



Revista Ingenierías Universidad de Medellín
ISSN: 1692-3324
revistaingenierias@udem.edu.co
Universidad de Medellín
Colombia

Giraldo, Jorge; Guzmán, Jaime A.; Ovalle, Demetrio A.
INTEGRACIÓN DE PROCESOS DE NEGOCIO BASADOS EN SERVICIOS WEB: COREOGRAFÍA Y
SATISFACCIÓN DE RESTRICCIONES
Revista Ingenierías Universidad de Medellín, vol. 7, núm. 12, enero-junio, 2008, pp. 141-155
Universidad de Medellín
Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=75011517007>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

INTEGRACIÓN DE PROCESOS DE NEGOCIO BASADOS EN SERVICIOS WEB: COREOGRAFÍA Y SATISFACCIÓN DE RESTRICCIONES

Jorge Giraldo¹
Jaime A. Guzmán²
Demetrio A. Ovalle³

Recibido: 07/03/2006

Aceptado: 14/04/2008

RESUMEN

Los procesos de negocio en ocasiones necesitan funcionalidades adicionales que pueden alcanzarse empleando servicios del mismo dominio o de uno externo. Durante este procedimiento, los servicios web involucrados deben integrarse, de tal forma que sus funcionalidades individuales generen un comportamiento global. La coreografía de servicios web controla la secuencia de las interacciones entre los servicios durante el proceso de integración. Este artículo presenta una aproximación al modelado basado en restricciones de la coreografía de servicios web, a fin de lograr su automatización.

Palabras clave: Coreografía de servicios web, servicios web, satisfacción de restricciones.

¹ Ingeniero de Sistemas, Magíster en Ingeniería de Sistemas Universidad Nacional de Colombia, Medellín. Docente Facultad de Ingenierías Politécnico Colombiano Jaime Isaza Cadavid. E-mail: jegiraldp@unal.edu.co, jegiraldo@elpoli.edu.co Tel: 4255378.

² Magíster en Sistemas, Docente Escuela de Sistemas Universidad Nacional de Colombia, Medellín. E-mail: jaguzman@unal.edu.co. Tel: 4255378

³ PhD en Sistemas, Docente Escuela de Sistemas Universidad Nacional de Colombia, Medellín. E-mail: dovalle@unal.edu.co. Tel: 4255378

INTEGRATION OF WEB SERVICE-BASED BUSINESS PROCESSES: WEB-SERVICE CHOREOGRAPHY AND CONSTRAINT SATISFACTION

ABSTRACT

Business processes sometimes need additional functions which can be reached through the use of services of the same domain or an external one. During this procedure, web services involved should be integrated in such a way that its individual functions can generate a global behavior. Web-service choreography controls interaction sequences among services during integration process. This article shows an approach to modeling based on web-service choreography restrictions with the purpose of achieving its automation.

Keywords: Web services choreography, web services, constraint satisfaction.

INTRODUCCIÓN

La integración de servicios web, bajo la arquitectura orientada a servicios (SOA), ha sido un área de activa investigación en los últimos años; es un paradigma que utiliza servicios web como elementos fundamentales para el desarrollo de aplicaciones complejas (Papazoglou y Georgakopoulos, 2003). Existen dos conceptos que abordan el problema de la integración efectiva de procesos de negocio basados en tecnologías web; ellos son: la coreografía y la orquestación de servicios web (Peltz, 2003). Estos conceptos dirigen sus esfuerzos a describir de manera global las interacciones y cambios de estado, así como el flujo interno del proceso. En una interacción de servicios la coreografía puede verse como una capa superior en abstracción sobre la orquestación, ya que describe el comportamiento global de la aplicación compleja. Para el manejo y representación de estos conceptos existen diversos estándares como WSDL (W3C, 2002), XLANG (Microsoft, 2001), WSFL (IBM, 2001) y el WSCDL (Kavantzas et al, 2004) que facilitan su procesado. No obstante, en el campo académico la descripción de procesos basados en servicios web se realiza por medio de lenguajes de marcado como el OWL-S (Martin y Burstein, 2001) no es más que un lenguaje de marcado para servicios web, basados en una ontología de los mismos. La estructura de la ontología es motivada por la necesidad de proveer tres tipos esenciales de conocimiento sobre un servicio, cada uno caracterizado con el fin de satisfacer los siguientes interrogantes: ¿Qué funcionalidad provee el servicio? ¿Cómo es usado? ¿Cómo se interactúa con este?

