

Composición de servicios: una aplicación de la *web* semántica y las técnicas de planificación automática

Web service composition: a semantic web and automated planning technique application

Jaime Alberto Guzmán Luna¹ y Demetrio Arturo Ovalle Carranza²

RESUMEN

Este artículo propone la aplicación de técnicas de la web semántica y de planificación de la inteligencia artificial en un modelo de composición de servicios web que enfrenta los problemas de la ambigüedad en la descripción de los servicios y el manejo de la información incompleta en el contexto de la web. Para tratar los problemas mencionados el modelo permite el uso de servicios OWL-S e implementa una técnica de planificación que maneja una semántica del mundo abierto en su proceso de razonamiento. Como resultado de este trabajo, se obtuvo un sistema de composición de servicios web que incorpora un módulo que interpreta servicios OWL-S y los convierte a un problema de planificación en PDDL, un módulo de planificación que maneja el problema del conocimiento incompleto y un módulo de ejecución de servicios que interactúa de manera concurrente con el planificador para ejecutar cada uno de los servicios del plan de composición.

Palabras clave: web semántica, planificación en inteligencia artificial, composición de servicios web semánticos, información incompleta.

ABSTRACT

This article proposes applying semantic web and artificial intelligence planning techniques to a web services composition model dealing with problems of ambiguity in web service description and handling incomplete web information. The model uses an OWL-S services and implements a planning technique which handles open world semantics in its reasoning process to resolve these problems. This resulted in a web services composition system incorporating a module for interpreting OWL-S services and converting them into a planning problem in PDDL (a planning module handling incomplete information) and an execution service module concurrently interacting with the planner for executing each composition plan service.

Keywords: semantic web, web service, intelligence artificial planning, semantic web composition services and incomplete information.

Recibido: marzo 13 de 2008

Aceptado: octubre 24 de 2008

Introducción

Hoy en día, la web nos permite acceder a sistemas remotos y ejecutar los servicios que estos sistemas tienen para ofrecer, utilizando para ello las tecnologías que soportan los servicios web (W3C, 2002). Sin embargo, esta prometedora tecnología se enfrenta con un inevitable cuello de botella: componer servicios complejos a partir de servicios sencillos, lo cual no es una tarea trivial.

Un problema que se debe enfrentar es que los estándares de los servicios web apoyados en el XML (XML, 2001), tales como el SOAP (Simple Object Access Protocol (Gudin *et al.*, 2003) y WSDL (Web Service Description Language (WSDL, 2005)), presentan un vacío de semánticas explícitas para que dos descripciones XML idénticas puedan significar cosas totalmente distintas, dependiendo del contexto en el cual son utilizadas. Esto limita la capacidad de identificar los servicios más adecuados para llevar a cabo la tarea de composición. Tal limitante en el caso particular de los servicios web, ha llevado a considerar integrar semánticas en ellos, lo cual ha redundado en el desarrollo de los servicios web semánticos (SWS), cuya descripción interna y externa está en un lenguaje que tiene semánticas bien definidas, interpretables por las máquinas (Sycara *et al.*, 2003).

Existen en la literatura una serie de propuestas de estos lenguajes, entre las que se destacan: OWL-S (OWL, 2004), WSMO (Lara *et al.*, 2004) y WSDL-S (WSDL, 2005). En la Tabla 1 se resumen las principales características de los mismos.

En los últimos años la composición de SWS ha sido ampliamente estudiada y numerosas técnicas han sido propuestas en búsqueda de una cada vez más adecuada solución. Estas técnicas van desde la planificación en inteligencia artificial (IA) hasta la síntesis de programas, cálculo de situaciones y máquinas de estado finitas por nombrar algunas. Aunque no se puede indicar cuál de estas técnicas prevalece sobre las otras, el trabajo sobre la planificación en inteligencia artificial (IA) parece ser lo bastante conveniente desde que los servicios web han sido considerados como sinónimos de las acciones en planificación.

