

Composición de Servicios Web: Clasificación y Evaluación de propuestas

Ana C. Alonso de Armiño¹, Pablo R. Fillottrani²

¹ Facultad de Informática. Universidad Nacional del Comahue.
Neuquén. Argentina
anacolinaalonsode@gmail.com

² Dpto de Ciencias e Ingeniería de la Computación. Universidad Nacional del Sur.
Comisión de Investigaciones Científicas, Provincia de Buenos Aires.
Bahía Blanca. Argentina
prf@cs.uns.edu.ar

Resumen. El advenimiento de los servicios Web y la arquitectura orientada a servicios (SOA) están provocando importantes cambios en la construcción de los sistemas y la forma en que esos sistemas interactúan con los sistemas externos. Esta tecnología se encuentra en la cúspide en cuanto a compatibilidad de los componentes de software, lo cual permitirá la reducción de los costos de desarrollo de sistemas de software a la vez que permite mejorar las características y funcionalidad ofrecidas, permitiendo integrar y reusar componentes de software, etc.

En este trabajo se analizan, los problemas y cuestiones actualmente en estudio, y en especial las dos técnicas principales de composición: basadas en workflow y basadas en planning. También se describen brevemente los estándares y propuestas surgidas en torno al tema de composición.

Abstract. The Web Services arrival and the services oriented architecture (SOA) are to causing important changes in the way of make systems and the way of interaction between those systems and the external systems. This technology is in the top about compatibility of the software components, and so this will to allow to reduce the development costs and to improve the characteristics and functionalities to present, integrate and reuse software component, and so.

In this work we analyze the problems and the open questions actually in study and also the two primary techniques to composition: based on workflow and based on planning. We also describe the standards and proposal to emerge around the composition topic.

Keywords: Composición de Servicios Web, SOA, Workflow, Planning

1 Introducción

Dada la heterogeneidad de los sistemas existentes, tanto las plataformas como los lenguajes usados, la integración de los mismos es una misión complicada. Los servicios Web son una tecnología que posibilitaría el desarrollo de complejas aplicaciones construidas sobre la base de componentes de software disponibles en la Web, pertenecientes a diferentes empresas y ejecutándose en plataformas heterogéneas. Por ello, uno de los temas a los que se han abocado los investigadores es la composición de los servicios Web, para posibilitar el reuso del software existente así como la integración de los mismos.

El modelo básico de Servicios Web consiste de tres tipos de entidades: el proveedor de servicios, el registro y el cliente que consume los servicios. El proveedor de servicios crea un servicio y lo ofrece, haciéndolo conocer a través del registro en el cual debe publicar la descripción en algún lenguaje basado en XML. El consumidor del servicio encuentra su descripción en el registro y la información para ponerse en contacto con el proveedor e invocar el servicio Web. Los tres pasos básicos son publicar, ligar e invocar. Para que sea posible la comunicación entre aplicaciones que se ejecutan en diferentes plataformas es necesario contar con estándares para realizar cada uno de estos pasos. Los estándares básicos están constituidos por:

Web Service Description Language WSDL, que usa el formato XML para describir los servicios Web (parámetros de entrada y salida, tipos de dato y protocolos de transporte tales como HTTP, etc).

Universal Description Discovery and Integration standard UDDI, es el registro donde se publica la información sobre el proveedor de servicios y los servicios ofrecidos.

Simple Object Access Protocol SOAP, es un protocolo usado para el intercambio de información XML entre las entidades.

Esta infraestructura puede ser suficiente para un escenario simple donde solo hay interacciones entre un cliente y un servicio Web. Pero cuando la implementación de un servicio Web requiere la combinación de otros servicios existentes surge la necesidad de replantear este esquema. En un servicio compuesto la lógica del negocio es implementada por varios servicios. El uso de lenguajes de programación convencionales y las plataformas heterogéneas en que conviven los servicios Web no son suficientes para realizar la composición de servicios Web, se requiere un middleware que provea la abstracción e infraestructura necesarias. Un middleware debe proveer mínimamente un lenguaje y modelo de composición, un entorno de desarrollo con una interface gráfica para incorporar y sacar componentes y un entorno de ejecución para ejecutar la lógica del negocio.

La composición de servicios presenta varios problemas en especial cuando se pretende automatizar esta tarea:

Especificación ambigua y observabilidad parcial. El modelo de interacción del usuario puede variar en el grado de completitud y observabilidad y ser diferente al modelo de interacción del servicio compuesto ya que el cliente no sabe cómo está constituido el servicio compuesto, lo cual dificulta la tarea de encontrar un servicio Web que satisfaga los requerimientos.

Especificación ambigua o incompleta de los servicios Web (falta marcado semántico), la incorporación de semántica y el uso de ontologías permiten superar este problema. También hay otras propuestas como las técnicas de aprendizaje, INDIGO es un sistema de planificación que puede realizar el marcado semántico de los servicios[31].