La coreografía puede ser vista como un problema basado en restricciones, debido a que las interacciones pueden ser modeladas de tal manera que restricciones aplicadas a estas restrinjan la funcionalidad y desempeño de las entradas y salidas

de los servicios web. Un método que se acomoda al contexto de los servicios web es el problemas de satisfacción de restricciones (PSR) (Russell y Norving, 2004); este es un método de solución de búsqueda de estados a partir de un conjunto de restricciones.

En este trabajo se busca proponer una solución para automatizar la especificación de una coreografía de servicios web en el lenguaje WSCDL a partir de un grupo de servicios web y un conjunto de objetivos; así se pretende modelar, con base en restricciones, las interacciones entre los servicios que componen la coreografía. El presente artículo aborda la definición de los componentes básicos de la coreografía, así como aquellos requerimientos necesarios para lograr cierto grado de automatización sin perder la esencia de esta. Asimismo, se presenta una revisión de estado de arte de los PSR. La propuesta es implementar un modelo de restricciones, que valide información provista por la definición de los servicios web y genere una descripción de un proceso de coreografía.

El artículo presenta una revisión del estado del arte en área de la coreografía de servicios web y de los PSR. La idea que se propone es implementar un modelo de satisfacción de restricciones, que valide información provista por la definición interna de los servicios web y genere una definición de un proceso de coreografía de servicios web.

El documento se estructura de la siguiente manera: en la sección 2 se presenta el concepto de coreografía de servicios web. En la sección 3 se presenta una revisión del estado de arte para la coreografía, tanto del ámbito industrial como académico. En la sección 4 se exponen los conceptos básicos de los PSR y los trabajos hechos en torno a estos. En la sección 5 se analizan los componentes básicos de la coreografía y su posible manejo mediante su representación como PSR. Finalmente en la sección 6, se exponen las conclusiones y se plantean trabajos futuros.

I. COREOGRAFÍA DE SERVICIOS WEB

Los servicios web día a día están presentes en las transacciones hechas por negocios que aprovechan estándares web. En algunos casos los requerimientos de información planteados por el usuario no son alcanzados a través de un simple servicio web, sino que se hace necesario combinar varios de ellos para generar un servicio que pueda satisfacer los objetivos del usuario. Las interacciones que ocurren entre estos a fin de lograr su cometido es interés de la coreografía, ya que esta describe la secuencia y condiciones en la que los datos son intercambiados por los servicios web.

La coreografía de servicios web describe las interacciones observables entre los servicios, con el fin de lograr los objetivos propuestos. El término observable se refiere a aquellas interacciones que sean referenciadas como pertinentes para el objetivo a alcanzar durante la integración.

La coreografía puede entenderse como un proceso público y no ejecutable; es público porque

define el comportamiento común y globalmente visible entre los diferentes participantes en una interacción; por otro lado, es no ejecutable porque no está pensado para ser llevado a cabo, sino para actuar como un protocolo de negocio que dicta reglas de interacción que deben ser cumplidas por las entidades participantes (Peltz, 2003).

En el siguiente pseudocódigo se aprecia la diferencia entre orquestación y coreografía. En el lado izquierdo se define la coreografía; aquí un servicio A envía una petición a un servicio B, luego B envía dicha petición en paralelo a C y D, y espera una confirmación, combina las respuestas de C y D y se la envía a A.

En el lado derecho se encuentra la definición del servicio B, desde le punto de vista de la orquestación; se puede apreciar el intercambio de mensajes, pero la descripción se hace a partir de un individuo, lo que conllevaría a realizar dicha descripción en todos los servicios incluyentes. Es posible realizar una coreografía mediante un conjunto de orquestaciones de cada uno de los componentes que la integran.

