

**AN ANALYSIS OF THE STATE-OF-THE-ART MANAGEMENT OF RISK FAILURE
DURING PARTIALLY SPECIFIED WEB SERVICES' COMPOSITION**

**UN ANALISIS AL ESTADO DEL ARTE EN EL MANEJO DE RIESGOS DE FALLA
DURANTE LA COMPOSICION DE SERVICIOS WEB PARCIALMENTE
ESPECIFICADOS**

Ing. Byron Enrique Portilla Rosero, MSc. Jaime Alberto Guzmán Luna

Escuela de Sistemas, Universidad Nacional de Colombia – Sede Medellín
SINTELWEB: Grupo de Investigación “Sistemas Inteligentes Web”
Calle 59A No 63 – 20, Medellín, Colombia
E-mail: beportillar@unalmed.edu.co, jaguzman@unal.edu.co

Abstract: In this paper we introduce a state-of-the-art review and show a solution for the planning of the Web services' composition, using specified Web services without the services behaviors' information presenting their possible failure factor during the composition process. The solution that we propose is developing a new technical composition composed by cost-based planning and machine learning.

Resumen: Este artículo se presenta una revisión del estado del arte y una discusión donde se plantea una solución frente al tema relacionado con la planificación de la composición de servicios Web, utilizando especificaciones parciales de servicios Web, donde no se tiene la información pertinente al comportamiento de los servicios en cuanto a su factor de riesgo de fallas que puedan presentar frente a este proceso de composición. La solución que se propone plantea el desarrollo de una nueva técnica de planificación que integre técnicas de planificación basada en costos y el aprendizaje de máquinas.

Keywords: Semantic Web services, services composition, management failure risk, planning in artificial intelligence, inductive logic programming.

1. INTRODUCCIÓN

Los trabajos realizados hasta el momento en cuanto a la composición de servicios Web, se caracterizan en su mayoría por la utilización de modelos, donde las especificaciones de los servicios Web utilizadas para generar un plan de composición, están totalmente definidas. Es así como mecanismos orientados a la composición que busquen minimizar el riesgo de fallas además de tener la información funcional de cada servicio, tienen la información

pertinente al factor de riesgo de que tales servicios fallen, esto con el fin de guiar al mecanismo de composición para seleccionar la mejor solución entre todas las posibles composiciones que minimice la posibilidad a que dicha composición falle.

Desafortunadamente esto en la realidad no se cumple y el problema principal de este proceso de composición, es obtener estas especificaciones. Es por esto que en este documento se propone una

solución al manejo de riesgos de fallas en la composición de servicios Web parcialmente especificados. Para tal fin este documento está organizado de la siguiente manera: en la sección 2, se presenta una descripción y los trabajos realizados en la composición de servicios Web y planificación en IA; en la sección 3 se describe la composición de servicios Web y el manejo de la calidad definiendo los conceptos principales y los trabajos realizados en este campo; en la sección 4, se describe la planificación en IA y el aprendizaje de máquinas y se definen las diferentes investigaciones relacionadas con este tema; en la sección 5, se presentan los trabajos realizados que integran la planificación y el aprendizaje de máquinas para la composición de servicios; en la sección 6, se realiza una discusión sobre los trabajos analizados en las secciones anteriores y se plantean algunas preguntas de aún por resolver asociadas con este tema y se plantea como se permitiría las preguntas planteadas.

2. COMPOSICIÓN DE SERVICIOS WEB Y LA PLANIFICACIÓN EN IA

Los servicios Web, son un conjunto de aplicaciones modulares, autónomas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web (Barreiro *et al.*, 2005) y que por medio del intercambio de sus datos, proporcionan un conjunto de servicios. Los servicios son solicitados por los usuarios a través de peticiones realizadas a los proveedores de servicios y, estos a su vez generan procedimientos remotos, que proveen un conjunto de información dinámica a los usuarios, a través de la Web.

Un aspecto importante asociado a los servicios Web es el tema de la composición de servicios Web, el cual está relacionado con el proceso de desarrollar servicios personalizados a través de otros ya existentes por medio de procesos de descubrimiento dinámico, integración y ejecución con el fin de satisfacer al usuario (Chakraborty *et al.*, 2002). La composición de servicios Web (WSC), es un aspecto básico de la computación orientada a los servicios (Quian *et al.*, 2005), que soporta procesos de negocios complejos y utiliza modelos que le permiten la composición flexible de sistemas heterogéneos a través de protocolos básicos de los servicios Web (SOAP, WSDL, UDDI).

Para soportar la composición de servicios, se han propuesto los servicios Web Semánticos (SWS) (Web services HP, 2007), los cuales son servicios Web cuyas descripciones internas y externas están en un lenguaje que tiene semánticas bien definidas interpretables por las máquinas (McIlraith *et al.*, 2001). Algunos de estos lenguajes son el OWLS (OWL-S, 2007), WSMO (WSMO, 2007) ó WSDL-S (WSDL-S, 2007).

