Servicios Heterogéneos que permite la evaluación y composición de servicios. Además, el desarrollo de servicios para reuso afronta la necesidad de reducir la complejidad de los servicios que afecta su comprensión e interoperabilidad. Para ello, se pueden utilizar métricas de complejidad de servicios y realizar refactorizaciones hasta alcanzar la complejidad deseada. Estos desafíos también son posibles por medio del Metamodelo de Servicios, para que un proveedor reajuste sus servicios y ofrezca nuevas soluciones en base a sus desarrollos previos.

Palabras Clave: Ingeniería de Software basada en Reuso – Software Orientado a Servicios – Servicios Web – WSDL – REST.

Contexto

La línea presentada se inserta en el contexto de los siguientes proyectos y acuerdos de cooperación:

- 04/F009-2: “Reuso Orientado a Servicios – Parte II”. Financiado por UNCo. (2017-2020).
- PIP GI 11220170100951CO: “Construcción de Líneas de Productos Software guiada por Estándares de Dominio”. Financiado por CONICET. (2018-2020).
- PICT-2017-1725: “Extensión de Metamodelos de Aplicaciones Orientadas a Servicios para Descripción, evaluación y despliegue de Servicios en la Nube”. Financiado por ANPCyT. (12/2018-12/2020).
 - Investigaciones conjuntas con ISISTAN-UNICEN, Tandil.
- Acuerdo de Cooperación con el Grupo Alarcos, Escuela Superior de Informática, Universidad de Castilla-La Mancha, España.

Introducción

Actualmente la industria de software observa cómo mediante el reuso de software se puede alcanzar un proceso de desarrollo de software acelerado y confiable al basarse en artefactos software que ya han sido probados en diferentes contextos de aplicación. Para ello se

adopta el concepto denominado “tercerización”, por medio del cual se acuerdan contratos para adquisición y provisión de artefactos software reusables y se establecen relaciones comerciales entre vendedores y clientes. Por lo tanto, desde el punto de vista de un cliente implica la posibilidad de acelerar el desarrollo de un producto software para reducir el lanzamiento al mercado, y desde el punto de vista de un proveedor implica la posibilidad de observar sus productos con una perspectiva nueva que los coloque dentro del mercado de artefactos reusables.

Un paradigma que promueve altamente el reuso de software se denomina Computación Orientada a Servicios (SOC), donde la funcionalidad a ser reusada adopta la forma de servicios, o unidades lógicas que presentan entornos heterogéneos de ejecución y pueden ser ensambladas para formar otras unidades lógicas de mayor nivel de abstracción que resuelvan (directamente o en parte) los procesos de negocios para un contexto de aplicación [SH05,PTDL07]. El paradigma SOC encontró una plataforma potencial de aprovechamiento mediante la Web, desde donde se desarrolló la tecnología de Servicios Web [NSS03, Wetal05], con notaciones formales específicas para la descripción las interfaces de servicios. Así el paradigma SOC bajo la implementación con Servicios Web ha logrado su amplia adopción en la industria, principalmente bajo la flexibilidad de ejecución remota que permite a las compañías descentralizar aún más sus procesos de negocios y la ventaja de que las plataformas específicas de ejecución se encuentran ocultas, por lo cual no se requiere de inversiones adicionales en tecnología (incluyendo costos y esfuerzo de aprendizaje) al adquirir funcionalidad de terceras partes. El beneficio que la tecnología de Servicios Web provee al paradigma SOC se ha denominado “relación sin responsabilidad”, donde una aplicación cliente no requiere asumir

cómo se ha implementado el servicio con el que se comunica. Sin embargo, los proveedores de servicios tienen la responsabilidad de evaluar la calidad de los productos ofrecidos como servicios y los consumidores de servicios a su vez deben ser capaces de identificar tal calidad que influirá sobre las aplicaciones en desarrollo.

El funcionamiento concreto del paradigma SOC se basa en la Arquitectura orientada a Servicios (SOA) [SH05] que se encuentra compuesta por tres actores principales: un proveedor, un consumidor y un registro de servicios; donde el proveedor desarrolla y publica servicios en el registro, para que luego el consumidor busque servicios y establezca una comunicación con el proveedor. Sin embargo, la búsqueda de servicios publicados en un registro UDDI (según la tecnología de servicios Web) [OASIS04], en general requiere invertir un esfuerzo considerable para distinguir servicios candidatos que satisfagan los requerimientos de la aplicación cliente [NSS03, Wetal05]. En particular, cuando varios candidatos ofrecen funcionalidades similares se requieren métodos eficientes de selección de servicios que discriminen tanto aspectos funcionales como no-funcionales, considerando además las interacciones válidas para un servicio candidato en función de los procesos de negocio que implementará la aplicación cliente. En particular el ensamblaje de servicios considerando procesos de negocio e interoperabilidad de servicios plantea el uso de dos conceptos de reciente investigación: Orquestación y Coreografía de servicios [P03, Wetal05]. El primero relacionado a una aplicación particular que describe un proceso de negocios específico, y el segundo relacionado a las interacciones válidas que pueden ocurrir entre distintos servicios predestinados a intervenir en una colaboración.