Soporte de Cambios. El entorno cambiante afecta la correctitud de la composición diseñada, hay algunas propuestas como las técnicas de adaptación de workflow para afrontar este problema. También si la interface de un servicio cambia esto no debe afectar el servicio compuesto; una solución es construir una capa de funcionalidad sobre la pila de estándares de composición, como se describe en [18].

Acuerdo entre las partes que interactúan. Por un lado los sistemas heterogéneos donde se ejecutan e interactúan los servicios imponen la necesidad de plantear estándares que permitan superar las incompatibilidades. Además debe existir un acuerdo entre el servicio compuesto y los servicios individuales, éstos últimos pueden especificar políticas con relación a la participación en un servicio compuesto [2]. También es necesario realizar la verificación de la composición para comprobar la sanidad y correctitud de la misma.

2. Problemas de la Composición

Un gran cambio se da en la industria de desarrollo al adoptar el enfoque orientado a servicios, donde se deben investigar nuevas tecnologías, métodos y herramientas que soporten la composición flexible, confiable, fácil de usar, de bajo costo y tiempo, y dinámica de las aplicaciones.

El desarrollo orientado a servicios implica la implementación de interfaces bien definidas para las unidades funcionales de software de forma que puedan ser integradas. La composición de servicios apunta a las técnicas para la integración de esos servicios.

La composición de servicios es la orquestación de un número de servicios para proveer otros servicios mas ricos que satisfagan algunos requerimientos del usuario. Las técnicas de composición implican expresar tanto los servicios elementales como los compuestos, así como la definición de un servicio compuesto requiere expresar los flujos de control y la información que se intercambia entre los servicios que lo componen.

La composición puede ser realizada en forma manual en tiempo de diseño conociendo previamente qué servicios intervendrán en la composición. La composición automática en cambio es de naturaleza más dinámica, permite optimizar algunos parámetros como ancho de banda y seleccionar de entre los servicios que están disponibles en ese momento, es una forma de modelado mas flexible importante para los escenarios del mundo real. Además el número creciente de servicios disponibles y la necesidad de realizar composiciones complejas hacen que la composición manual sea inadecuada.

Otra clasificación de la composición de servicios diferencia la obligatoria (mandatory) donde se requiere el correcto funcionamiento de todos los componentes, y la composición opcional donde los servicios pueden no estar en operación en el

momento en que se realiza la ejecución del servicio compuesto sin afectar la funcionalidad del mismo.

Para que la composición de servicios sea posible se requieren formas comunes de pasar la información entre los servicios. Estándares tales como WSDL, SOAP y HTTP constituyen la infraestructura de comunicación mas común y junto con XML permiten gran flexibilidad para la codificación y transformación de los datos.

La composición automática tiene la ventaja de que es posible la sustitución de servicios concretos cuando ocurre una falla por ejemplo, pero no es posible actualmente adaptarse y manejar los nuevos servicios que ofrecen nuevas funcionalidades o asegurar el funcionamiento de una composición si los servicios seleccionados son de-registrados durante la ejecución de la composición.

La composición automática de servicios se realiza sobre un requerimiento de un servicio en particular, por lo que es posible considerar las peculiaridades de la solicitud. En cambio la composición manual se realiza sobre un requisito general que luego responderá a las necesidades de varios usuarios, debiendo ser por este motivo, lo suficientemente general como para poder responder a un conjunto de requisitos.

Para que la automatización sea posible es necesario contar con ontologías que provean los conceptos y contar con lenguajes de descripción adecuados que ofrezcan construcciones del lenguaje apropiadas, de forma que se pueda realizar la descripción semántica de las propiedades funcionales y no funcionales de los servicios.

Todo proceso de composición de servicios Web, representado en la figura 1, debe ser iniciado por un servicio solicitante, que puede representar al usuario. El proceso de definición y composición de servicios puede requerir la traducción de lenguajes, de lenguajes de diseño a un lenguaje más formal que es utilizado por el generador del proceso de composición. Pueden generarse mas de un modelo de composición, en especial debido a que varios servicios tienen funcionalidad similar por lo que luego los servicios compuestos deben ser evaluados usando la información de los atributos no funcionales, y luego de seleccionar un servicio compuesto éste será ejecutado.