Por otro lado esta lo que se conoce como planificar que es el razonamiento sobre la acción. Este es un proceso de deliberación explícita y abstracta que escoge y organiza las acciones anticipándose a los resultados esperados (Nau *et al.*, 2004). Esta deliberación busca alcanzar de la mejor manera que sea posible algunos objetivos preestablecidos. La planificación automática es un área de la Inteligencia Artificial (IA) que estudia este proceso de deliberación de manera computacional. Aunque existen en la literatura diferentes técnicas de planificación IA, se destacan dos técnicas que están íntimamente unidas; la planificación heurística y la planificación basada en costos. La planificación heurística se basa en la utilización de funciones de estimación que exploran el espacio de búsqueda y se pueden clasificar en admisible y no admisible (Nilsson, 1994). La primera permite alcanzar soluciones óptimas y la segunda alcanzar soluciones donde no se garantiza la optimización de los resultados, aunque algunos investigadores han encontrado que disminuye notablemente el tiempo de cómputo (Haslum *et al.*, 2000). La planificación basada en costos consiste en la búsqueda del costo óptimo de un plan generado durante el proceso de planificación. Ésta evalúa los costos de las acciones que se ejecutan durante un plan relajado y, realiza una serie de funciones para propagar el plan. Las funciones de costo, se propagan durante todo el proceso de planificación. Estas, capturan la diferencia entre el costo accesible y los índices de la capa de proposiciones para propagar el mejor costo, y alcanzar la siguiente preposición cada nivel (Bryce, 2007 y Jörg, 2003).

En los últimos años en asocio con los SWS, la composición de servicios ha venido siendo investigada y un número de técnicas que van desde las técnicas de planificación en Inteligencia Artificial (IA) hasta la síntesis de programas, pasando por el cálculo de situaciones, máquinas de estados finito por nombrar algunas han sido empleadas. Aunque no se puede indicar cuál de estas soluciones prevalece sobre las otras, los

trabajos basados en las técnicas de planificación IA han mostrado ser lo suficientemente adecuados desde que los servicios Web han sido considerados en (Mithun, 2004 y Evren, 2004), como sinónimos de las acciones en planificación. A continuación se relacionan los principales trabajos que emplean este tipo de técnicas.

McDermont utiliza (*OPTOP*) un planificador de regresión estimada (Drew, 2004) es decir un planificador de espacios de estado, que es guiado a través de medidas heurísticas para determinar que tan cerca está una situación para satisfacer un objetivo, y que tan buena es, de acuerdo a la función objetivo con el fin de demostrar su efectividad en cuanto a la composición de servicios Web (Chan *et al.*, 2007). Este planificar se enfoca mas sobre el razonamiento profundo de situaciones y transiciones que sobre una ejecución inexperta. Además, la versatilidad de *Optop* está estrechamente vinculada a la descripción completa de las situaciones. Según (McDermott, 1999) *Optop* gasta mucho tiempo en los estados de búsqueda que otros planificadores. Sin embargo, en algunos dominios explora tan pocos estados que el tiempo de ejecución se compara con sistemas optimizados altamente.

(Evren *et al.*, 2004), han propuesto planificadores para la composición automática de servicios Web por ejemplo, *SHOP2* un sistema de planificación de dominio independiente que utiliza la composición de tareas parcialmente ordenadas y para resolver algunos problemas de planificación en algunos dominios, lo hace a través de métodos que hacen posible descomponerlas en subtareas para alcanzar unas metas. Estos métodos se cumplen siempre y cuando se cumplan las condiciones previas, la descomposición es recursiva hasta que puedan ser realizadas por los planificadores, construye planes hacia adelante del estado inicial, y las realiza dependiendo de cómo cada una se vaya ejecutando; así genera los pasos de cada plan en el mismo orden que estos serán ejecutados más tarde; por lo tanto, se conoce el estado actual en cada paso del proceso de planificación; esto reduce la complejidad de razonamiento para eliminar una gran cantidad de incertidumbre acerca del mundo.

De igual forma, (Paolucci *et al.*, 1999) utiliza *RETSINA*. Este planificador es similar a *SHOP2* en la forma como intercala la planificación y la ejecución. Este sistema de planificación multiagente provee a cada agente con un componente de

planificación interno. Cada agente, formula y ejecuta planes detallados para lograr objetivos a nivel local y global. El conocimiento del dominio se distribuye entre los agentes, por lo que cada agente sólo tiene un conocimiento parcial de la situación del mundo. Además, el dominio cambia dinámicamente, por lo tanto, los conocimientos disponibles podrían convertirse en obsoletos (Paolucci *et al.*, 1999).

(McIlraith *et al.*, 2001) amplía y adapta un lenguaje de programación basado en cálculos de situaciones (Golog) para la composición automática de servicios Web. Utilizan métodos formales para definir, caracterizar y computar las condiciones y los efectos condicionales para una acción compleja. Además, permite una ejecución concurrente, los problemas de planificación, formulados como parte de un programa de Golog, son resueltos por un moderno planificador durante la ejecución del programa. Tiene sus ventajas en cuanto un usuario puede escribir programas que limitan la búsqueda de un plan ejecutable en una manera flexible. Sin embargo, cuando es necesaria la planificación general los soportes de Golog están solos (Jens *et al.*, 2007). De igual forma Congolod. Es un lenguaje de programación lógica que se utiliza para guiar al planificador en la producción de planes certeros que respetan la imposición de restricciones para los programas.

