

Sistema multiagente para la composición de servicios web semánticos

Jorge Giraldo Plaza,^{*a} Jaime Guzmán Luna,^a Albert Ledesma Castillo^a
(a) *Universidad Nacional de Colombia, Medellín.*
**jegiraldp@unalmed.edu.co*

(Recibido: Noviembre 10 de 2007 - Aceptado: Mayo 15 de 2008)

RESUMEN

La Composición de Servicios Web permite brindar valor agregado a un Servicio Web con el fin de satisfacer las necesidades del cliente, se complementa con las bondades que ofrece la Web Semántica y la Inteligencia Artificial. Este artículo expone la implementación de la metodología MAS-CommonKADS para el desarrollo de un Sistema de Composición Automática de Servicios Web Semánticos.

PALABRAS CLAVE: Sistemas Multiagente, Planificación, Servicios Web.

Multi-Agent system for semantic web services composition

ABSTRACT

The Web Services Composition allows giving some added values to a Web Service with the purpose of satisfying the user's requirements. It is complemented with the goodness offered by the Semantic Web and the Artificial intelligence. This paper shows the MAS-CommonKADS implementation for the development of an automatic composition system for Semantic Web Services.

KEYWORDS: Multiagent System, Planning, Web Services.

1. INTRODUCCIÓN

Los Servicios Web son componentes independientes de la plataforma que pueden ser fácilmente publicados, localizados e invocados mediante lenguajes y protocolos web estándares [1], tales como: XML (eXtensible Markup Language) [2], SOAP (Simple Object Access Protocol) [3], UDDI (Universal Description Discovery and Integration) [4] y WSDL (Web Services Definition Service) [5].

La composición de Servicios Web aborda situaciones donde los requerimientos de información planteados por el usuario no se alcanzan a través de un servicio web simple y es necesario combinar varios de ellos mediante un proceso de composición para generar un servicio compuesto que pueda satisfacer dichos objetivos [6]. Dependiendo del punto de vista del diseño de la composición, puede verse ésta como una Coreografía [7] u Orquestación [8].

Con base en los fundamentos de la programación orientada a servicios (SOC) [9] y su arquitectura (donde se emplean los Servicios Web como los elementos fundamentales), se identifican posibles roles [10] que los Servicios Web juegan dentro del proceso de composición [11], siendo posible la automatización de la composición por medio de técnicas de Inteligencia Artificial. Distintas tecnologías han aportado soluciones de composición de servicios web [12], entre las que se destacan los Sistemas Multiagente (SMA) [13], que gracias a sus características de modularidad y distribución de funcionalidades entre nodos de red [14], son apropiados para la composición de servicios web.

El artículo propone la arquitectura de un SMA para la composición automática de servicios web semánticos, denominado INDIGO, para tal fin el documento se estructura de la siguiente manera: La Sección 2 presenta el estado del arte de las tecnologías de los servicios web y la composición de servicios web; La Sección 3, cita algunos trabajos relacionados con la composición automática de servicios web. La Sección 4 presentan parte de la documentación de la metodología MAS-CommonKADS. La Sección 5 expone aspectos de implementación del prototipo.

Por último, en la Sección 6 se presentan conclusiones y los trabajos futuros.

2. TECNOLOGÍAS DE LOS SERVICIOS WEB Y COMPOSICIÓN DE SERVICIOS WEB

En el marco de las aplicaciones académicas e industriales [15] de los Servicios Web, distintas tecnologías abordan problemas tales como: la composición [16], la seguridad [17] y los aspectos semánticos de los mismos [18]. Por el lado industrial se destacan los estándares internacionales, como el WSDL [5] y el BPEL4WS [19] y del ámbito académico, en el área de la IA, las técnicas como la planificación y mecanismos de razonamiento como los SMA y la web semántica [20].

La Planificación [21] en IA es una de las técnicas que en los últimos años ha sido utilizada para llevar a cabo la composición de servicios web semánticos. Básicamente un problema de planificación en IA tiene los siguientes componentes [22]: (i) Una descripción de las posibles acciones a realizar por parte de los operadores, las cuales pueden ser especificadas en algún lenguaje formal. (ii) Una descripción del estado actual del mundo, es decir determinar el valor de las variables en un momento dado. Por último, (iii) una descripción de una serie de objetivos deseados por el usuario.