Los Servicios Web pueden ser desarrollados mediante diferentes estilos

distintivos, entre los cuales se encuentran los servicios SOAP y los servicios RESTful. Los servicios SOAP se describen mediante el lenguaje WSDL (Web Service Description Language) que permite especificar las operaciones e intercambio de mensajes para un Servicio Web. Existen diferentes versiones del lenguaje WSDL (1.0, 1.1, 2.0), siendo la versión WSDL 2.0 definido como estándar OMG. Los servicios SOAP se consideran servicios “grandes”, debido a que el intercambio de mensajes se realiza por medio de documentos XML, que también es la base para el lenguaje WSDL. Esto hace que se deba afrontar una sobrecarga de transferencia de archivos. Los servicios RESTful se pueden describir mediante diferentes lenguajes que intentan solventar la sobrecarga mencionada, mediante el intercambio de mensajes livianos, generalmente usando JSON o YAML. El primer intento de notación para REST fue WADL (Web Application Description Language) como analogía de WSDL, y también basado en XML. La industria ha propuesto diferentes lenguajes, entre los que se encuentra OpenAPI/Swagger, REST API Modeling Language (RAML), o MIT’s API Blueprint, entre otros. Evaluar y consumir tal variedad de servicios heterogéneos podría reducir el espacio de solución de los servicios a consumir, de acuerdo con la tecnología que se pueda soportar. De este modo, la amplia gama de servicios probablemente reusables, se vuelve seriamente limitada. Para ello, hemos desarrollado un Metamodelo de Servicios Heterogéneos basado en estándares para ampliar la gama de compatibilidad de servicios de diversas tecnologías, que pudieran ser potenciales candidatos para aplicaciones consumidoras. Los principales estándares para la descripción del servicio incluyen: WSDL 2.0, WADL y OpenAPI, considerando así ambos estilos de servicios, basados en SOAP y REST.

Por otro lado, los proveedores de servicios se ven en la necesidad de ajustar sus servicios de acuerdo a los requerimientos de los clientes. En particular, la complejidad del servicio puede afectar la comprensión de sus aspectos funcionales, y reducir su descubrimiento e interoperabilidad. Para ello, se ha investigado sobre métricas de complejidad de servicios y la posibilidad de realizar refactorizaciones de servicios, hasta alcanzar servicios con una complejidad deseada. Esto también permitiría que los servicios SOAP pudieran ser migrados a servicios RESTful, que son más livianos en su consumo desde una aplicación cliente. Aunque incluso los servicios REST podrían superar los niveles deseables de complejidad, afectando seriamente su reuso, por lo cual también son susceptibles de ser refactorizados con los mismos propósitos.

Líneas de Investigación, Desarrollo e Innovación

El perfil de esta línea puede definirse en base a las actividades de investigación y transferencia, a las que da soporte el grupo GIISCo. Los temas específicos consideran los desafíos diferentes asociados al crecimiento de la Tecnología de la Información y las Comunicaciones. Actualmente, abordamos los siguientes aspectos:

- Compatibilidad y selección de servicios.
- Adaptación y Composición de servicios.
- Complejidad y legibilidad de servicios
- Refactorización de servicios
- Testing de servicios.
- Herramientas para evaluación, selección, composición y testing de servicios.
- Definición de aplicaciones en dominios específicos.

Resultados y Objetivos

En [FCGMRAC19] hemos enumerado una serie de contribuciones anteriores. Durante el año 2019, hemos profundizado la investigación en aspectos de *evaluación de compatibilidad y complejidad de servicios*, generando métodos y herramientas enfocados en las interfaces y comportamiento dinámico de los servicios [GF19]. Parte de este avance se ha efectuado en colaboración con investigadores de ISISTAN (UNICEN) [MZFM19], y del Politecnico di Milano (Italia) [BMGGQ19], [TNLGDG19]. Las líneas de investigación convergen en el tratamiento del desarrollo de software basado en el reuso de servicios desde la perspectiva de las aplicaciones orientadas a servicios. Una aplicación orientada a servicios implica una solución de negocio que consume servicios de uno o más proveedores y los integra en un proceso de negocio [SW04]. Además puede verse como una aplicación basada en componentes que integra dos tipos de componentes: internos localmente empujados en la aplicación, y externos estática o dinámicamente enlazados a algún servicio [CMZC14]. El Metamodelo de Servicios Heterogéneos habilita la posibilidad de reusar Servicios Web de diferentes estilos y tecnologías, extendiendo las oportunidades para la solución de aplicaciones cliente. Las capacidades que están desarrollando para la composición de servicios heterogéneos abren otro abanico de posibilidades cuando los servicios candidatos recuperados no proveen las funcionalidades requeridas. El Metamodelo de Servicios ha sido complementado con nuestro framework para evaluación y selección de servicios desarrollado en trabajos previos [GRLFMCZ18]. Por otro lado la refactorización de servicios basado en métricas de complejidad permite no solamente la adecuación de Servicios Web que un proveedor ofrece, sino

también la posibilidad de estudiar escenarios de composición de servicios, para validar nuestra propuesta. Se prevee la aplicación de estos modelos y las herramientas de soporte a dominios específicos, con particular énfasis en aquellos que requieran rigurosidad como aporte de validación efectiva. La visión de esta línea de investigación se resume en:

“Definir técnicas y herramientas para la mejora del desarrollo de software, en función del reuso de servicios web. La definición de modelos de evaluación, selección y refactorización de servicios, y la posibilidad de composición de servicios”.

Formación de Recursos Humanos

Este proyecto se compone de 11 investigadores, entre los que se cuentan docentes y estudiantes del Grupo GIISCO de UNComa y asesores externos. Algunos de los docentes-investigadores se encuentran realizando carreras de postgrado. Se cuenta actualmente con 3 doctores (1 investigador adjunto y 1 investigador asistente CONICET), 2 doctorandos y 1 maestrando entre los miembros del proyecto. Dirección de Tesis de Grado durante 2019: 5 tesis.

Referencias

- [BMGGQ19] Baresi, L., Mendonça, DF., Garriga, M., Guinea, S. Quattrocchi, G. (2019). *A Unified Model for the Mobile-Edge-Cloud Continuum*. ACM Transactions on Internet Technology (TOIT) 19 (2): 1-21. **(Indexed SCI IF JCR2018: 2.382)**.
- [CMZC14] Crasso, M., Mateos, C., Zunino, A., Campo, M. (2014). *EasySOC: Making Web Service Outsourcing Easier*. International Journal on Information Sciences, 259: 452–473.
- [GRLFMCZ18] Garriga, M., De Renzis, A., Lizarralde, I., Flores, A., Mateos, C., Cechich, A., Zunino, A. (2018). A Structural-Semantic Web Service Selection Approach to Improve Retrievability of Web Services. Information Systems Frontiers, 20(6): 1319-1344 **(Indexed SCI IF JCR2017: 3.232)**.
- [GF19] Garriga, M., Flores, A. (2019). *Standards-driven Metamodel to Increase Retrievability of Heterogeneous Services*. 34th ACM/SIGAPP Symposium on Applied Computing, pp. 2507-2514. **(Qualis A1)**.
- [FCGMRAC19] Flores, A., Garriga, M., Cechich, A., Moyano, M., De Renzis, A., Anabalon, D., Corgatelli, F. (2019). *Reuso de Servicios basado en Análisis de Compatibilidad y Complejidad*. WICC'19, XXI Workshop de Investigadores en Ciencias de la Computación. Abril, San Juan.
- [MZFM19] Mateos, C., Zunino, A., Flores, A., Misra, S. (2019). *COBOL Systems Migration to SOA: Assessing Antipatterns and Complexity*. Information Technology and Control 48 (1): 71-89 **(Indexed SCI IF JCR2018: 0.707)**.
- [NSS03] Nagappan, R.; Skoczylas, R.; Sriganesh, R. (2003). *Developing Java™ Web Services: Architecting and Developing Secure Web Services Using Java*. Wiley Publishing Inc.
- [OASIS04] OASIS Consortium (2004). *UDDI Version 3.0.2*. UDDI Spec Technical Committee Draft, October.
- [P03] Peltz, C. (2003). *Web Services Orchestration and Choreography*. IEEE Computer, 36(10): 46–52.
- [PTDL07] Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F. (2007). *Service-Oriented Computing: State of the Art and Research Challenges*. IEEE Computer, 40(11): 38–45.
- [SH05] Singh M.; Huhns, M. (2005). *Service-oriented computing: Key concepts and principles*. IEEE Internet Computing, 9(1): 75–81.
- [SW04] Sprott, D.; Wilkes, L. (2004). *Understanding Service-Oriented Architecture*. The Architecture Journal. MSDN Library. Microsoft Corporation, <http://msdn.microsoft.com/en-us/library/aa480021.aspx>
- [TNLGDG19] Tsigkanos, C. Nenzi, L. Loreti, M. Garriga, M. Dustdar, S. Ghezzi, C. (2019). *Inferring Analyzable Models from Trajectories of Spatially-Distributed Internet of Things*. IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), pp. 100-106.
- [W3C05] W3C Candidate Recommendation (2005) *Web Services Choreography Description Language Version 1.0*. <http://www.w3.org/TR/ws-cdl-10/>
- [Wetal05] Weerawarana, S.; Curbera, F.; Leymann, F.; Storey, T.; Ferguson, D. (2005). *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Prentice Hall PTR.