Investigaciones realizadas por (Prashant *et al.*, 2005 y Aiqiang *et al.*, 2005) utilizan procesos de decisión de Markov en la composición de servicios Web. Doshi presentó un enfoque basado en políticas para la composición dinámica de servicios Web, que resultan en flujos de trabajo. Desarrollaron una herramienta para convertir una política en un flujo de trabajo, en especificaciones BPEL4WS utilizando BPWS4J API.

Como describen (Michele *et al.*, 2006) ASTRO es un proyecto de investigación que consta de una serie de herramientas que amplían plataformas existentes para el diseño y ejecución de servicios Web, con composición automática y ejecución que monitorea funcionalidades así como aplicaciones orientadas a los servicios enfocados en la integración de procesos de negocios que son distribuidos entre entidades distintas. ASTRO consta de varios componentes: *WS-gen*, *WS-mon*, *WS-verify*, *WS-console* y *WS-animator*. *WS-gen* es el encargado de generar la composición automática. Sin embargo, aunque las aplicaciones asociadas a

estos trabajos funcionan adecuadamente, se ha identificado que muchas de estas no son capaces de enfrentar algunos problemas relacionados con el contexto de la Web.

3. COMPOSICIÓN DE SERVICIOS WEB Y MANEJO DE CALIDAD

Cuando un servicio Web no puede alcanzar la solución para el requerimiento del cliente, existe la posibilidad de integrar diferentes servicios de diferentes proveedores, con el objeto de satisfacer el requerimiento del usuario; estos nuevos sistemas Web deben ser receptivos, robustos y estar siempre disponibles. Además de soportar demandas de concurrencia, estos requerimientos están enfocados a la calidad de los servicios QoS (Jörg, 2003). Un servicio Web Compuesto, reacciona a cualquier cambio en algunos comportamientos de servicios constitutivos. Los sistemas orientados a servicios (SO) están sujetos a riesgos causados por la vulnerabilidad arquitectónica, como: problemas técnicos, niveles de seguridad y por los intereses que están en conflicto de los servicios involucrados. Para tener un control sobre los riesgos, aparece el manejo de riesgos que proporciona una forma de recopilar los datos precisos, para realizar un buen juicio técnico o de negocio (Jörg, 2003).

Un aspecto de suma importancia es el manejo de fallas en las composiciones de servicios Web. La composición de servicio es un proceso complicado que involucra un análisis cuidadoso de los requisitos del servicio a componer, la semántica y el comportamiento de los servicios existentes, las pruebas del servicio compuesto y su adaptación.

A pesar de todos los esfuerzos bajo el enfoque de la planificación IA, los problemas tanto a nivel técnico como de comportamiento pueden aparecer. Por ejemplo, las modificaciones de los servicios involucrados y sus fallas inesperadas pueden afectar una aplicación del cliente. Los servicios fallidos pueden ser sustituidos con servicios análogos que permitan corregir el comportamiento de las composiciones en tales situaciones. Ya que se requiere de mucho trabajo para introducir de manera segura un nuevo componente en un sistema, es necesario conocer de antemano diferentes servicios alternativos a tales componentes.

Debido a la inestabilidad del entorno de la Web, los sistemas basados en servicios requieren un monitoreo constante de los mismos en tiempo de ejecución. Las estadísticas sobre la experiencia de los usuarios con los servicios Web, se puede utilizar para seleccionar los servicios de buen comportamiento. La calidad del servicio (QoS) es un conjunto de parámetros tales como: el costo de la ejecución del servicio, su rendimiento, fiabilidad, robustez y similares. Es así como para la composición de servicios, el análisis de la QoS de los servicios Web es de suma importancia.

Múltiples propuestas para el manejo de la QoS destinadas a la evaluación y selección de los mejores servicios, han aparecido a causa de las múltiples dimensiones y la volatilidad de los parámetros de calidad de servicio. Varios trabajos que buscan enfrentar el problema de la composición, la calidad de los servicios a intervenir; con el fin de obtener el servicio compuesto con la mejor calidad dependiendo de los parámetros asociados. En estos trabajos se plantea el problema de la composición bajo el supuesto de que los parámetros de calidad de los servicios y sus valores de ya, son conocidos. Los principales trabajos relacionados con este aspecto se detallan a continuación.

(Ran, 2003) plantea un nuevo modelo de descubrimiento de servicios Web a través de sus parámetros básicos no funcionales es decir, la calidad de los servicios. Entre estos parámetros se encuentra: Rendimiento de procesamiento y transferencia, capacidad, latencia (tiempo de envío y respuesta), tiempo de reacción, disponibilidad, confiabilidad, reputación, coste de la ejecución. La importancia de este modelo es la satisfacción y confiabilidad de los consumidores sobre la calidad del servicio que sea descubierto. Por lo tanto este trabajo propone un nuevo modelo de servicio, donde la calidad del servicio se toma como limitaciones en la búsqueda de servicios Web. Esto se supone daría confianza a los consumidores de servicios Web acerca de la calidad del servicio que están a punto de invocar.