La web semántica [23] emplea ontologías para describir en su totalidad las características de los Servicios Web, tanto simples como compuestos. OWL-S [24] (Originalmente DAML-S [25]) es una ontología de Servicios Web, la cual permite describir semánticamente su flujo de procesos y la forma en la que el servicio puede ser accedido.

Trabajos en el área de los SMA han permitido que se cuente con metodologías apropiadas para la documentación de su desarrollo, tal es el caso de la metodología MAS-CommonKADS [26]. Varias aplicaciones en este campo [27], han empleado esta metodología para abordar soluciones a problemas específicos.

En el proceso de Composición de Servicios Web, se identifican dos pasos generales: (i) la identificación de los requerimientos del usuario por medio de un método por el cual se pueda generar un plan o flujo de actividades y (ii) la selección de los Servicios Web para su posterior ejecución y monitoreo.

Las investigaciones en el área de la composición se enfocan básicamente en tres áreas del conocimiento [22]: vía procesos de negocios [28], planificación en IA [29] y la Síntesis de Programas [30]. Si bien no es posible saber cuál predomina sobre la otra, la planificación IA ha demostrado su efectividad en este campo.

3. TRABAJOS RELACIONADOS

En [31] se utiliza una técnica de Planificación basada en reglas, con la cual se generan servicios compuestos desde una descripción declarativa en alto nivel.

SWORD [32] es una herramienta para construir composiciones de servicios web usando reglas basadas en un plan generado. No emplea las tecnologías emergentes en los Servicios Web tales como WSDL y DAML-S [36] (nombre anterior de OWL-S), en lugar de ello emplea el Modelo Entidad-Relación (E-R) para especificar los servicios web.

En [33] se presenta una adaptación de Golog [34] para la programación en la Web Semántica. Golog es un lenguaje de programación lógico, construido teniendo como base el cálculo de situaciones [35]. Golog declara un formalismo natural para la composición automática de tareas primitivas de los servicios web.

En [36] se presenta un planificador que compone servicios web atómicos descritos en DAML-S [36], su mayor aporte se centra en determinar que a partir de anotaciones semánticas es posible componer servicios web. En este proyecto se construye un planificador para lograr descomposición y validación de tareas que usa el conocido STRIPS [37]. También se emplea la plataforma “Sistema Shell experto en java” (JESS) [38] para implementar el planificador y un conjunto de reglas JESS que son trasladadas de

descripciones DAML-S a operadores del mundo de la planificación.

En [39] se propone el desarrollo de una herramienta basada en PDDL [40] para la composición. El principal aporte es proveer un método práctico en escenarios complejos. El problema del proceso de composición de servicios web es presentado mediante PDDL, lo que hace que pueda ser ejecutado en la mayoría de los planificadores existentes.

En [41] se presenta una arquitectura multiagente para la composición de servicios web, que se implementa en la plataforma AGORA [42]. Se emplea el enfoque multiagente para representar los distintos roles que intervienen en el proceso de composición de servicios web.

En [43], se propone el uso de una técnica de planificación HTN [44], empleando una adaptación del planificador SHOP2 [45] para la composición de servicios web en dominios en los cuales la información sobre el estado inicial no es completa. El lenguaje nativo para el manejo del planificador es PDDL (Lenguaje de definición de dominios de planificación) [40].

4. MODELADO DEL SMA

MAS-CommonKADS [46], es una metodología de propósito general para el desarrollo de SMA [13], como resultado de la extensión de CommonKADS, la cual incluye técnicas de ingeniería de protocolos y orientación a objetos.

4.1 Fase de Conceptualización y Análisis

El usuario del sistema juega un rol determinante en el proceso de composición, ya que éste declara los objetivos que desea se lleven a cabo, además de que activa el proceso de composición. Bajo este orden de ideas se identificaron los siguientes casos de uso: (i) Describir Objetivos, el cual permite identificar cuáles son las características del servicio web compuesto mediante una técnica de relación de objetivos; (ii) Componer Servicio, permite definir la secuencia de servicios que se deberán ejecutar para alcanzar los objetivos solicitados por el usuario; (iii) Ejecutar Servicio, este caso se orienta a la ejecución de la secuencia

de servicios que representa al servicio web compuesto generado por el sistema.