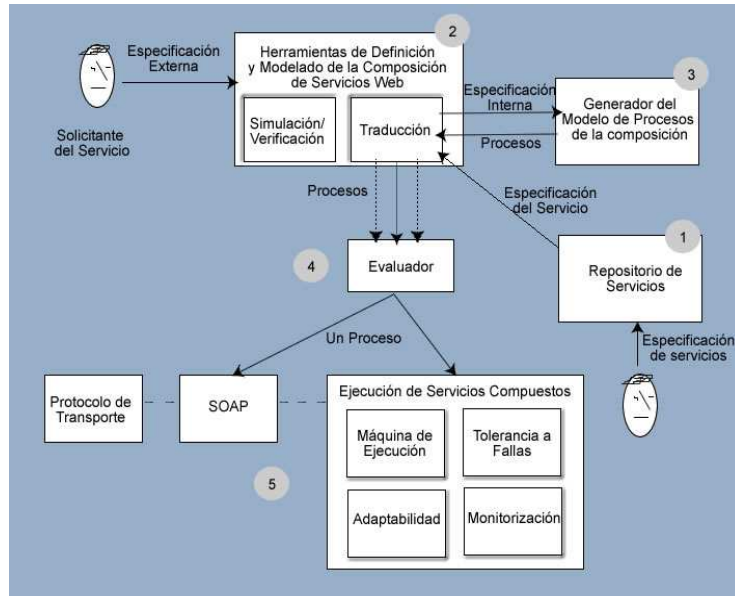


Fig.1. Arquitectura genérica de composición.

2.1 Workflow

Un servicio compuesto puede ser visto como un workflow ya que incluye un número de servicios atómicos que se comunican generando flujos de control y de datos. La ventaja de esta propuesta es que workflow es una técnica ampliamente estudiada, por lo que los avances en relación con la generación de workflow flexibles y que se adapten automáticamente pueden ser aprovechados al diseñar técnicas de composición usando esta herramienta.

Los avances en el campo de los servicios electrónicos requieren que la tecnología de workflow sea cada vez mas eficiente, flexible y fácil de usar para que pueda integrarse con otras tecnologías como XML y servicios Web.

2.2 Planning

Esta técnica supone que cada servicio puede ser especificado en términos de sus precondiciones y efectos, los cuales corresponderían a los parámetros de entrada y de salida respectivamente. El estado del mundo integra a las precondiciones, el estado antes de la ejecución de servicio, y el estado resultante tras la ejecución del servicio integra los efectos. A continuación se describen los métodos de composición basados en IA Planning:

- **Cálculo de situación:** es un método para modelar sistemas dinámicos basado en lógica de primer orden, modela en forma explícita el hecho de que distintas situaciones se puedan dar en el tiempo. Se definen agentes que razonan sobre los servicios Web y los usuarios especifican los requerimientos y restricciones.
- **Planning basado en reglas:** en este método se usan reglas de composición sintácticas y semánticas. Se distinguen cuatro fases para llevar a cabo la composición: especificación (especificación a alto nivel de la composición), matchmaking (generación de planes), selección (selección de un plan usando parámetros de calidad) y generación (genera una descripción de la composición a realizar).
- **Model Checking:** en esta técnica se utiliza un procedimiento para determinar en forma automática si las especificaciones son satisfechas por los grafos que representan el estado actual del problema y cómo se comportan las soluciones parciales con respecto a las metas.
- **HTN (Hierarchical Task Network):** en este enfoque las tareas son el concepto central y el sistema de Planificación HTN[38] se basa en la descomposición de las tareas en un conjunto de subtareas y estas a su vez en un nuevo conjunto de subtareas, el proceso continúa hasta que se llega a tareas atómicas (o primitivas) las cuales pueden ser ejecutadas directamente por la invocación de algunas operaciones atómicas. Puede incorporar información semántica a través de construcciones como *efectos, condiciones, y restricciones y criticos*.

3 Requerimientos para la Composición de Servicios Web

La composición de servicios Web debe atender algunas cuestiones que se enumeran a continuación.

1) Coordinación

Cuando se componen servicios se requiere la coordinación de la ejecución de las operaciones. Los protocolos usados en el modelo inicial, tales como WSDL no son suficientes, ya que por ejemplo no se puede expresar el orden en que deben ejecutarse las operaciones ni expresar qué información debe ser obtenida antes de realizar una operación dada. Tampoco es posible expresar si una operación debe ser realizada dentro de una transacción. WSCI es un estándar que provee nuevas construcciones para expresar el comportamiento de un servicio en el contexto de un proceso dado, el cliente que invoca el servicio sabe cómo se debe interactuar con el servicio.

Otros estándares surgidos para atender esta cuestión son WS-Coordination de IBM y OASIS [25][27] y WS-CF de Sun y ORACLE [26] y <http://www.infoworld.com/d/developer-world/sun-oracle-others-propose-transaction-specification-062> [6].

La principal capacidad del WS-Coordination [25] es la posibilidad de registrar un

servicio como integrante de una función específica de una clase de dominio.

WS-Context provee un modelo de sesión para el entorno de los servicios web. Los mensajes SOAP que son procesados dentro del alcance de una actividad contienen un header de contexto que permite identificar claramente la actividad. WS-Coordination extiende este modelo con protocolos para definir el contexto de registración extendiéndolo para registrar el end-point de registración. La registración en el contexto de una actividad agrega información sobre el grupo de actividades, lo cual puede ser de utilidad para coordinar señales.

La coordinación es un requerimiento presente en varios aspectos de las aplicaciones distribuidas, por ejemplo en workflow, transacciones atómicas, replicación, seguridad, actividades business-to-business, etc.