<p>*A envía petición a B</p> <p>*En paralelo, B reenvía la petición</p> <p>Espera una confirmación de recibido y respuesta por parte de C y D</p> <p>*B Combina las dos respuestas de C y D y se la envía a A.</p>	<p>Recibir petición de A</p> <p>hacer en paralelo</p> <p>*enviar petición a C</p> <p>recibir confirmación de C</p> <p>recibir respuesta de C</p> <p>*enviar petición a D</p> <p>recibir confirmación de D</p> <p>recibir respuesta de D</p> <p>combinar respuestas de C y D y enviar respuesta combinada a A</p>
--	--

La coreografía de servicios web es tema de investigación por parte de varios grupos, tanto del ámbito industrial como del académico. En el siguiente apartado se expone la manera en que estos modelan y describen la coreografía.

2. DESCRIPCIÓN Y MODELAMIENTO DE UNA COREOGRAFÍA

En el ámbito industrial es común encontrar distintos estándares que ayudan a la descripción de la integración de procesos de negocio; estos

abordan tanto la orquestación como la coreografía; entre ellos se encuentran: XLANG (Microsoft, 2001), WSFL (IBM, 2001), WSCI (W3C, 2001), WSCDL (Kavantzas et al, 2004) y BPEL4WS (Andrews et al., 2003).

WSCL (Web Services Conversation Language) es un lenguaje estándar para conversaciones simples entre dos servicios web; con frecuencia es modelado como autómatas de estado finito (Kuno et al, 2001). XLANG es una especificación desarrollada por Microsoft para su servidor BizTalk; este estándar se ocupa de la creación de procesos de negocio y las interacciones entre proveedores de servicios. Por su parte, WSFL (Web Services Flow Language) fue una propuesta de la IBM que describe tanto los flujos privados como públicos.

La interfaz de coreografía de servicios web (WSCI) es una especificación gestada por Sun, SAP, BEA e Italio, que define un lenguaje basado en XML (Berners-Lee et.al, 2001) para la colaboración entre servicios web (Papazoglou y Yang, 2002). Esta especificación soporta correlación de mensajes, reglas de secuencias, manejo de excepciones, transacciones y colaboraciones dinámicas. Su sucesor es el lenguaje de descripción de coreografía de servicios web (WS-CDL), basado en sintaxis XML, que describe la colaboración punto a punto entre las partes involucradas y permite definir una vista global, así como el comportamiento global observable de los servicios web. La especificación de WS-CDL contiene las estructuras provenientes de su antecesor, lo que garantiza control y ordenamiento de los mensajes que componen las interacciones generadas entre los servicios web.

La especificación de BPEL4WS fue desarrollada por Microsoft, IBM, Siebel Systems, BEA y SAP. BPEL4WS modela el comportamiento de servicios web que participan en la integración de procesos de negocio. La descripción del servicio es ejecutada por un motor de orquestación que coordina todas las actividades y compensa el proceso global cuando ocurre algún error; lo que puede ser con mensajes

de inactividad o de advertencia.

En la figura 1 se presenta un fragmento de código WSCDL, en donde se describe parte de una coreografía de subastas electrónicas, la cual se compone de tres roles: comprador, subasta y vendedor; también se definen las relaciones entre estos roles y su tipo.

Desde el enfoque académico, Brogi et al (2004) presentan una formalización para las coreografías descritas en WSCI, por medio del enfoque del álgebra de procesos (Bergstra, 2001) en donde chequea la compatibilidad entre servicios web para su interoperación (Canal et al., 2001). La manera en que se representa la información es a través de la formalización de los procesos; se identifican en estos los canales de comunicación para las interacciones, valores para los datos intercambiados y la secuencia de dichos valores.

McKinley et al (1999) exponen un marco de trabajo para el desarrollo colaborativo de aplicaciones web; en sí es una capa intermedia la cual se rige por el paradigma objetual. Otro trabajo de investigación es el de Benatallah et al (2002), donde se describe el diseño e implementación de un sistema de composición de servicios web, usando un paradigma declarativo; la herramienta de representación de las interacciones son los diagramas de actividades de UML (Booch, 1998).