Los agentes que conforman el sistema y que permiten realizar los anteriores casos de uso, son: (i) Agente Usuario, éste está asociado con el caso de uso Describir Objetivos y representa la interfaz con la cual interactúan los actores humanos para ingresar los objetivos; (ii) El Agente TraductorA, el cual está asociado al caso de uso Componer Servicios; Selecciona y transforma la descripción OWLS de los servicios disponibles en el repositorio en el contexto del problema y genera los archivos que definen un problema de planificación en el lenguaje PDDL; (iii) El Agente Compositor, está también asociado al caso de uso Componer Servicios y es el encargado de generar la secuencia de ejecución de servicios (plan); (iv) El Agente TraductorB, asociado al caso de uso Ejecutar Servicio, permite transformar la secuencia de ejecución en un servicio web compuesto OWLS; (v) El Agente Ejecutor, asociado al caso de uso Ejecutor Servicio, realiza el proceso de ejecución del servicio web compuesto OWLS.

4.2 Fase de Diseño.

Esta fase pretende transformar las especificaciones de los modelos del análisis [26] en un nivel de abstracción cercano a la especificación de la implementación.

El sistema se complementa con varios sistemas externos; uno es el SimPlanner [47], que es un planificador IA independiente del dominio, el cual es controlado a través del Agente compositor. Otro es el sistema ejecutor, que emplea parte de la API OWL-S [48], provee de manera programática lectura, escritura y ejecución de documentos OWL-S. Es controlado a través del Agente Ejecutor. El repositorio de servicios web, registra los servicios y permite al Agente TraductorA acceder a sus descripciones.

En la arquitectura planteada, el usuario interactúa con una interfaz gráfica, controlada por el Agente usuario, la cual permite ingresar los objetivos y ejecutar la orden de inicio del proceso de composición. Para ello, debe generarse una

definición del problema (estado inicial y estado objetivo), así el planificador puede saber los objetivos que persigue.

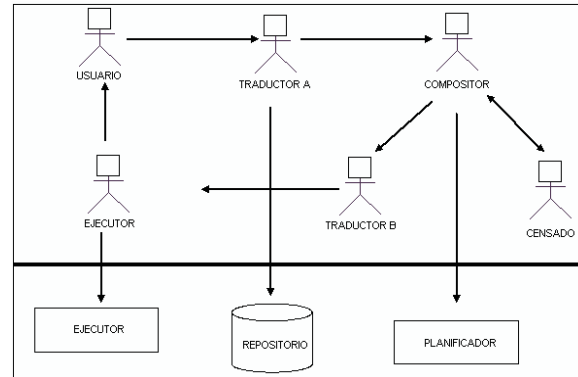


Fig. 1. Arquitectura general del SMA

5. ASPECTOS DE IMPLEMENTACIÓN DEL PROTOTIPO

INDIGO tiene implementados los agentes Interfaz de Usuario, Planificador, TraductorA y Compositor, los cuales ya se encuentran en pruebas de validación. El sistema se está desarrollando en la plataforma para SMA JADE “Java Agent DEvelopment Framework” [49], la cual está soportada en el lenguaje JAVA [50].

En la figura 2, se detalla la interfaz gráfica asociada al agente Usuario, la cual se compone de varias listas desplegables que se van activando de acuerdo al número de objetivos o condiciones iniciales que necesiten. Tiene como evento principal “Componer Servicio”. Una vez se genera esta orden, los objetivos son enviados al agente TraductorA, para que se generen los archivos PDDL que representan el problema de composición como un problema de planificación (archivos dominio.pddl y problema.pddl).

Un punto fuerte del Sistema INDIGO es que el TraductorA, está capacitado para leer una descripción en OWL-S y generar los archivos PDDL respectivos, lo cual garantiza que se puede emplear cualquier tipo de planificador que soporte este lenguaje, en reemplazo del planificador SimPlanner utilizado en este sistema.