El framework presentado en [27] está integrado por tres servicios componentes: un servicio de Activación, con una operación que le permite a una aplicación crear una instancia o contexto de coordinación; un servicio de Registración, con una operación que le permite a una aplicación registrar protocolos de coordinación; y un conjunto de tipos de protocolos de coordinación. Estos elementos se ven diagramados en la figura 2.

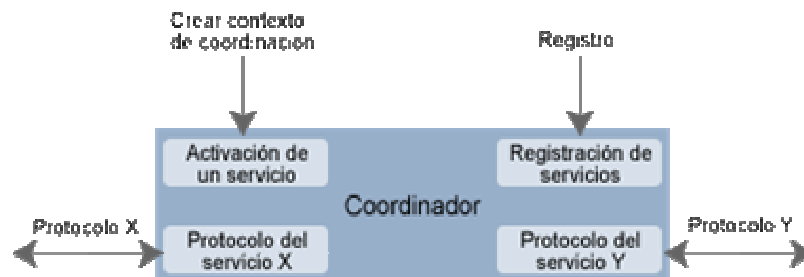


Fig. 2. Elementos que integran el framework de coordinación.

Las aplicaciones crean el contexto de coordinación de una actividad usando el servicio de activación que luego puede ser enviado a otras aplicaciones. Una aplicación que recibe un contexto puede usar el servicio de registración de la actividad original o el impuesto por el coordinador. El contexto de coordinación sirve para pasar información entre los servicios que son coordinados.

2) Transacción

Las transacciones son un concepto fundamental para la construcción de aplicaciones distribuidas confiables, definiendo un mecanismo para asegurar que los participantes en una aplicación respetan mutuamente un acuerdo. Cuando las operaciones deben ser ejecutadas dentro de transacciones deberían asegurarse que se cumplen los requisitos de atomicidad, consistencia, integridad y durabilidad. Algunos estándares que han surgido para atender esta cuestión son WS-Transaction [27][28] [3,4,5] propuesto por IBM y WS-TXM [7] propuesto por Sun.

La especificación de WS-Transaction está integrada por [27]: Protocolos para transacciones atómicas, que manejan actividades de corta duración. El Completion protocol es usado por una aplicación para decirle al coordinador si una transacción fue completada (commit) o abortada. El protocolo de dos fases (2PC) define como múltiples participantes acuerdan sobre la terminación de las transacciones: en la primer fase todo el trabajo es completado, siendo exitoso el resultado de una actividad si todas las operaciones fueron realizadas o no exitoso si no todas las actividades pudieron ser realizadas, lo ocurrido es informado a otros usuarios en la segunda fase. Protocolos para transacciones de negocios, manejan actividades largas, por lo que se requieren mecanismos para resolver fallas y mecanismos de compensación de forma de que las actividades que han sido completadas exitosamente no sean afectadas.

El contexto de coordinación puede tener un atributo Expires, que especifica el momento máximo en que la transacción puede terminar, en caso de no cumplirlo se puede decidir hacer un roll back de la transacción.

3) Contexto

Una actividad representa la ejecución e interacción de un conjunto de servicios Web, estas interacciones se relacionan a través del contexto, y se define así un agrupamiento conceptual de servicios que cooperan para realizar una tarea. El contexto es la forma concreta en el cual se efectúa dicho agrupamiento, y es representado por la información usada por los servicios que debe ser actualizada durante la ejecución. Un estándar para realizar esta especificación es WS-Context [29] [8] que integra los estándares WS-CAF propuestos por Sun.

El elemento principal de la especificación del WS-Context es la estructura del contexto, la cual define el modelo organizacional de la información del contexto, a través de estructuras anidadas (relación padre-hijo) y mecanismos para pasar información del contexto. Se define el Contexto del Servicio para manejar los contextos de las actividades, define el alcance de una actividad y cómo la información sobre ella puede ser referenciada y propagada en entornos distribuidos. Un contexto es identificado por una URI y puede ser referenciado por múltiples servicios Web que están asociados a la misma actividad. El Contexto del Servicio mantiene un repositorio con todos los contextos compartidos, los cuales pueden ser propagados dentro de mensajes.

4) Conversación

Modelar las conversaciones facilita el descubrimiento y ligadura dinámica de servicios, validación de la composición y la generación de templates de composición. Algunos estándares creados para apuntar a esta cuestión son WSCL(Web Services Conversation Language) [1] y WSCI (Web Service Choreography Interface) [36]. También están relacionados con este tópico el WS-Coordination y WS-Transaction ya que proponen las conversaciones específicas que pueden usarse para coordinar la interacción entre las partes.

4 Criterios de Comparación

Los investigadores han conducido sus trabajos a través de varias dimensiones, a continuación presentamos un análisis de los enfoques que se han propuesto para atender las cuestiones relacionadas con la composición de servicios Web:

- **Composición estática vs. composición dinámica:** considera el momento en que es realizada la composición. Algunos enfoques son: composición dirigida por modelos, composición dirigida por reglas del negocio, y composición declarativa.