Burdett y Kavantzas (2003), por su parte, aprovechan la capacidad de comunicación que ofrecen los servicios web sobre redes de comunicación adicionando etiquetas de control a los mensajes SOAP embebidos en los documentos XML que son intercambiados durante una integración de servicios.

Wombacher et al (2004) argumentan que bajo la arquitectura orientada a servicios no existe en la actualidad un servicio web que realice una búsqueda y acoplamiento de procesos de negocio; por tanto, propone un esquema para solucionarlo; el nivel de abstracción manejado aquí para el modelamiento de la coreografía tiene en cuenta el flujo de mensajes y su tipo.

```

<package>
  <role name="Comprador"/>
  <role name="Subasta"/>
  <role name="Vendedor"/>
  <relationship name="Buyer_AS">
    <role type=" Comprador"/>
    <role type=" Subasta"/>
  </relationship>
  <relationship name="AS_Vendor">
    <role type=" Subasta "/>
    <role type="Vendedor"/>
  </relationship>
  <channelType/>
  <choreography name="GeneralChoreo">
    <relationship type="AS_Vendor">
    <relationship type="Buyer_AS">
    <variableDefinitions>
      <variable name="AuctionCreate" silent-action="true" role=" Subasta "/>
    <variableDefinitions/>
  
```

Fuente: Kuno, et al, 2001

Figura 1. Fragmento código WSCDL

Foster et al (2004) proponen una aproximación basada en un modelo para la verificación de las interacciones que conjuntamente trabajan para coordinar la composición de servicios web. Lo que se intenta es, a partir de la orquestación de cada uno de los servicios involucrados, trasladar dichas descripciones al dominio de los procesos de estado finito, en donde se identifican los mensajes. La solución desarrollada actualmente es un plugin (Foster, 2003).

En un posterior trabajo (Foster et al, 2006) se aborda el comportamiento de una composición de servicios web implementados en BPEL; el modelado se realiza en WSCDL. Los aspectos tenidos en cuenta son restricciones u obligaciones sobre las interacciones entre las partes; estas obligaciones son modeladas en diagramas de secuencia de mensajes.

Jansen et al (2005) proponen un modelo para poder extraer un protocolo a partir de una especificación de un programa y verifica el nivel de adherencia generado por el protocolo frente a su antecedente. Es complementaria la idea de la generación automática de una descripción de una coreografía de servicios web, ya que un protocolo de comunicación solo aplica reglas a los individuos, es decir, los comportamientos individuales, dejando a un lado la vista global (Vallecillo et al, 2000). El enfoque que se sigue en cuanto a la definición de coreografía es el manejo de los mensajes y algunas condiciones entre estos.

En Mendling y Hafner (2005) se expone un mapeo conceptual entre WSCDL y BPEL, de manera que pueda generarse una descripción de una orquestación a partir de un lenguaje que describe coreografías. Esta mapeo tiene en cuenta las etiquetas donde se describen los tipos de mensajes, como son de petición, respuesta o confirmación. A lo anterior se vislumbra soluciones en cuanto al manejo y análisis de documentos de definiciones de procesos a fin de lograr cierto de grado de automatización y formalidad en su manejo.

La siguiente tabla presenta un resumen de los distintos enfoques que se tienen para trabajar en dominios de la coreografía de servicios web. Aquí se comparan los trabajos realizados en torno a la coreografía versus características tales como: tipo de mensajes (sean de petición, respuesta o ambos), estructura del mensaje (formato de los datos), condiciones de envío de información (restricciones) y por último tecnología usada, determinando su participación con un SI o un NO.

Tabla 1. Enfoques hacia la coreografía.