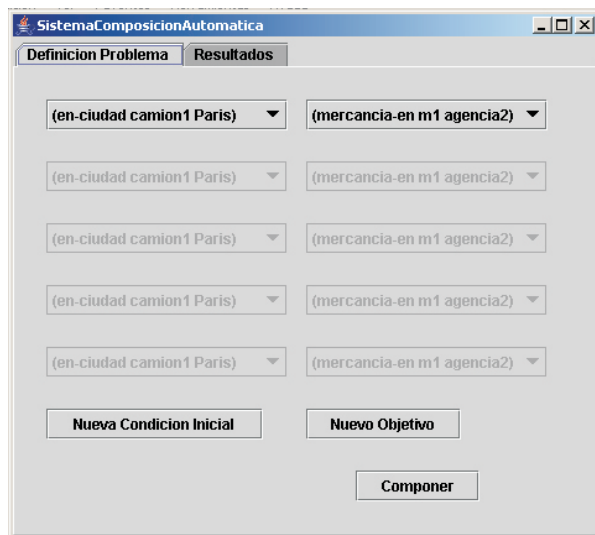


Fig. 2. Interfaz gráfica del Agente Usuario

6. CONCLUSIONES Y TRABAJOS FUTUROS

Aunque no se puede definir un enfoque particular para la composición de servicios, sí se puede afirmar que dos de las líneas más optimistas de trabajo en este campo, parecen estar en: los Sistemas Multiagentes y las técnicas de Planificación en Inteligencia Artificial, donde según el análisis preliminar los primeros son apropiados para llevar a cabo de manera colaborativa las tareas específicas que se presentan en este proceso; y las segundas, permiten generar la secuencia de ejecución de servicios para lograr los objetivos requeridos por el usuario, cuando no existe un servicio que los satisfice.

Como una propuesta de aplicación de las herramientas anteriormente expuestas, se presenta el sistema INDIGO, el cual recibe del usuario la especificación de los objetivos del servicio requerido, dando inicio a su vez al proceso de composición, donde se traduce primero los objetivos junto con las especificaciones de los servicios OWL-S, a una especificación de un problema de Planificación PDDL, que luego, mediante el uso de un planificador IA se obtiene la secuencia de ejecución que representa el servicio

compuesto que cumple con los objetivos propuestos por el usuario para su posterior ejecución.

Como trabajo futuro inmediato, se plantea completar la fase de implementación del sistema, adicionando la funcionalidad de ejecución del servicio compuesto, para tal fin se planea implementar los agentes TraductorB y el Ejecutor. Adicionalmente, se está implementando un ambiente de evaluación consistente en recrear bajo la óptica de servicios web semánticos diferentes problemas de planificación propuestos en las competencias del IPC [51]. Como trabajo futuro a largo plazo, se plantea estudiar a fondo dos problemas relacionados con la composición de servicios como son: (i) El manejo de la información incompleta durante la composición, ya que actualmente INDIGO supone que conoce toda la información y en el mundo real la información almacenada en las bases de datos de los servicios web no es conocida de manera a priori. Bajo este enfoque, la idea consiste en integrar a nuestro sistema técnicas de sensado durante el proceso de composición. (ii) El manejo de la especificación parcial semántica de los servicios, ya que hasta ahora nuestro sistema sólo maneja servicios web especificados semánticamente, y en realidad existen muchos servicios web que no están marcados semánticamente y se hace necesario que el sistema sea capaz de marcarlos automáticamente para utilizarlos. En esta línea la idea es evaluar la integración de técnicas de aprendizaje de máquinas al sistema de composición INDIGO.