¹ Ingeniero civil, Universidad Nacional de Colombia, Medellín, Colombia. M.Sc., en Ingeniería de Sistemas, Universidad Nacional de Colombia, Medellín, Estudios de doctorado en Ingeniería, énfasis Sistemas e Informática, Universidad Nacional de Colombia, Medellín. Profesor asistente, Universidad Nacional de Colombia, Medellín. jaguzman@unal.edu.co

² Ingeniero de sistemas y computación, Universidad de Los Andes, Bogotá, Colombia. Master, en Informatique, l'Institut National Polytechnique de Grenoble, Francia. Docteur, en Informatique, Université Joseph Fourier, Grenoble, Francia. Profesor asociado, Universidad Nacional de Colombia, Medellín. dovalle@unal.edu.co

Tabla 1. Comparación de los lenguajes para la especificación de SWS

PROPIEDADES	OWL-S [Ankolekar et al., 2002]	WS-BPEL [Andrews et al., 2004]	WSMO [Lara et al., 2004]	WSDL-S [WSDL, 2006]
TIPO DE SERVICIO QUE PERMITE DESCIBIR	Atómico (Service profile) Compuesto (Service Model)	Compuesto	Atómico (capability) Compuesto (interface)	Atómico
INTEGRACIONES SEMÁNTICAS	Sí	No	Sí	Sí
SOPORTE PARA DATOS	Sí	Sí (Doc XML)	Sí	Sí

En general, un problema de planificación es un problema de búsqueda que requiere encontrar una secuencia eficiente de acciones que conducen a un sistema desde un estado inicial hasta un estado objetivo. Desde el punto de vista de agentes inteligentes, el campo de la planificación busca construir algoritmos de control que permitan a un agente sintetizar una secuencia de acciones que le lleve a alcanzar sus objetivos (Weld, 1994). Más formalmente, un problema de planificación puede ser descrito como una tupla (S, S_0, G, A, R) , donde S es el conjunto de todos los posibles estados del mundo, $S_0 \subset S$ denota el estado inicial del mundo, $G \subset S$ señala el estado objetivo del mundo en el que el sistema de planificación intentará buscar, A es el conjunto de acciones que el planificador puede llevar a cabo para cambiar de un estado a otro en el mundo, y la relación de traslación $R \subseteq S \times A \times S$ define las precondiciones y efectos para la ejecución de cada acción (Ghallab et al., 2004).

La definición de un problema de planificación IA es comparable con el problema de composición de servicios web, en donde S_0 y G son el estado inicial y el estado objetivo especificados en los requerimientos del solicitante del servicio web, A es un conjunto de servicios disponibles y R indica la función de cambio del estado al ejecutar cada servicio. Como resultado de la anterior semejanza, un número de algoritmos de planificación se han utilizado para tratar el problema de la composición de servicios web. En la literatura bajo el enfoque de la planificación existen varios trabajos para llevar a cabo la composición de SWS, donde vale la pena destacar dos propuestas.

Una primera propuesta es (Mithun, 2004) es aquella donde se utiliza una técnica de planificación como búsqueda. Específicamente, en este trabajo se describe como el planificador STRIPS, mediante un algoritmo regresivo que fue usado para manejar la composición e invocación de servicios. La arquitectura incluye un planificador y una máquina de invocación de servicios. En este el conocimiento de los servicios expresados en OWL-S como operadores atómicos es extraído para conformar los operadores del planificador STRIPS con sus atributos de entradas, salidas, precondiciones y efectos. El planificador STRIPS es implementado mediante la herramienta para sistemas expertos JESS (JESS, 2008). En él, si todos los objetivos son cumplidos, el plan resultante es pasado a un mecanismo de invocación que toma cada operador a manera de un encadenamiento hacia adelante ejecutando cada servicios web.

Una segunda propuesta es (Klusch et al., 2005) aquella donde se aplica una combinación de la planificación heurística hacia adelante con la planificación HTN en un planificador llamado XPlan para llevar a cabo la composición de servicios web. OWLS-XPlan convierte los servicios OWL-S a descripciones equivalentes de dominio y problema en el lenguaje PDDL (Gudin et al., 2003; PDDL, 2003) y lo invoca en el planificador XPlan para generar una secuencia de un plan de composición de servicios que satisface un objetivo dado.