	TIPO DE MENSAJE	ESTRUCTURA MENSAJE	CONDICIONES	TECNOLOGÍA	MECANISMO DE FORMALIZACIÓN
Brogi & Canal	Sí	No	No	WSCI, BPEL	Álgebra de procesos
Burdett & Kavantzias	Sí	Sí	Sí	SOAP	~
McKinley & Malenfant	Sí	Sí	No	Plugins	~
Benatallah, Dumas & Sheng.	Sí	No	Sí	Lenguajes declarativos	UML
Wombacher, & Fankhauser.	Sí	No	No	~	Autómatas de estado finito
Foster, Uchitel, & Kramer.	Sí	No	Sí	BPEL & Plugins	Procesos de estado finito, UML
Mendling & Hafner.	Sí	Sí	No	BPEL & WSCDL	~
Jansen, Yang.	Sí	No	Sí	~	Máquinas de estado finito

Fuente: elaboración propia

Se puede apreciar que pocos estudios se detienen a trabajar con la estructura del mensaje, al igual que con condiciones que estos deben cumplir para poder continuar con la coreografía planteada. En la columna de mecanismos de formalización en algunas celdas no existe información; esto se debe al nivel en que abordan las soluciones, ya que

algunas solo alcanzan fases de análisis, sin pensar en la implementación.

Debido a la versatilidad de descripciones para la coreografía de servicios web, es posible representarla desde un nivel en el cual solo se definen los mensajes a ser enviados y su secuencia, hasta condiciones específicas de la estructura de estos.

Por ello se hace necesario en la automatización de descripciones de coreografía de servicios web delimitar los alcances de la funcionalidad.

Una alternativa es aplicar restricciones que deben ser satisfechas por parte de los componentes involucrados en la coreografía a través de sus interacciones. Sistématicamente las restricciones pueden ser satisfechas por medio de propuestas provenientes de áreas como la inteligencia artificial. A este tipo de problemas pertenecen, los problemas de satisfacción de restricciones (PSR). En el siguiente apartado se presentan algunos conceptos básicos referentes al tema en cuestión y cómo estos pueden involucrarse en el contexto de los servicios web.

3. SATISFACCIÓN DE RESTRICCIONES

Un problema de satisfacción de restricciones (PSR) está definido por un conjunto de variables, X_1, X_2, \dots, X_n y un conjunto de restricciones, C_1, C_2, \dots, C_n . Cada variable X_i tiene asociado un dominio no vacío D_i , de valores posibles (Russell, 2004). Cada restricción C_i implica algún subconjunto de variables y especifica las combinaciones aceptables de valores para ese subconjunto. Un ejemplo conocido que puede ser formulado como un PSR es el clásico problema de coloración del mapa (Appel, 1977).

Algunos PSR con frecuencia requieren soluciones que maximicen una función objetivo, como en Larrosa y Meseguer (2003), donde se presentan algunas técnicas para resolver un PSR definido por una red de restricciones. Aunque existen más variaciones de dichas técnicas (Dechter, 2003), solo se presentan métodos como el de búsqueda, inferencia e híbrida.

Se distinguen dos tipos de búsqueda para resolver de una manera directa un PSR. La búsqueda sistemática (Barber y Salido, 2003) y la búsqueda

local (Morris, 1993). Los métodos de búsqueda local resultan en ocasiones eficaces en la resolución de muchos PSR. Estos utilizan una formulación de estados completa; el estado inicial asigna un valor a cada variable, y la función sucesora; por lo general, trabajan cambiando el valor de cierta variable a la vez.

En la búsqueda sistemática, el espacio del estado del problema se estructura mediante un árbol de búsqueda. Cada nodo del árbol representa una asignación de variables definida por el camino que une el nodo con la raíz. Los algoritmos de Backtracking (Barták, 1999) son la base fundamental de los algoritmos de búsqueda sistemática para la resolución de PSR.

Los PSR se han implementado en varios campos de la ciencia, entre ellos cuentan mucho los problemas de asignación de recursos, tanto en terminales aéreos (Chow, 1997), como en atracaderos de barcos (Kima et al., 2004). En áreas como la Ingeniería de Sistemas los PSR apoyan procesos como la localización y detección de errores en modelos (Piechowiak, 2005). En este trabajo se propone un procedimiento para detectar y corregir errores en los modelos de hardware y software; este procedimiento se trabaja desde el punto de vista de la Diagnosis (Console y Torasso, 1991). Un modelo basado en Diagnosis consiste en comparar el comportamiento que un componente puede ofrecer con otro de similares características, pero en ambiente real.