En la composición estática los servicios son elegidos y ligados en tiempo de diseño, Microsoft Biztalk y Bea WebLogic son algunos motores de composición de este tipo. Si un servicio es reemplazado por otro o deja de funcionar se debe ligar al nuevo servicio y algunos casos puede requerir redefinir los procesos del negocio. La composición dinámica permite la adaptación a los cambios de este tipo, un motor que entra en esta categoría es eFlow y StarWSCoP.

- **Composición manual vs. automática:** en el segundo caso entran en juego las tecnologías de la Web Semántica. La Web Semántica provee la información necesaria para determinar los servicios que mejor se adecuan para alcanzar una solicitud. Y a través de diversos enfoques tales como Procesos de negocios e Inteligencia artificial se puede llevar a cabo la composición automática. Las técnicas de Inteligencia Artificial han resultado adecuadas para realizar la tarea ya que es posible razonar, realizar inferencias, adaptar la composición diseñada a cambios en el entorno, etc. También el uso de workflow y las técnicas de adaptación presentadas en varias propuestas como en [10].

Los principales problemas para implementar la composición automática de servicios son la selección y la interoperación de los mismos, donde se debe enfocar en las características sintácticas y semánticas de los servicios y ser transparente al usuario. Para poder realizar esto las ontologías constituyen una conceptualización compartida y capturan la semántica de los servicios Web. Algunos sistemas propuestos que realizan composición dinámica son por ejemplo Colombo y StarWSCoP.

- **Ejecución centralizada vs. ejecución descentralizada:** cuando se trata de un servicio compuesto es necesario controlar la ejecución de los servicios implicados. Esto puede ser hecho en forma centralizada o descentralizada. La ejecución centralizada es similar al paradigma cliente-servidor donde el servidor controla la ejecución de los componentes, tal como lo hace la plataforma eFlow [17]. En la ejecución distribuida cada host tiene un coordinador que controla la ejecución de los servicios locales y debe colaborar con otros coordinadores para asegurar la ejecución ordenada y correcta de los servicios. La plataforma SELF-SERV [30] realiza este tipo de ejecución.

5 Comparación de los Enfoques

Dado que los servicios Web están en auge en la actualidad por su capacidad para integrar aplicaciones dentro y entre empresas, dada la posibilidad de realizar esta integración en entornos y plataformas heterogéneas, el bajo acoplamiento de los componentes (servicios) y la posibilidad de adaptarse a cambios en el entorno, por ejemplo cambios en los parámetros de calidad, es que los investigadores han apuntado sus esfuerzos para mejorar las tecnologías y propuestas existentes. Una de las cuestiones que se han investigado y aún continúa en estudio es el de la composición automática y dinámica de los servicios Web. Muchas de las propuestas se basan en las técnicas de *planning* y *workflow*. A continuación pretendemos realizar una comparación que nos permita vislumbrar todo el espectro de alternativas, analizarlas y compararlas, para poder hacer un análisis y reflexión sobre cada una de ellas en comparación con el resto.

Para realizar esta comparación debemos considerar varios espectros que ya hemos analizado: coordinación, transacción, contexto, conversación, composición estática/dinámica, composición manual/automática, y finalmente composición centralizada/descentralizada.

En la primera columna se describen los enfoques analizados. En la segunda columna se describe la forma en que cada uno de los enfoques representa la información del entorno (situación, contexto o estado el mundo), lo cual es importante ya que la composición y ejecución de los servicios se ve afectada por esto. En la tercera columna se describe la forma en que se representa la conversación y orquestación de los servicios Web, lo cual es fundamental dado que para completar una tarea los servicios deben interactuar, comunicarse, coordinarse e intercambiar información. La cuarta columna describe algunas consideraciones especiales que caracterizan cada uno de los enfoques. En la quinta columna se describe la forma en la que se lleva a cabo la ejecución de los servicios. Y en la sexta columna se enumeran algunos ejemplos de propuestas que implementan cada uno de los enfoques.

Tabla 1. Características de los enfoques para realizar la composición de servicios Web.

Enfoques	Representación del estado del mundo - contexto	Representación de la conversación - orquestación	Consideraciones	En ejecución	Ejemplos referencias - estándares
Máquinas de estados finitos	Estados	Colas de mensajes	Observador Clases mensajes	Mediador	Colombo[12] [19] [21] [22] [23]
Métodos de Planning	Situaciones	Acciones Precondiciones y efectos	FO-SMDP		Haley[11]
Cálculo de	Situaciones	Acciones	Observabilidad	Servicios	[20]

Situación	Fluentes		d parcial	de búsqueda de información	
Model Checking	Estados	Acciones	No determinismo Observabilidad parcial	Considera el contexto para ejecutar una acción Heurísticas	BPEL4WS EaGLE [13] [15]
Hierarchica Task-Network	Estados: átomos verdaderos	Acciones o Tareas	Descomposición de tareas	SHOP2 planifica el orden exacto de ejecución de las operaciones	Strips [14] SHOP2 [16]
Workflow		Actividad de trabajo Reglas	Control y monitoreo de los procesos		AZTEC[24]

En la clasificación realizada previamente podemos analizar la utilidad de los diferentes enfoques en las etapas de la composición: desarrollo del plan y ejecución del plan. Las técnicas que usan máquinas de estados finitos para diseñar la composición no se involucran con la etapa de ejecución de la misma, por lo cual son deficientes en el entorno de los servicios Web, donde se requiere una gran dinamicidad para poder adaptar la composición realizada en respuesta a cambios del entorno, de los requisitos, de la calidad, de la disponibilidad de los servicios, etc.