El problema de las anteriores propuestas es que se basan en el supuesto de que su planificador conoce todos los objetos y sus relaciones, por lo cual usan el razonamiento del mundo cerrado (Golden et al., 1994). Esta suposición indica que si un hecho no está contenido en su base de conocimiento local, entonces este simplemente se asume como falso. Sin embargo, esa afirmación no es realista en el contexto de la web porque no se puede esperar que se tenga toda la información relevante en la base de conocimiento local del planificador, ya que la web es un mundo abierto y en constante movimiento. Por lo tanto, el planificador debe ser capaz de considerar que desconoce determinada información sobre el mundo. Como se puede observar, la composición de servicios web bajo las técnicas de planificación no es una tarea fácil, ya que es necesario tener en cuenta la ambigüedad en la descripción de los servicios y el manejo de la información incompleta en el contexto de la web. Es por ello que en este trabajo se presenta una solución al problema de la composición de servicios web la cual tiene como objetivo atacar los anteriores problemas bajo el enfoque de los SWS, haciendo uso de una técnica de planificación IA que maneja una semántica del mundo abierto en su proceso de razonamiento, donde no es posible asumir como falsa una afirmación por el hecho de no estar en su base de conocimiento. Aunque el trabajo presenta la limitación de no manejar servicios web compuestos, es el más acorde con la realidad de la web cuando de composiciones atómicas de servicios se trata.

A fin de detallar la propuesta, el artículo se organiza de la siguiente manera: primero se presenta el modelo de composición de servicios propuesto, que combina la especificación semántica de los servicios con la técnica de planificación; luego se describe cómo se manejan los SWS en el modelo y su conversión en un problema de planificación y se define el proceso de planificación de la composición de SWS teniendo en cuenta el problema de la información incompleta de la web. Finalmente, se recopilan las conclusiones.

Un modelo de planificación para la composición de servicios web

Como una alternativa de solución a la composición de SW se propone un modelo que aborda eficientemente problemas tales como la ambigüedad en la descripción de los servicios y el manejo de la información incompleta en el contexto de la web. Tal modelo incluye un conversor de especificaciones semántico que además permite el manejo del comportamiento de los respectivos parámetros de entrada y salida de un SWS, usando el lenguaje de descripción OWL-S, el cual debe traducir a su respectiva representación en PDDL (PDDL es el lenguaje de modelado entendible por la mayoría de planificadores en inteligencia artificial), mientras que el proceso de composición de SWS se realiza mediante una técnica de planificación IA cuyo algoritmo usa una semántica del mundo abierto en su proceso de razonamiento, además de considerar la ejecución concurrente con el proceso de planificación.

La arquitectura propuesta está compuesta por tres módulos principales, como se detalla en la Figura 1: un convertidor de especificaciones OWL-S a PDDL, un planificador que genera un plan de composición SWS y un ejecutor que inicializa el problema de planificación y ejecuta de manera concurrente, una a una, las acciones planteadas por el planificador.

El comportamiento funcional que se propone consiste básicamente en dos fases: i) la conversión de especificaciones de un ambiente de ontologías *web* a un ambiente de planificación IA, y ii) la planificación y ejecución de la composición del servicio *web*.