Con el uso de la programación orientada a componentes (Szyperski, 1997), aspectos como su distribución y su posterior integración abren la posibilidad de distintas aplicaciones en esta área bajo la perspectiva de la satisfacción de restricciones; en especial en ambientes distribuidos (Faltings y Yokoo, 2005).

En (Aiello y Papazoglou, 2002) se presenta una arquitectura de trabajo para la interacción de servicios web basada en planificación (Rao,

2003) y satisfacción de restricciones. Un lenguaje de peticiones para servicios web (WSRL - Web Services Request Language) se diseñó sobre la base de esta arquitectura. Las restricciones se presentan al momento de averiguar circunstancias que evitan la correcta generación del plan, ayudando a su reconfiguración, búsqueda de recursos o creación de interacciones adicionales.

En un posterior trabajo basado en conceptos manejados por el WSRL, se plantea enriquecer sus estructuras con componentes semánticos, lo que da paso a un lenguaje para la planificación de la composición de servicios (Yang y Papazoglou, 2002). Un trabajo similar codifica como restricciones las peticiones del cliente a un proceso de composición (Lazovik et al, 2005).

Un trabajo que se enfoca en la verificación a priori es (Cernura del Río, 2002a); especialmente trabaja en el diseño de software, y se fundamenta en el conocimiento disponible por parte del componente. Dicha solución tiene en cuenta restricciones de viabilidad inmediata y transferencia tecnológica, verificación estática y automática.

En Cernura del Río (2002b), se plantea una aproximación al dominio de los servicios web y sus procesos de integración como la orquestación y la coreografía; asimismo, los lenguajes que apoyan su definición y descripción, el BPEL y el WS-CDL. Se considera importante realizar dicha detección de errores en paralelo a la descripción de los procesos de orquestación y coreografía.

La satisfacción de restricciones es usada para representar y resolver problemas relacionados con el área de la inteligencia artificial; asimismo, ofrece un potencial llamativo para la representación del conocimiento. En Channa et al (2005), se emplean las restricciones y su posterior satisfacción en la composición dinámica de servicios web, para encontrar el conjunto apropiado de servicios, los cuales deben cumplir ciertos parámetros representados por las restricciones definidas.

4. DISCUSIÓN

Observando distintos enfoques y mecanismos de representación, manejo de información y validación, la coreografía de servicios web puede ser considerada mediante la simple definición de una secuencia de mensajes. Sin embargo, a fin de lograr un cierto nivel de automatización en la descripción de coreografías, los siguientes componentes son hasta ahora necesarios para lograrlo. El primero de ellos es el tipo de mensaje, que define los patrones de interacción; estos según estructuras propuestas por el estándar WSCDL son expresados como una petición y petición-respuesta. No obstante, variaciones de estas pueden ser determinantes al momento de aplicar restricciones sobre las interacciones.

Las restricciones a las interacciones deben ser tenidas en cuenta si se desea controlar la participación de los servicios web. Sin embargo, este punto es dependiente de la formalización que se haga a las interacciones, ya que la expresividad brindada facilita o no la definición de restricciones apropiadas a un proceso de coreografía.

Por último, y no menos importante, está la estructura de los mensajes; esto quiere decir acceso a los tipos de datos intercambiados y posibles manipulación o aprovechamiento de estos, al momento de validación de restricciones.

No obstante, cabe notar que todos estos componentes integran las interacciones; a esto se suma que son una actividad clave para expresar el intercambio de mensajes que pueden presentarse. Por ello es indispensable realizar un análisis y plantear una representación formal de tales interacciones, bajo el contexto de la coreografía de servicios web.

4.1 Interacciones y su formalización

Basadas en los elementos que dispone el WSCDL, se presentan las actividades interacción, asignación y nulo (Barros et al, 2005). Una interacción es el resultado del intercambio de información entre

los roles participantes y una sincronización posible entre estos. Una asignación declara un nuevo valor sobre una variable o una expresión de otra variable. Por último, la actividad nula no realiza ninguna operación, pero puede ser empleada para definir relaciones permanentes o validación de estas.

