

## Evaluación de Compatibilidad y Complejidad para el Reuso de Servicios

Andres Flores, Alejandra Cechich, Martin Garriga,  
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, alejandra.cechich]@fi.uncoma.edu.ar

### Resumen

El reuso de artefactos software brinda oportunidades para proveedores y clientes, tanto para acelerar el proceso de desarrollo de software como para establecer oferta de productos reusables. 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 bajo su implementación con la tecnología de Servicios Web, que provee flexibilidad de ejecución remota que oculta las plataformas específicas de ejecución y permite descentralizar los procesos de negocios. SOC requiere la publicación de servicios en un registro (UDDI de acuerdo a Servicios Web), los cuales luego son identificados y evaluados para una aplicación en desarrollo. Sin embargo, aún este proceso necesita métodos exhaustivos y eficientes, tanto para identificación como para selección de servicios, en el cual se puede considerar la aplicación de técnicas de Pruebas de Software y el uso de dos conceptos actuales: Orquestación y Coreografía de servicios.

**Palabras Clave:** Ingeniería de Software basada en Reuso – Software Orientado a Servicios – Servicios Web – Calidad de Software – Verificación y Validación.

### Contexto

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

- 04/F009: “Reuso Orientado a Servicios – Parte II”, sub-proyecto del Programa “Desarrollo de Software basado en Reuso – Parte II”. Financiado por UNCo. (2017-2020).
- PIP 2017-2019 11220170100951CO: “Construcción de Líneas de Productos Software guiada por Estándares de Dominio”. Financiado por CONICET.
- Acuerdo de Cooperación con ISISTAN-CONICET, 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], cuya base es el esquema estándar XML y el Lenguaje de Descripción de Servicios Web (WSDL), que facilitan ampliamente el desarrollo y mantenimiento de especificaciones formales 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. Para ello, las estrategias de Pruebas de Software deben ajustarse a este contexto específico donde los servicios están acordados como cajas negras que sólo permite evaluar el comportamiento y cualidades observables externamente [BDN10, Z08].

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. En este contexto se cuenta actualmente con diversos lenguajes de descripción y frameworks de ejecución, tales como BPEL4WS [OASIS07], y WSCDL (Web Services Choreography Description Language) [W3C05].

### **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:

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

### **Resultados y Objetivos**

En [FCGMRAC17] hemos enumerado una serie de contribuciones anteriores. Durante el año 2017, 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. Este avance se ha efectuado en colaboración con investigadores de ISISTAN (UNICEN) [AFMZM17, GRFCZ17, MZMAF17a, MZMAF17b, RGFCMZ17a, RGFCMZ17b].

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 empotrados en la aplicación, y externos estática o dinámicamente enlazados a algún servicio [CMZC14]. No solamente se enfocará en el reuso de servicios individuales, sino también en la composición de servicios como forma de tercerizar una funcionalidad. Se adoptará la visión de proceso de negocio para la definición de comportamiento, donde se aplicará testing de servicios para una evaluación dinámica. Se complementará el modelo de selección y composición de servicios mediante las últimas plataformas y avances tecnológicos incluyendo semántica y estandarización. 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 identificación y selección 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 CONICET, y 1 investigador asistente CONICET), 2 doctorandos (becarios CONICET), 1 maestrando, y 1 becario EVC-CIN entre los miembros del proyecto. Dirección de Tesis de Grado durante 2017: 3 tesis.

## Referencias

- [AFMZM17] Anabalón, D., Flores, A., Mateos, C., Zunino, A., Misra, S. (2017). *Controlling Complexity of Web Services Interfaces through a Metrics-driven Approach*. ICCNI'17, International Conference on Computing Networking and Informatics. pp. 1-9. IEEE Computing Society Press. Ota, Nigeria.
- [BDN10] Baresi, L.; Di Nitto, E. (2010). *Test and Analysis of Web Services*. Springer.
- [CMZC14] Crasso, M., Mateos, C., Zunino, A., Campo, M. (2014). *EasySOC: Making Web Service Outsourcing Easier*. International Journal on Information Sciences, 259: 452–473.
- [FCGMRAC17] Flores, A., Cechich, A., Garriga, M., Moyano, M., De Renzis, A., Anabalón, D., Corgatelli, F. (2017). *Reuso Orientado a Servicios: Compatibilidad y Complejidad de Servicios*. WICC'17, Workshop de Investigadores en Ciencias de la Computación. pp. 622-626. Buenos Aires.
- [GRFCZ17] Garriga, M., De Renzis, A., Flores, A., Cechich, A., Zunino, A. (2017). *Assessing Web Services Interfaces with Lightweight Semantic Basis*. Computing & Informatics, 36(5):1173-1206. (**Indexed SCI IF JCR2015: 0.504**).
- [MZMAF17a] Mateos, C., Zunino, A., Misra, S., Anabalón, D., Flores, A. (2017). *Managing Web Service Interface Complexity via an OO Metric-based Early Approach*. CLEI Electronic Journal, 20(3): 1-22. (**Latindex**).
- [MZMAF17b] Mateos, C., Zunino, A., Misra, S., Anabalón, D., Flores, A. (2017). *Migration from COBOL to SOA: Measuring the Impact on Web Services Interfaces Complexity*. ICIST'17, International Conference on Information and Software Technologies, pp. 266-279. Springer-Verlag CCIS. Lithuania.
- [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.
- [OASIS07] OASIS Standard (2007). *Web Services Business Process Execution Language Version 2.0*. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

- [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.
- [RGFCMZ17a] De Renzis, A., Garriga, M., Flores, A., Cechich, A., Mateos, C., Zunino, A. (2017). *Assessing Readability of Web Service Interfaces*. CLEI Electronic Journal, 20(2): 1-23. (**Latindex**).
- [RGFCMZ17b] De Renzis, A., Garriga, M., Flores, A., Cechich, A., Mateos, C., Zunino, A. (2017). *A Domain Independent Readability Metric for Web Service Descriptions*. Computer Standards & Interfaces, 50: 124–141. Elsevier (**Indexed SCI, IF JCR2016: 1,268**).
- [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>
- [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.
- [Z08] Zhou, X. (2008). *Testing and Verifying Web Services. From the Researcher's Perspective*. VDM Verlag.