Las técnicas de planificación y las de workflow permiten considerar ambas etapas, teniendo en cuenta que a la hora de ejecutar el servicio compuesto puede ser necesario adquirir información (consideración de la observabilidad parcial, uso de servicios de búsqueda de información, etc) y se considera también el comportamiento no determinístico de los servicios. Para que estas cuestiones puedan ser consideradas se debe poder modificar el plan de composición a medida que se ejecuta. En el análisis que realizamos a continuación se efectúan las comparaciones entre las mismas.

Tabla 2. Comparación de los enfoques para realizar la composición de servicios Web.

	Características	Composición estática/ dinámica	Composición manual/ automática	Ejecución centralizada/ descentralizada	Adaptación
Máquinas de estados finitos					
Colombo [12]	Estado del mundo Considera no determinismo, almacenamiento local,	Composición dinámica usando la información de cada servicio y del	Logra automatización con técnicas basadas en Lógica proposicional	Mediador, se considera el valor de los parámetros según el almacenamiento	--

	restricciones de integridad Colas para cada <i>tipo</i> de mensaje	propio mediador	dinámica	o local	
Métodos de Planning					
Haley [11]	Enfoque jerárquico Lógica de primer orden para representar precondiciones y efectos	La composición es dinámica usando el conocimiento y las probabilidades	Calcula la probabilidad con que ocurren las acciones Crea una base de conocimiento Determina en forma automática el servicio	Se realiza en forma centralizada	--
Calculo de situación y Golog [20]	Usa estados, acciones, Usa calculo de situación para representar precondiciones y efectos Golog para representar acciones complejas	Los servicios web son compilados como cajas negras	Técnica de planning basada en operador	--	--
Servicios Web BPEL4WS [13]	Usa model checking Considera no determinismo observabilidad parcial metas extendidas	Se realiza en forma dinámica	Composición automática Usa el estado	Se realiza en forma centralizada	Detecta señales externas Obtención de información del dominio
SHOP2 [16]	Enfoque jerárquico Puede ejecutar procesos en DAML-S	La composición es dinámica usando el razonamiento y las fuentes de información	Realiza la composición automática Utiliza las precondiciones y hace inferencia y razonamiento Integra fuentes de información externa	Se ejecutan las tareas en el orden exacto en que son planificadas	--

Métodos de Workflow					
AZTEC [24]	Diagramas de flujo Eventos Repositorio que representa el contexto	Selección y ensamble de servicios dinámicamente mientras se ejecutan	Esquemas de procesos usando HTN Selección y ensamble de servicios automáticamente Uso de políticas y un componente de administración	Motor de ejecución dirigido por eventos	Schema Management, permite modificar los esquemas durante la ejecución
Ejecución adaptiva en un modelo orientado a servicios [35]	Grafo (actividades como nodos y flujos de control y de datos como arcos) Repositorio que guarda información sobre servicios	--	--	--	Selección de servicios que cubran parte del workflow, consideraciones de QoS, mecanismo de pesos.
A-WSCE [9]	Chequeo de la calidad en forma aleatoria Impacto de adaptación Estadísticas sobre volatilidad de los parámetros Múltiples workflows (VOC)	Creación de templates (etapa lógica) manualmente o usando planning Selección de un grupo de templates maximizando los valores de QoS	--	El Runtime Mannager ejecuta el workflow chequeando los valores de QoS.	Adaptación incremental Cambiar una instancia usando el mismo template (etapa física) Cambiar el template cuando no hay instancias disponibles (etapa lógica)
eFlow [17]	Los servicios se modelan como procesos de negocios, y son representados como grafos Considera transacciones	Reglas de selección de servicios para descubrimiento dinámico de servicios Los servicios se instancian	El usuario puede manualmente hacer cambios	Los procesos de negocios se configuran automáticamente Reglas de consistencia y de autorización	Responde a eventos (modificación dinámica de procesos) Descubrimiento

		en forma dinámica (en ejecución)			dinámico de servicios Selección de la conversación n dinámicamente Nodos multiservicio y genéricos
--	--	----------------------------------------	--	--	----------------------------------------------------------------------------------------------------------------------------------

6. Conclusiones

El paradigma Orientado a Servicios, cuya esencia se basa en permitir a dos entidades de software interactuar en forma automática, independientemente de las plataformas en las que se ejecuten, se sigue afianzando día a día con las propuestas y tecnologías que surgen del trabajo de grupos dedicados a profundizar en las diferentes cuestiones que aún permanecen abiertas. Es el caso de la composición de servicios Web, área que continúa en el foco de atención de los investigadores que siguen refinando y ampliando las propuestas con el fin de mejorar la capacidad de este paradigma para lograr el grado de automatización requerida.