(Sherchan *et al.*, 2005), establece que es factible deducir de las exigencias y las preferencias de los usuarios, la selección de los servicios Web, tomando solo una parte del historial de la evaluación en el funcionamiento de los servicios. También hacen referencia a la importancia de las medidas recientes de los servicios. Asimismo tiene

en cuenta los parámetros de calidad de servicio, los cuales dependen de múltiples factores que deberían ser tomados en cuenta, en tempranas etapas del desarrollo.

(Menasce *et al.*, 2002) proporciona una metodología para planificar la capacidad del servicio. Para el análisis es esencial, el conocimiento sobre el número de usuarios potenciales, la frecuencia de la invocación del servicio y las distribuciones de tiempos. Plantea una serie de 8 pasos para ello: i. comprender la arquitectura del servicio, ii. Caracterizar la carga de trabajo, iii. Obtener parámetros de los modelos de trabajo, iv. Previsiones de evolución del volumen de trabajo, v. Desarrollar modelo de actuación, vi. Calibrar y validar modelos, vii. Predecir el rendimiento de los servicios y viii. Análisis de costo-desempeño.

(Cardoso *et al.*, 2004), describen el modelo que permite predecir la calidad de los servicios para flujos de trabajo, basados en atributos de servicios atómicos en la calidad de servicios. Uno puede compensar la deficiencia de la composición, si existen muchos servicios con funcionalidad compatible. La calidad de los servicios en los flujos de trabajo, representan las características cuantitativas y cualitativas de una aplicación de flujo de trabajo necesario para lograr un conjunto de necesidades iniciales. Los caracteres cuantitativos pueden ser evaluados en términos de medidas concretas, tales como flujo de trabajo de tiempo de ejecución, coste, entre otros y las características cualitativas especifican los servicios esperados ofrecidos por el sistema, como la seguridad y los mecanismos de tolerancia a fallos. QoS debe considerarse como un aspecto integral de los flujos de trabajo, por lo que deben integrarse con las especificaciones de flujo de trabajo.

(Zeng *et al.*, 2004), Presenta una plataforma de software que se ocupa de la selección de los servicios web, con el propósito de su composición de manera que maximice la satisfacción de los usuarios expresada como funciones de utilidad sobre atributos de QoS, satisfaciendo las restricciones establecidas por el usuario y por la estructura de la composición de servicios. Esto se aplica tanto a los servicios Web autónomos y servicios Web compuesto por otros servicios de la web (Servicios compuestos). Consideran la tarea de selección del servicio, como un problema global de optimización. La programación se aplica para

encontrar soluciones que representen la composición del servicio, que optimice la función objetivo. Al final, se define como una combinación lineal de cinco parámetros: disponibilidad, frecuencia exitosa de ejecución, tiempo de respuesta, costo de ejecución y reputación.

(Martín Díaz *et al.*, 2005), propone una solución programada de restricciones para la consecución de servicios, cuyas demandas y ofertas son temporalmente consistentes. En (Benton *et al.*, 2005), el problema es modelado como un programa lineal de enteros, mezclados donde se especifican las restricciones globales y locales.

(Yu *et al.*, 2005), modelan la selección del servicio como un problema complejo multi-objetivo. Parten del hecho, que la selección del servicio puede hacerse de forma estática o dinámica y utilizan restricciones de calidad de los servicios (tiempo, costos, servicios disponibles, servicios confiables) y a través de estos alcanzan una selección de servicios más confiable. Proponen una estructura basada en *broker (QCWS)* que utiliza dos pasos para realizar la composición de servicios Web. El primero es una composición del servicio y el segundo la selección del servicios. Estos pasos son ejecutados a través de un manejador de composición y de la selección, respectivamente.

(ChafI *et al.*, 2005), propone un mecanismo para la propagación y recuperación de fallas en la orquestación descentralizada de servicios. La arquitectura descentralizada da lugar a la complejidad adicional, que requiere la propagación de fallas entre las particiones ejecutadas independientemente.

El problema de estas propuestas es que utilizan una técnica basada en la optimización de una función objetivo mediante la simple suma de los pesos de los parámetros de QoS de los posibles servicios a intervenir, teniendo en cuenta su comportamiento individual (información provista por los proveedores de servicios en la mayoría de los casos), en vez de considerar su comportamiento en el propio servicio compuesto final como un todo. Lo anterior permitiría determinar de una manera más real el factor de riesgo que presentan los servicios elegidos en el proceso de composición, con el fin de minimizar el riesgo de falla del propio servicio compuesto resultante. Desafortunadamente, esta información no es posible obtenerla de manera a priori al propio problema de composición.