7. AGRADECIMIENTOS

Este trabajo está parcialmente basado en el proyecto “Modelo Multiagente de Planificación bajo Conocimiento Incompleto para la composición Automática de Servicios Web - 201001006096”, asociado al grupo de Investigación SINTELWEB, de la Universidad Nacional de Colombia y la tesis Doctoral “Modelo Multiagente de Planificación y Ejecución Concurrente para la Composición de Servicios Web Semánticos en Ambientes Parcialmente Observables”, Apoyado por Colciencias y la Universidad Nacional.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] B. Srivastava, J. Koehler, *Web Services Composition Current Solutions and Open Problems*, IBM research laboratory, 2003.
- [2] XML Core Working Group Public Page, *XML 1.1 Recommendation*, Disponible en: <http://www.w3.org/XML/Core/#Publications>.
- [3] N. Miltra, *SOAP Version 1.2 Part 0: Primer W3C Recommendation*, disponible en: <http://www.w3.org/TR/2003/REC-soap12-art0-20030624/>, 24 Junio 2003.
- [4] L. Clement, A. Hately, C. Riegen, T. Rogers, *UDDI Spec Technical Committee*, OASIS, 2004.
- [5] *Web Services Description Language (WSDL) 1.1. W3C Note 15 March 2001*, Disponible en: <http://www.w3.org/TR/wsdl>
- [6] B. Srivastava, J. Koehler, *Planning with workflows An emerging paradigm for web services composition*, IBM research laboratory, 2004.
- [7] C. PELTZ, *Web Services Orchestration and Choreography*, Hewlett-Packard Company, Publicado por IEEE Computer Society, 2003.
- [8] D. BERARDI, *Automatic Service Composition. Models, Techniques and Tools*, Ph.D. Thesis, Universidad La Sapienza, Diciembre 2, 2004.
- [9] M. Papazoglou, *Service Oriented Computing: Concepts, Characteristics and Directions*, Tilburg University, INFOLAB. Dept. of Information System and Manegement. mikep@uvt.nl
- [10] R. Lara, H. Lausen, S. Arroyo, *Semantic Web Services: Description Requirements and Current Technologies*, 2003.
- [11] M. Paloucci, Kawamura, *Sycara: Semantic Matching of Web Services Capabilities*, In Int. Semantic Web Conference, Sardinia, Italy, pp. 333-347, June 2002.
- [12] N. Milanovic, M. Malek, *Current Solutions for Web Service Composition*, Published by the IEEE Computer Society, November, December 2004.
- [13] M. Wooldridge, *An Introduction to MultiAgent Systems*, Disponible en: <http://www.csc.liv.ac.uk/~mjlw/pubs/imas/>, 2002.
- [14] J. Bryson, D. Martin, S. Mcilraith, *Agent-Based Composition Services in DAML-S. The Behavior-Oriented Design of an Intelligent Semantic Web*, 2004.
- [15] S. SCOTT, *Creación de Servicios Web XML para la Plataforma Microsoft.NET*, McGrawHill, 2002.
- [16] U. Kuter, E. Sirin, D. Nau, B. Parsia, *Informating Gathering During Planning for Web Services Composition*, Department of Computer Science, University of Maryland, 2003.
- [17] T. Fernando, *Desarrollo de aplicaciones seguras: Servicios Web y Seguridad Informática*, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 2003.
- [18] H. Wang, *Web Service: Problems and Future Directions*, Department of Computer Science and Technology, Nanjing University, 2004.
- [19] F. Curbera, *Business Process Execution Language for Web Services*, disponible en: <http://www106.ibm.com/developerworks/webserices/library/wsbpel/>, 2002.
- [20] T. Berners-Lee, J. Hendler, And O. Lassila, *The Semantic Web*, Scientific American.com., May 17, 2001.
- [21] M. Carman, L. Serafin, P. Traverso, *Web Service Composition as Planning*, In ICAPS Workshop 2003: Planning for web services.
- [22] J. PEER, *Web Services Composition as AI Planning A survey*. Second revised version, March 22, 2005.