Otro componente son las actividades que presentan estructuras de control para su secuencia de intercambio; estas incluyen la secuencia, el paralelo y la escogencia. Una actividad secuencial describe una o más actividades que son ejecutadas secuencialmente. Una actividad en paralelo describe una o más actividades que se ejecutan en cualquier momento o en el mismo tiempo. La representación formal de las interacciones y especialmente en ambientes de procesos de negocio y su integración, como lo es la coreografía, ha venido siendo objeto de aproximaciones por parte del dominio del álgebra de procesos. El propósito de este documento es modelar mediante álgebra de procesos las interacciones de la coreografía de servicios para posteriormente definir descripciones y la manera de formalizarlas, y así puedan ser manipuladas por algoritmos de soluciones basadas en restricciones. El álgebra de procesos permite expresividad al momento de describir el comportamiento de los servicios web. Algunos autores (Mederiht y Bjorg, 2003) argumentan que las interacciones entre servicios web deben ser descritas por medio de lenguajes de descripción de procesos como el BPEL, pero este último no permite la definición de procesos globales de interacción ni tampoco razonar sobre el mismo.

El álgebra de procesos ha tenido sus acercamientos al mundo de los servicios web, en especial pi-cálculo (Parrow, 2001), pero su razonamiento en particular se orienta hacia la representación de flujos de trabajo, el cual se relaciona más con la orquestación; tal es el caso del trabajo de Pulhlmán y Weske (2005), donde facilitan la formalización de patrones para flujos de trabajo con pi-cálculo.

Otra álgebra de procesos es el CSS (Milner, 1989); su mayor potencialidad es poder ser traducida a cualquiera otra álgebra. Brogi et al (2004) presentan una formalización a coreografías descritas en WSCI; las características intrínsecas del WSCI sugieren el uso de álgebra de procesos para formalizarlo. A continuación se presenta una breve introducción a la sintaxis del CSS. Cabe destacar que operadores como restricción o condicional no son tenidos en cuenta, lo que limita un poco el enfoque que se persigue.

Un proceso en CSS está dado de la siguiente forma:

$$\begin{aligned} P := & 0 \mid \alpha.P \mid P + P \mid P \parallel P \mid A(\tilde{x}) \\ \alpha := & a?(x) \mid a!(x) \mid \tau \end{aligned}$$

Donde a es un canal, x es un valor de un dato, \tilde{x} es una secuencia de valores y 0 denota un proceso vacío. Cada proceso puede ser expresado por una acción atómica α , compuesta (en paralelo “ \parallel ” o escogencia “ $+$ ”) por otros procesos. Las acciones atómicas están dadas por entradas $a?(x)$, salidas $a!(x)$ o simplemente acciones internas τ . Un trabajo similar, donde se emplea el álgebra CSS se presenta en (Gwen et al, 2004).

Hongli et al (2006) proponen un pequeño lenguaje denominado CDL como un modelo formal simplificado del WSCDL. CDL incluye conceptos importantes relacionados con los roles participantes y la colaboración entre estos a fin de lograr una coreografía. No obstante, enfatiza en las estructuras de control (secuencia, paralelo y escogencia) que describen las actividades y no en las interacciones y sus patrones de representación.

Las interacciones, interés de este documento, han sido formalizadas por Busi et al (2005) a través del diseño de un lenguaje simple de coreografía, el CL (Choreography Language). Este lenguaje representa el primer paso para la definición de un marco de trabajo para el análisis, diseño y desarrollo de coreografías.

La definición que brinda el CL se compone de una parte declarativa y una conversacional. La primera se encarga de la definición de los roles incluidos por los participantes en la conversación. La parte conversacional describe el orden de las interacciones entre los roles.

CL se fundamenta en los conceptos de roles e interacciones. Los primeros representan el comportamiento que un participante exhibe a fin de totalizar las actividades definidas por la coreografía. Mientras las interacciones enfatizan sobre el intercambio de información entre los roles.