4. PLANIFICACIÓN EN IA Y APRENDIZAJE DE MÁQUINAS

Independiente al problema específico a resolver, una forma de mejorar las técnicas de planificación IA ha sido el emplear técnicas de aprendizaje de máquinas. El aprendizaje usa experiencias y preconceptos del pasado para mejorar la habilidad de actuar en el futuro. En la comunidad de la planificación IA, el aprendizaje de máquinas es visto como un poderoso mecanismo que dota a este tipo de agente de gran autonomía y flexibilidad, con el fin de compensar el conocimiento incompleto que el diseñador le brinda acerca del mundo que el agente enfrentará. Aunque existen varias técnicas de aprendizaje de máquinas, una técnica que ha sido ampliamente utilizada con éxito en ambientes de planificación, es la programación lógica inductiva (ILP). La ILP es una disciplina que investiga la construcción inductiva de teorías causales de primer orden, de conocimientos de ejemplos y generalidades (Eduardo, 2006). Esta técnica de aprendizaje es un área de investigación definida como la intersección del aprendizaje inductivo y la programación lógica (Stephen et al, 1994). De la programación lógica ILP hereda las bases teóricas profundas, y del aprendizaje de máquinas obtiene un enfoque experimental y una orientación hacia aplicaciones prácticas (Nada *et al*, 1996). Su lenguaje de representación o expresividad es esencialmente proposicional (Michele, 2006), y su objetivo apunta a un esquema formal tan bueno como los algoritmos prácticos para el aprendizaje inductivo de descripciones relacionales en forma de programas lógicos (Nada *et al*, 1996). El aprendizaje inductivo utiliza métodos con el objeto de obtener hipótesis a partir de observaciones y lograr un conocimiento a partir de la experiencia.

La planificación en IA ha sido por más de 30 años un campo activo de investigación en el cual, las técnicas de aprendizaje pueden ser empleadas para mejorar su desempeño. La siguiente revisión de los principales trabajos asociados con aplicar técnicas de aprendizaje en la planificación, para mejorar diferentes aspectos de este proceso se centra principalmente en el enfoque relacionado con la técnica de aprendizaje de máquina (ILP).

(Yoon *et al.*, 2006), Se enfoca en un aprendizaje automático para heurísticas de planes relajados. La idea es aprender una función de regresión lineal de planes entrenados, que reflejen la diferencia en la longitud de un plan relajado y el costo actual para

alcanzar el objetivo. La técnica es considerada como un método inductivo que aprende al aumentar un dominio independiente heurístico, con el fin de mejorar el rendimiento en la planificación del dominio.

(Zimmerman *et al.*, 2005), Describen un sistema que intenta aprender reglas aproximadamente correctas, relajando las restricciones del sistema de UCPOP-EBL que requiere explicaciones de las fallas retornadas de todas las ramas de un subárbol de búsqueda, antes de que se construya una regla del control de búsqueda.

(Bergmann *et al.*, 1996), Desarrollan un esquema para reutilizar casos en varios niveles de abstracción, permitiendo describirlos, analizarlos y aprenderlos. Argumentan como la abstracción puede ser usada con éxito para mejorar la adaptación y la recuperación de una tarea sintética.

(José *et al.*, 2000) Se enfocan en la utilización de un aprendizaje automático a través de las reglas de reescritura de un plan, basadas en la comparación de los planes iniciales y óptimos, con el objeto de obtener un modelo de planificación eficiente de alta calidad.

Otros como (Muhammad, 2001) Utilizan la reglas de reescritura para mejorar la ejecución de un sistema de dominio independiente. Estas reglas pueden ser utilizadas por un sistema de solución de problemas después de generar una solución inicial, para transformar esta entrada en una solución de alta calidad. Las reglas de reescritura constan de dos secuencias equivalentes de acciones de manera que una de ellas puede ser remplazada por otra.

(Daniel *et al.*, 1997), utilizan un método de aprendizaje, que combina una estrategia deductiva y una inductiva para aprender eficientemente conocimiento de control con experiencia. Utilizan un planificador llamado Hamlet, que aprende conocimiento de control para mejorar la eficiencia y la calidad de las soluciones generadas por el planificador no lineal por dygy 4.0. Para este método (Daniel *et al.*, 1997), identifica tres aspectos principales que son: explicación de éxito, refinamiento incremental del conocimiento adquirido, y aprendizaje a invalidar el comportamiento por defecto del problema resuelto.

Jiménez y otros (Sergio, 2007 y Jesús, 2007) buscan a través de la integración de técnicas de aprendizaje y procesos de planificación y ejecución desarrollar

un planificador que pueda automáticamente mejorar el conocimiento inicial del entorno, teniendo en cuenta la ejecución de las acciones para lograr planes robustos.

(Sergio *et al.*, 2006) han implementado una arquitectura que permite aprender información bajo incertidumbre como conocimiento de control. Para ello la arquitectura trata de resolver problemas en dominios probabilísticos, utilizando una representación inicial de un dominio determinista.

Utiliza técnicas de Ilp para analizar los datos obtenidos después de ejecutar la acción determinando si son éxitos o fracasos y finalmente utilizar esta información para generar mejores planes.