- [23] C. LUIS, *Internet Invisible y la web semántica*, Revista Tradumática, Noviembre, 2003.
- [24] OWL-S 1.1 Release. Disponible en: <http://www.daml.org/services/owl-s/1.1/>, Consultado Septiembre de 2005.
- [25] A. Ankolenkar, M. Burstein, *DAML-S: Semantic Markup for Web Services*, In The First International Semantic Web Conference (ISWC), Sardinia (Italy), 2003.
- [26] C. Iglesias, *Definición de una Metodología para el Desarrollo de Sistemas Multiagente*, Tesis Doctoral, Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 1998.
- [27] G. Barrera-Sanabria, *Applying the MAS-CommonKADS Methodology to the Flights Reservation Problem: Integrating Coordination and Expertise*, Proceedings of the Fifth Joint Conference on Knowledge-Based Software Engineering, Frontiers of Artificial Intelligence and Applications Series, IOS Press, 2002.
- [28] R. Rodríguez, F. Sánchez, J. Conejero, *Modelando Procesos de Negocio Web desde una Perspectiva Orientada a Aspectos*, Software Engineering Group, Universidad de Extremadura, 2002.
- [29] R. Eliane, K. Kevin, *Inteligencia Artificial. Segunda edición*, McGrawHill, Revista Tradumática, Noviembre 2003, Madrid 1994.
- [30] J. Rao, *A Survey of Automated Web Services Composition Methods*, Department of Computer and Technology, 2003.
- [31] B. Medjahed, A. Bouguettaya, A. K. Elmagarmid, *Composing Web Services on the Semantic Web*, The VLDB Journal, Vol. 12 No. 4, November, 2003.
- [32] S. Ponnekanti, A. Fox, *SWORD: A Developer Toolkit for Web Service Composition*, in Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA, 2002.
- [33] S. Mcilraith, T. Son, *Adapting Golog for Composition of Semantic Web Services*, McGrawHill, 2002.
- [34] H. Levesque, R. Reiter, *GOLOG: A Logic Programming Language for Dynamic Domains*, Journal of Logic Programming, Vol. 31 No. 1-3, pp. 5984, April-June, 1997.
- [35] J. McCarthy, P. Hayes, *Some Philosophical Problems from the Standpoint of Artificial Intelligence*, Machine Intelligence, Vol. 4, 1997.
- [36] M. Sheshagiri, M. Desjardins and T. Finin, *A Planner for Composing Services Described in DAML-S*, AAMAS Workshop on Web Services and Agent-Based Engineering, 2003.
- [37] N. Nilson, *Intelligence Artificial: Una Nueva Sintesis*, McGrawHill, Madrid, 2002.
- [38] E. J. Friedman-Hill, *Jess, The Expert System Shell for the Java Platform*, 2002.
- [39] J. Peer, *A PDDL Based Tool for Automatic Web Services Composition*, Springer Verlag., 2004.
- [40] D. McDermott, *Estimated-regression Planning for Interactions with Web Services*, in Proceedings of the 6th International Conference on AI Planning and Scheduling, Toulouse, France, AAAI Press, 2002.
- [41] J. Rao, *Semantic Web Service Composition Via Logic-Based Program Synthesis*, Ph.D. Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, 2003.
- [42] M. Matskin, O. Kirkeluten, *Infrastructure for Agents, Muddlti-Agents, and Scalable Multi-Agent Systems. Support in Multi-Agent Systems*, pp. 28-40, Springer Verlag, 2001.
- [43] U. Kuter, E. Sirin, *Information Gathering During Planning for Web Services Composition*, American Association for Artificial Intelligence, 2004.

[44] K. Erol, J. Hendler, *Semantics for Hierarchical Task Network Planning*, 1994.

[45] J. Hendler, D. Wu, E. Sirin, *Automatic Web Services Composition Using Shop2*, in Proceedings of The Second International Semantic Web Conference (ISWC), 2003.

[46] G. Schreiber, H. Akkermans, A. Anjewierden, *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, 1999.

[47] O. Sapena, *Planificación Independiente del Dominio en Entornos Dinámicos de Tiempo Restringido*, Tesis Doctoral de Informática, Universidad Politécnica de Valencia, Julio, 2005.

[48] E. Sirin, B. Parsia and J. Hendler, *Filtering and Selecting Semantic Web Services with Interactive Composition Techniques*, IEEE Intelligent Systems, Vol. 19 No. 4, pp. 42-49, 2004.

[49] F. Belfemine, A. Poggi, G. Rimassa, *JADEA FIPA-compliant agent framework*, Italy, 1999.

[50] S. Wang, *JAVA Con programación orientada a objetos y aplicaciones en la WWW*, International Thomson Editores, 2000.

[51] IPC Internacional Planning Competition. <http://www.cs.toronto.edu/aips2004>, 2000.