En este trabajo hemos identificado algunos de los problemas que afectan la automatización de la composición de servicios Web, hemos descrito algunas de las propuestas que pretenden resolver dichos problemas y hemos comparado los diferentes enfoques. Hemos analizado algunas de las propuestas más relevantes en las dos técnicas más utilizadas para realizar la composición de los servicios Web, técnicas que en general han evolucionado en un primer nivel para automatizar esta tarea y luego en un segundo lugar para considerar y adaptar la composición realizada a los cambios en el entorno. Las propuestas que se basan en workflow para sintetizar la composición, tienen las características de ser dinámicas, automáticas y adaptables. Cada una de ellas con sus mecanismos propios. En cambio las técnicas que se basan en planificación en general no han evolucionado a tal punto, excepto la propuesta en la que se definen los servicios como Servicios Web en **BPEL4WS**[13] brinda estas características, las otras podrían evolucionar de manera similar para considerar los cambios producidos en el entorno, a través del uso del VOC para actualizar la composición realizada en forma dinámica durante la ejecución de la misma. Otras ideas que ayudarían a mejorar dichas propuestas serían por ejemplo considerar las preferencias de los usuarios al hacer la composición, de forma que ésta sea más flexible; considerar la compatibilidad de los servicios al realizar la composición; etc.

La elección de las alternativas para implementar los servicios Web y para llevar a cabo la composición, dependerán de las preferencias de cada desarrollador-usuario, ya que hay varias alternativas que han demostrado ser efectivas para realizar una implementación.

7 Referencias

1. "Web Services Conversation Language (WSCL) 1.0". <http://www.w3.org/TR/wscl10/>
2. B. Medjahed, A. Bouguettaya, A.K. Elmagarmid. "Composing Web Services on the Semantic Web". VLDB J., vol. 12, no. 4, pp. 333-351, 2003.
3. S. R. Ponnkanti and A. Fox. "SWORD: A Developer Toolkit for Web Service Composition". In Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA, 2002.
4. D. McDermott. "Estimated-regression Planning for Interactions with Web Services". In Proceedings of the 6th International Conference on AI Planning and Scheduling, Toulouse, France, 2002. AAAI Press.
5. Tim Berners-Lee, James Hendler and Ora Lassila, "The Semantic Web", Scientific American, May 2001.
6. "GRDDL (Gleaning Resource Descriptions from Dialects of Languages)." <http://www.w3.org/TR/2006/WD-grddl-primer-20061002/>
7. "SPARQL". <http://www.w3.org/TR/rdf-sparql-query/>
8. "EARL". <http://www.w3.org/TR/EARL10-Guide/>
9. G.Chafle, P.Doshi, J.Harney, S.Mittal, B.Srivastava. "Improved Adaptation of Web Service Composition using Value of Changed Information". In Proceeding of IEEE International Conference on Web Services, Salt Lake City, UT, 2007. ICWS 2007.
10. T.C.Au, U.Kuter, D.S.Nau. "Web Service composition with volatile information". En International semantic Web Conference, pages 52–66, 2005.
11. H.Zhao, P.Doshi. "Haley: A Hierarchical Framework for Logical Composition of Web Services". International Conference on Web Services (ICWS), July, 2007.
12. D. Berardi, D. Calvanese, G. De Giacomo, R. Hull, and M. Mecella. "Automatic composition of Web services in Colombo". In Proc. of 13th Italian Symp. on Advanced Database Systems, June 2005.
13. M. Pistore F. Barbon, P. Bertoli, D. Shaparau, P. Traverso. "Planning and Monitoring Web Service Composition". In The 11th International Conference on Artificial Intelligence, Methodologies, Systems, and Applications (AIMSA), pages 106–115, 2004.
14. Kutluhan Erol, James Hendler, Dana S. Nau. "Semantics for Hierarchical Task-Network Planning". Technical report CS-TR-3239, UMIACS-TR-94-31, Computer Science Dept., University of Maryland, March 1994.
15. P Bertoli, A. Cimatti, M. Pistore, M. Roveri, P. Traverso. "MBP: a Model Based Planner". In Proceeding of ICAI-2001 workshop on Planning under Uncertainty and Incomplete Information, Seattle, WA, 2001, pp. 93–97.
16. Dan Wu, Evren Sirin, James Hendler, Dana Nau, Bijan Parsia. "Automatic Web Services Composition Using SHOP2". In Proceedings of 2nd International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, (2003).
17. Fabio Casati, Ming-Chien Shan. "Dynamic and adaptive composition of e-services". In The 12th international conference on advanced information systems engineering (CAiSE 00) Pages: 143 – 163. 2001.
18. Santhosh Kumaran, Prabir Nandi. "Conversation Support for Web Services". <http://www-106.ibm.com/developerworks/WebServices/library/ws-conver/>
- 19 T. Bultan, X. Fu, R. Hull, and J. Su, "Conversation Specification: A New Approach to Design and Analysis of E-Service Composition", Proceedings of the 12th International World Wide Web Conference (WWW 2003), ACM, 2003, pp. 403–410.
20. S. McIlraith and R. Fadel. "Planning with complex actions". In Proceedings of the Ninth International Workshop on Non-Monotonic Reasoning (NMR'02), pages 356–364, Toulouse, France, April 19-21 2002.
21. D. Brand and P. Zafropulo. "On communicating finite-state machines". Journal of the ACM, 30(2):323–342, 1983.
22. Lucas Bordeaux, Gwen Salañ, Daniela Berardi, and Massimo Mecella. "When are Two Web Services Compatible?". In M.-C. Shan, U. Dayal, and M. Hsu, editors, Proceedings of the 5th International