La herramienta que utilizan para inducir las acciones de éxito es ALEPH, esta arquitectura genera automáticamente reglas de control que guían al planificador en puntos de decisión del árbol de búsqueda.

5. INTEGRANDO LA PLANIFICACIÓN Y EL APRENDIZAJE DE MÁQUINAS PARA LA COMPOSICIÓN DE SERVICIOS

Algunos trabajos se han orientado a enfrentar la especificación parcial de los servicios, para lo cual; durante el proceso de planificación de la composición, han intentado obtener de manera automática diferentes aspectos de esta especificación.

(Carman *et al.*, 2007), propone un algoritmo que utiliza aprendizaje inductivo para hacer uso de los grandes volúmenes de información disponibles en la Internet. Este es un enfoque automático dirigido al aprendizaje de definiciones para fuentes de información en línea donde la importancia radica en hacer una buena selección de las entradas y salidas, obtener una descripción de ellas y determinar sus relaciones semánticas.

Con el fin de aprender dominios de HTN, (Robert *et al.*, 2005), han adoptado un nuevo enfoque llamado minería de *workflow* o (minería de procesos) que tiene por objetivo generar una descripción del *workflow* del servicio Web a partir del rastro de ejecución de los servicios Web.

6. CONCLUSIONES

Aunque existen varios trabajos orientados a utilizar diferentes técnicas de planificación en IA al problema de la composición de servicios, estos se han concentrado en generar dicha composición a partir de especificaciones completas del dominio del problema en las que se incluyen las respectivas especificaciones detalladas de los servicios.

Dentro de la composición de servicios Web usando diferentes técnicas de planificación IA, se han realizado diferentes trabajos relacionados con el manejo de la calidad durante este proceso los cuales, incluyen el manejo de tiempos de ejecución de los servicios asociados al proceso de la composición de los mismos, el manejo de la confiabilidad, costos y disponibilidad de los estos, como lo propone Igor (1998) y Liangzhao (2003). Sin embargo, la información asociada a estos parámetros, es obtenida antes de iniciar el propio proceso de planificación de la composición la cual es provista por los propios proveedores de los servicios, con lo cual la especificación está completamente especificada para este tipo de composición.

Particularmente, en los trabajos orientados a la composición y el manejo de la calidad, si bien existen algunos trabajos que se orientan al manejo de las fallas, la información relacionada con esta (métricas y valores), es conocida como paso previo al proceso de composición y es específicamente otorgada, como se indicó anteriormente, por los proveedores de los servicios.

Una forma de mejorar las técnicas de planificación, es el uso de técnicas de aprendizaje de máquinas. En este documento se han estudiado la técnica de aprendizaje de máquinas (ILP) programación lógica inductiva y, se han revisado diferentes trabajos en los que se aplica esta técnica para mejorar diferentes aspectos del proceso de la planificación IA.

Por último se han identificado dos casos particulares en donde de manera integrada, se han empleado la planificación y el aprendizaje en el proceso de composición de servicios, con el fin de completar las especificaciones del dominio en el que se realiza la composición de servicios Web; sin embargo, ninguno se ha orientado a completar de manera automática las especificaciones de las métricas empleadas en la composición orientada a la

calidad, mediante un proceso de planificación, ejecución y aprendizaje de la composición de servicios, en el caso de que se tengan una especificación parcial de este tipo de especificaciones.

Particularmente, no se ha encontrado en esta revisión del estado del arte trabajos relacionados con la especificación parcial de los parámetros de calidad relacionados con la detección de riesgos ocasionados durante la ejecución del servicio para minimizar el riesgo de falla de una composición de un servicio Web compuesto.

Lo anterior conlleva a indagar si ¿Es posible proponer un modelo integrado de planificación y aprendizaje que permita adquirir información relacionada con el factor de riesgo durante la composición de servicios Web? y teniendo en cuenta esto ¿Cuáles serían los criterios y las métricas respecto al factor de riesgo apropiadas a tener en cuenta en el proceso de composición de servicios Web?; ¿Qué algoritmos de planificación, permitirían el manejo adecuado de los criterios de calidad durante un proceso de composición?; ¿Cuáles son las técnicas de aprendizaje de máquinas inductivas adecuadas que permitan evaluar los criterios asociados al factor de riesgos durante la composición?; ¿Es posible plantear un modelo integrado de planificación y aprendizaje que permita adquirir información relacionada con el factor de riesgo durante la composición de servicios Web?; ¿Cómo es posible evaluar el modelo integrado de planificación y aprendizaje para llevar a cabo el manejo de riesgos durante la composición de servicios Web a partir de una especificación parcial del dominio?.

En respuesta a estas preguntas es interesante proponer un modelo integrado de planificación basada en costos y aprendizaje de máquinas basada en Programación Lógica Inductiva que permita de manera automática adquirir el conocimiento relacionado con el factor de riesgo que presentan los servicios que participan en una composición considerando su comportamiento en el propio servicio compuesto final como un todo con el fin de generar servicios Web compuestos que minimicen el riesgo de falla de tales composiciones.