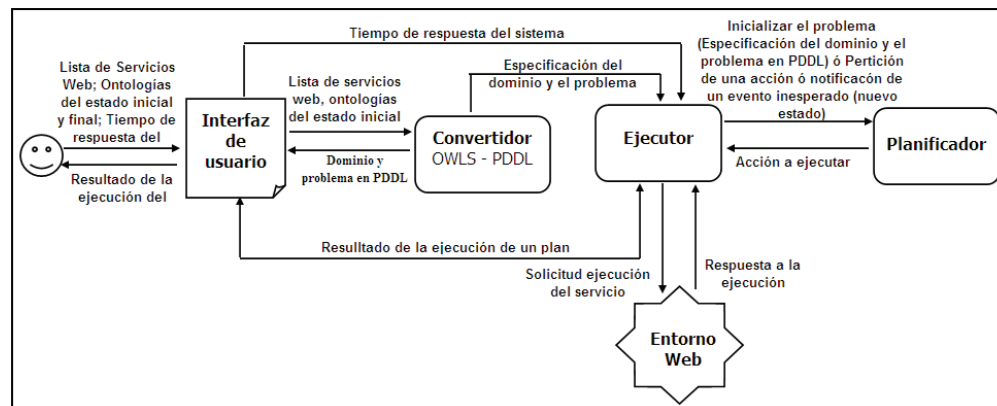


Figura 1. Arquitectura propuesta

En la primera fase el convertidor de especificaciones traduce las ontologías del dominio y de la consulta, junto con las descripciones de los servicios (implementadas respectivamente en OWL y OWL-S), en una especificación del dominio PDDL y su respectivo problema de planificación (el estado inicial de la *web* y los objetivos a alcanzar). En la segunda fase el ejecutor inicia el proceso de planificación enviándole las especificaciones PDDL del dominio y su respectivo problema al planificador, con lo cual se inicia una interacción entre el sistema de planificación y ejecución para el envío de acciones y notificaciones de eventos inesperados.

Manejando los servicios web semánticos

Para efectos de este trabajo, OWL-S ha sido considerado como el lenguaje de descripción de los servicios. La coalición OWL-S define una ontología y un lenguaje para los SWS llamado OWL-S. La ontología permite la descripción de servicios *web* semánticos en términos de conceptos y relaciones complejas entre ellos, incluyendo no sólo clases, relaciones de subclases, restricciones de cardinalidad, etc., sino que incluye también toda la información de los tipos de XML. La conversión de las descripciones de los servicios OWL-S para conformar el dominio en el PDDL requiere inicialmente de la transcripción de las clases (*Class*) y propiedades (*Property*) incluidas en las ontologías OWL del estado inicial y final a los tipos (*Type*) y predicados (*Predicate*) PDDL, respectivamente. Luego, se realiza el mapeo de los servicios a las acciones PDDL.

Cualquier parámetro del servicio del tipo *hasPrecondition* puede directamente ser trasladado a la precondition de la acción usando Predicados. Lo mismo pasa con el parámetro de la condición *hasEffect*. En el caso de la conversión PDDL de la salida de un servicio individual OWL-S, que es la información del servicio ofrecido al mundo, esta es una tarea no tan directa de realizar. El problema es que la condición *hasEffect* del servicio describe explícitamente como el estado del mundo cambiará, mientras que para un valor del parámetro *hasOutput* este no sería su caso, ya que está más asociado a cómo cambia el conocimiento del agente de planificación. Sin embargo, PDDL no permite describir el conocimiento no físico (el conocimiento del agente), por ejemplo, las conexiones de un tren producidas como la salida de un servicio a través de un *hasOutput*. Para resolver este problema se puede mapear el parámetro *X* de la salida del servicio, a un tipo especial del parámetro *hasEffect*. En particular, la variable *X* del parámetro de salida *hasOutput* es descrita y adicionada al estado del mundo actual del

planificador por medio de un nuevo predicado en PDDL llamado *agentHasKnowledgeAbout(X)*. Similarmente, cada variable de entrada *Y* de un servicio *web*, *hasInputY*, es mapeada a un parámetro de entrada *Y* de una acción PDDL complementada por el predicado de precondition *gentHasKnowledgeAbout(Y)*.

Bajo esta solución, la descripción de un servicio *web* en esta representación de una acción de planificación, sólo se podrá utilizar si el predicado de la correspondiente precondition adicional, "*agentHasKnowledgeAbout(Y)*", es satisfecha. Sin embargo, tal solución puede fallar durante la ejecución del servicio ya que el chequear las condiciones del servicio puede revelar que ellas no son satisfe-

chas en el estado actual del mundo. Tal evento debe ser considerado por el algoritmo de planificación.

Planificando la composición de SWS bajo información incompleta

Para manejar la información incompleta sobre el estado actual de la *web* se propone una extensión definida como lógica trivaluada (verdadero, falso y desconocido), la cual, contrario a muchos planificadores, permite dudar del valor de verdad de un literal que no se haya incluido explícitamente en la descripción de un estado. Tal literal es descrito como desconocido; así, los niveles de literales se denominarán niveles proposicionales, ya que estos contendrán proposiciones lógicas (no literales). Igualmente, el primer nivel L_0 , tendrá todas las proposiciones lógicas que se satisfacen en S_0 . Los niveles de acciones A_i contienen todas las acciones cuyas condiciones positivas y negativas se encuentran en el nivel L_i ; el siguiente nivel, L_{i+1} , extiende L_i con los efectos positivos y negativos de las acciones A_i . Por lo anterior, un nivel proposicional puede contener dos proposiciones que representen un mismo literal, uno en forma positiva y otro en forma negativa.