- Workshop on Technologies for E-Services (TES'04), Toronto, Canada, volume 3324 of Lecture Notes in Computer Science, pages 15–28. SpringerVerlag, Berlin, 2004.
23. Richard Hull, Jianwen Su. “Tools for Composite Web Services: A Short Overview”. SIGMOD Record, 34(2):86–95, 2005.
 24. Vassilis Christophides, Richard Hull, Gregory Karvounarakis, Akhil Kumar, Geliang Tong, Ming Xiong. “Beyond Discrete E-services: Composing Session-oriented Services in Telecommunications”. Proceedings of the 2nd VLDB International Workshop on Technologies for e-Services (VLDB-TES 2001), LNCS, vol. 2193, Springer, 2001, pp. 58 – 73.
 25. Web Service Coordination Framework (WS-CF).
[http://www.oasis-open.org/2Fcommittees%2Fdownload.php%2F10889%2FWSCF-Working-12-22.pdf&rct=j&q=Web+Services+Coordination+Framework+\(WS-CF\)&ei=v1jES4PdEYUQuAe8zdWbDw&usg=AFQjCNEi-T4gOoIEhm60y6ZfdrkCEottFw](http://www.oasis-open.org/2Fcommittees%2Fdownload.php%2F10889%2FWSCF-Working-12-22.pdf&rct=j&q=Web+Services+Coordination+Framework+(WS-CF)&ei=v1jES4PdEYUQuAe8zdWbDw&usg=AFQjCNEi-T4gOoIEhm60y6ZfdrkCEottFw)
 26. Mark M. Davydov. “Managing State in Service-Oriented Architecture”.
http://www.oracle.com/technology/pub/articles/davydov_soa.html
 27. Web Service Transaction specifications, IBM.
<http://www.ibm.com/developerworks/library/specification/ws-tx/>
 28. Freund, Tom & Tony Story: “Transactions in the world of Web Services, Part 1. An overview of WS-Transaction and WS-Coordination”. In: <http://www-106.ibm.com/developerworks/webservices/library/ws-wstx1/>
 29. Web Services Context Specification (WS-Context) Version 1.0. OASIS Standard. <http://docs.oasis-open.org/ws-caf/ws-context/v1.0/wsctx.html>
 30. Sheng, Quan Z., Boualem Benatallah, Marlon Dumas & Eileen Oi-Yan Mak: “SELF-SERV – A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment”. Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002
 31. J.G. Plaza, J. Guzmán Luna, A. Ledesma Castilloa. “Sistema multiagente para la composición de servicios web semánticos”. Universidad Nacional de Colombia, Medellín.
 32. John Harney, Prashant Doshi. “Selective Querying for Adapting Hierarchical Web Service Compositions Using Aggregate Volatility”. IEEE International Conference on Web Services (IWCS), pages 43–50, 2009.
 33. John Harney, Prashant Doshi. “Speeding up Adaptation of Web service Compositions using Expiration Times”. In WWW, pages 1023–1032, 2007.
 34. Shirin Sohrabi, Nataliya Prokoshyna, and Sheila A. McIlraith. “Web Service Composition via the Customization of Golog Programs with User Preferences”. Conceptual Modeling Foundations and Applications, pages 319–334, 2009.
 35. J Klingemann, J Wasch, “Adaptive Outsourcing in Cross-Organizational Workflows”. K Aberer - 2001 – portal.acm.org
 36. BEA, Intalio, SAP, and Sun. “Web Service Choreography Interface (WSCI) 1.0”, W3C Document. <http://www.w3.org/TR/wsci/>, 2002.
 37. D. Fensel. Vrije. “The Web Service Modeling Framework WSMF”. Universiteit Amsterdam (VU). C. Bussler. Oracle Corporation.
 38. K. Erol, J. Hendler, and D. Nau. “Semantics for Hierarchical Task Network Planning”. 1994.