Es así como a través de las diversas ejecuciones del propio servicio Web compuesto, se haría posible conocer el factor de riesgo de cada servicio y,

asociar este conocimiento a futuras composiciones similares al problema que originó este tipo de servicio Web compuesto con el fin de que el propio modelo de composición, pueda lograr que los planes de composición de servicios Web se vayan refinando con el tiempo y la experiencia del sistema.

Sin embargo, se debe tener en consideración los siguientes aspectos: la definición de los criterios y sus métricas relacionados con el factor de riesgo en la realización de un proceso de composición de servicios Web, la evaluación y selección de los algoritmos de planificación basada en costos que permitan el manejo de criterios de calidad con el fin de modelar los criterios asociados al factor de riesgo durante el proceso de planificación de la composición de servicios Web, la evaluación y selección de las técnicas de aprendizaje de máquinas inductivas que permitan el aprendizaje automático de los criterios asociados al factor de riesgo del proceso de composición de servicios Web y la unificación de los criterios de planificación basada en costos, aprendizaje de máquinas basada en ILP dentro de un solo modelo el cual permita de manera automática obtener la información relacionada con los riesgos de falla de los servicios Web en un proceso de composición, para generar planes de composición que se vayan refinando con el tiempo y la experiencia del sistema.

REFERENCIAS

- Aiqiang, Gao Dongqing Yang, Shiwei Tang and Zhang Ming. (2005). Web Service Composition Using Markov Decision, In: *Processes Proceedings of the WAIM*.
- Barreiro Claro Daniela, Albers Patrick and Hao, Jin-Kao, (2005). Web Services Composition, In: *Proceedings of the ICWS 2005 Second International Workshop on Semantic and Dynamic Web Processes*.
- Benton J., Do, M. B. and Kambhampati S., (2005). Over-subscription planning with numeric goals, In: *Nineteenth International Joint Conference on Artificial Intelligence*.
- Bergmann, R. and Wilke, W. (1996). On the Role of Abstractions in Case-Based Reasoning, In: *Proceedings of EWCBR-96 the European Conference on Case-Based Reasoning*.
- Bryce D. and Kambhampati S., (2007). A Tutorial on Planning Graph Based Reachability Heuristics, *AI Magazine*, 28, 47-60.

- Cardoso J., Sheth A., Miller J., Arnold J. and Kochut K., (2004). Quality of service for workflows and web service processes, *Journal of Web Semantics*, 1(3). 281–308.
- Carman Mark James and Knoblock Craig A. (2007). Learning Semantic Descriptions of Web Information, In: *International Joint Conference on Artificial Intelligence*.
- ChafI G., Chandra S., Kankar P. and Mann V. (2005). Handling Faults in Decentralized Orchestration of Composite Web Services, In: *International Conference on Service-Oriented Computing-ICSOC 2005*.
- Chakraborty D., Perich F., Joshi A., Finin T. and Yesha Y. (2002). A Reactive Service Composition Architecture for Pervasive Computing Environments. In: *7th Personal Wireless Communications Conference*.
- Chan May K. S., Judith Bishop and Luciano Baresi, (2007). Survey and Comparison of Planning Techniques for Web Services Composition *Technical Report*.
- Daniel Borrajo and Daniela Veloso, (1997). Lazy Incremental Learning of Control Knowledge for Efficiently Obtaining Quality Plans, *Artificial Intelligence Review*, 11(1–5). 371–405.
- Drew McDermott, (2004). The Optop Planner, In: *The International Planning Competition*.
- Eduardo Morales (2006). Programación Lógica Inductiva (ILP), <http://ccc.inaoep.mx/~emorales/> ultima visita septiembre de 2007..
- Evren Sirin, Bijan Parsia, Dan Wu, James Hendler and Dana Nau, (2004). HTN Planning for Web Service Composition Using SHOP2, *Journal of Web Semantics*, 1(4). 377-396.
- Haslum P. and Gener H., (2002). Admissible heuristics for optimal planning, In: *International Conference on AI Planning and Scheduling*.
- Igor Jurisica, Brian A. Nixon, (1998). Building Quality into Cases Based Reasoning Systems, *Lecture Notes in Computer Science*, 1413, 363-380.
- Jens Claßen, Patrick Eyerich, Gerhard Lakemeyer and Bernhard Nebel, (2007). Towards an Integrations of Golog and Planning, In: *20th International Joint Conference on Artificial Intelligence*.
- Jesús Lanchas, Sergio Jiménez, Fernando Fernández and Daniel Borrajo, (2007). Learning action durations from executions, In: *Working notes of the ICAPS'07 Workshop on AI Planning and Learning*.
- José Luis Ambite, Craig A. Knoblock and Steven Minton, (2000). Learning Plan Rewriting Rules, In: *Fifth International Conference on Artificial Intelligence Planning and Scheduling*.
- Jörg Hoffmann, (2003). The Metric-FF Planning System: Translating Ignoring Delete Lists to Numeric State Variables, *Journal of Artificial Intelligence Research*, 20.
- Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam and Quan Z. Sheng, (2003). Quality Driven Web Services Composition, In: *Proceedings of the 12th international conference on World Wide Web*.
- Martin Diaz, O, Ruize Cortes A., Duran A and Muller C. (2005). An Approach to Temporal-Aware Procurement of Web Services, In: *International Conference on Service-Oriented Computing*.
- Massimo Paolucci, Katia Sycara and Takahiro Kawamura (2003). Delivering Semantic Web Services, In: *Proceedings of the 12th International World Wide Web Conference*.
- McDermott D. (1999). Using Regression-match Graphs to Control Search in Planning. *Artificial Intelligence*, 109,(1–2). 111–159.
- McIlraith S., Son T. C., and Zeng H. (2001). Semantic Web Services, *IEEE Intelligent Systems*, 16(2). 46 – 53.
- Menasce D., and Almeida V. (2002). Capacity Planning for Web services, *Springer Berlin / Heidelberg*, 2459, 293-302.
- Michele Trainotti, Marco Pistore, Fabio Barbon, Piergiorgio Bertoli, Annapaola Marconi, Paolo Traverso and Gabriele Zacco, (2006). ASTRO: Supporting Web Service Development by Automated Composition, Monitoring and Verification, In: *Proceedings of the 16th International Conference on Automated Planning and Scheduling*.
- Mithun S. (2004). Automatic Composition and Invocation of Semantic Web Services, *Tesis para Mag*, University of Maryland.
- Muhammad Afzal Upal, (2001). Learning Plan Rewriting Rules, In: *fourteenth International Florida Artificial Intelligence Research Society Conference*.
- Nada Lavrac and Saso Dzeroski, (1996). Inductive Logic Programming Techniques and Applications, *Springer Netherlands, Holanda*, 23(1), 103-108.
- Nau D., Ghallab M. and Traverso P. (2004). Automated Planning: Theory and Practice, *Morgan Kaufman*.