Adicionalmente, el planificador cuenta con un tipo de acción asociado al manejo de dominios de información incompleta; estas son conocidas como "acciones de censado", las cuales son similares a las acciones normales, difiriendo en sus efectos, las que corresponden a un conjunto de literales (no de proposiciones lógicas) cuyo valor de verdad será obtenido en la ejecución, para este caso se asume también que los efectos de la acción se cumplen en el mundo. Si alguna de dichas asunciones resulta ser falsa, el mecanismo de monitorización del ejecutor deberá notificarle este hecho al planificador, el cual se encargará de calcular otros planes para el nuevo estado.

El algoritmo de planificación se basa en la idea de descomponer el problema de planificación original *PP*, en *n* subproblemas independientes. En el que $PP = \langle F, L, A, S_0, G, m \rangle$, donde *F* es un conjunto de variables numéricas; *L* es un conjunto de literales o hechos; *A* es un conjunto de acciones posibles, S_0 es el estado inicial del problema, *G* es el conjunto de objetivos del problema y *m* es el criterio de optimización (métrica del problema). De manera que se subdivide en $PP_i = \langle F, L, A, S_0, G_i, m \rangle, i=1...n$, donde *n* corresponde al número de objetivos del problema ($n=G/G_i$). Se genera, por lo tanto, un plan P_i , distinto para cada uno de los subproblemas. Finalmente, se estudian los conflictos existentes entre

- Erfurt, Germany, September 27-30, Proceedings, Lecture Notes in Computer Science, pp 254–269.
- Mithun, S., Automatic Composition and Invocation of Semantic Web Services., Tesis de Maestría, University of Maryland, August 2004.
- OWL Service Coalition., OWL-S: Semantic markup for web Services, 2004. OWL-S 1.1, Disponible en línea: <http://www.daml.org/services/owl-s/1.1/>. Última visita octubre 2008.
- PDDL: Planning Domain Definition Language., Disponible en línea: <http://www.ida.liu.se/~TDDA13/labbar/planning/2003/writing.html>, Última visita diciembre 2008.
- Pistore, M., Traverso, P., Bertoli, P., Planning and Monitoring Web Service Composition., presented en memorias AIMS 2004, Varna, Bulgaria, 2004.
- Pistore M., Traverso P, and Bertoli P., Automated Composition of Web Services by Planning in Asynchronous Domains, Memorias Proceedings of the 15th International Conference on Automated Planning and Scheduling, 2005, pp. 2-11.
- Russell, S. y Norvig, P., Artificial Intelligence: A Modern Approach., 2nd. Edition, Upper Saddle River, (ed) Prentice Hall, 2003
- Sycara, K., Pupriya Ankolekar, M., Naveen, S., Automated discovery, interaction and composition of semantic Web services, J. Web Sem, 1(1), 2003, pp. 24-46.
- UDDI V 3.0 Ratified as OASIS Standard., Disponible en Línea: <http://www.uddi.org>. Última visita julio 2008.
- XML., Disponible en Línea: <http://www.xmlnews.org>, fecha de publicación 2001. Última visita diciembre de 2008.
- Weld D., An Introduction to Least Commitment Planning., AI Magazine, No. 4, Winter 1994, pp 27-61.
- WSDL., Web Service Semantics, Disponible en Línea: <http://lstdis.cs.uga.edu/library/download/WSDL-S-V1.pdf>. Publicado abril 2005. Última visita octubre 2008.
- Wu, D., Sirin, E., Hendler, J., Nau, D., Parsia, B., Automatic Web services composition using SHOP2., Memorias Workshop on Planning for Web Services, Trento, Italia, Junio, 2003
- W3C., Web Service Architecture Requirements., Disponible en Línea: <http://www.w3.org/TR/wsa-reqs/>, 2002.