- Nilsson N. J.(1994), Principles of Artificial Intelligence, *Morgan Kaufmann*, 1994.
- OWL-S. Semantic Markup for Web Services. <http://www.w3.org/Submission/OWLS/>, última visita septiembre de 2007.
- Paolucci M., Kalp D., A. Pannu, Shehory O. and Sycara K. (1999). A Planning Component for RETSINA Agents, In: *Pre-Proceedings ATAL'99*.
- Prashant Doshi, Richard Goodwin, Rama Akkiraju and Kunal Verma, (2005). Dynamic Workflow Composition: Using Markov Decision Processes, In: *International Journal of Web Service Research*, 2(1), 1-17.
- Qian Shuzhong, Lu SangLu and Lie Xie, (2005). Mobile Agent Based Web Services Composition, *Springer Berlin / Heidelberg*, 3795, 35-46.
- Ran Sh. (2003). A Model for Web Services Discovery With QoS, *ACM SIGecom Exchanges*, 4(1), 1–10.
- Robert Gombotz and Schahram Dustdar, (2005), On web services workflow mining, *Business Process Management Workshops*.
- Sergio Jimenez, (2007) Planning and Learning under Uncertainty, *disertacion para Ph.D., Escuela Politécnica Superior Ingeniería Informática, Universidad Carlos III de Madrid*.
- Sergio Jimenez, Fernando Fernandez and Daniel Borrajo, (2006). Inducing non-deterministic actions behavior to plan robustly in probabilistic domains”, In: *Working notes of the ICAPS'06 Workshop on Planning under Uncertainty and Execution Control for Autonomous Systems*.
- Sheila McIlraith and Tran Cao Son, (2002). Adapting Golog for Composition of Semantic Web Services, In: *Proceedings of the 8th International Conference on Principles and Knowledge Representation and Reasoning*.
- Sherchan W, Krishnaswamy Sh and S. Loke W. (2005). Relevant Past Performance for Selecting Web Services, In: *International Conference on Quality Software*.
- Stephen Muggleton and Luc De Raedt, (1994). Inductive logic programming: Theory and methods, *Journal of Logic Programming*, 19/20, 629-679.
- Web services HP, Web Services Concepts: a Technical Overview, HP Document, http://www.bluestone.com/downloads/pdf/web_services_tech_overview.pdf, última visita septiembre de 2007.
- WSMO. Web Service Modeling Ontology, <http://www.wsmo.org/2004/d2/v1.0/>, última visita septiembre 2007.
- WSDL-S. Web Service Semantics WSDL-S, <http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.pdf>, última visita febrero 2007
- Yoon S., Fern A. and Givan R.. (2006). Learning heuristic functions from relaxed plans, In: *Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling*.
- Yu T. and Lin K. J. (2005). Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints, In: *International Conference on Service-Oriented Computing*.
- Zeng L., Benatallah B., Anne H. H., Marlon Dumas, Jayant Kalaganam and Henry Chang. (2004). QoS-aware Middleware for Web Services Composition, *IEEE Transactions on Software Engineering*, 30(5), 311–327.
- Zimmerman T. and Kambhampati S. (2005). Using memory to transform search on the planning graph, *Journal of Artificial Intelligence*, 23, 533-585.