Cada rol puede almacenar variables y exhibir operaciones. A cada rol se asocia un grupo de variables, las cuales representan el estado interno de cada rol, que serán usados por las interacciones. Las operaciones de un rol p son esencialmente el punto de acceso empleado por otros roles para interactuar con p . Las operaciones pueden ser una de las siguientes modalidades o patrones de interacciones: petición o petición-respuesta. La sintaxis empleada en este lenguaje es la siguiente:

Sea Var un conjunto de variables representadas por x, y, z, k . Sean \tilde{x} y \tilde{y} duplas de variables, que se puedan expresar por ejemplo como $\tilde{x} = (x_1, x_2, x_3, \dots, x_n)$.

Sea OpName un conjunto de operaciones representadas por o . Así mismo $\text{OpType} = \{\text{ow}, \text{rr}\}$ es un conjunto de tipos de operaciones, donde ow representa one-way (Petición) y rr denota la interacción request-respond (petición-respuesta). Por tanto, una operación es descrita por un nombre y un tipo de operación, como sigue.

$$\text{Op} = \{ \{o, t\} \mid o \in \text{OpName}, t \in \text{OpType} \}$$

Un rol es descrito por un nombre, un conjunto de operaciones ω que exhibe y un conjunto de variables Var . Es decir, RName es un conjunto de roles, representados por ρ , el conjunto Role contiene todos los posibles roles y está definido por

$$\text{Role} = \{ (\rho, \omega, V) \mid \rho \in \text{RName}, \omega \in \{\text{Op}\}, V \in \text{P}(\text{Var}) \}$$

Una vez se definen los componentes que conforman la parte declarativa de la coreografía, se deben definir los componentes que intervienen en la parte conversacional. Dicha conversación entre roles enfatiza en los patrones de interacciones, una para expresar una operación de petición y la otra una operación de petición-respuesta.

CL permite componer interacciones aprovechando los operadores de control que se pueden aplicar, como son de secuencia, paralelo y de opción. Formalmente, CL es un conjunto de conversaciones definidas como sigue.

$$\begin{aligned} C := & 0 \\ | & \text{OW}(\rho A, \rho B, o, \tilde{x}) \\ | & \text{RR}(\rho A, \rho B, o, \tilde{x}, \tilde{y}, \tilde{z}, \tilde{k}, C) \\ | & C; C \\ | & C | C \\ | & C + C \end{aligned}$$

Una conversación puede ser una conversación terminada (0), o puede presentar un patrón de interacción petición (one-way OW) o de petición-respuesta (request-respond RR). El patrón $\text{OW}(\rho A, \rho B, o, \tilde{x})$ representa una invocación a la operación o del rol ρB ejecutada por el rol ρA . \tilde{x} y \tilde{y} , son secuencias de variables de los roles ρA y ρB , respectivamente.

La primitiva $\text{RR}(\rho A, \rho B, o, \tilde{x}, \tilde{y}, \tilde{z}, \tilde{k}, C)$ es usada para describir el patrón de operación petición-respuesta. Esta describe la invocación de la operación o del rol ρB , ejecutada por ρA . Las duplas \tilde{x} y \tilde{y} son secuencias de variables de los roles ρA y ρB , respectivamente. Posteriormente, el rol ρB ejecuta la conversación C , lo que finalmente conlleva a que el contenido de las variables \tilde{k} del rol ρB sea asignado a las variables \tilde{z} del rol ρA .

Finalmente, las conversaciones pueden ser secuenciales ($C; C$), en paralelo ($C | C$) o en composición u opción ($C + C$).

5. CONCLUSIONES Y TRABAJO FUTURO

La coreografía de servicios web es posible describirla en varios niveles de abstracción, por así decirlo, ya que puede gestarse con la definición de sus componentes y la secuencia de las interacciones que se presentan entre estos. Así mismo es posible manipular la estructura de los datos que son intercambiados y así generar una coreografía con mayor nivel de expresividad.

Dicha expresividad permite un control sobre la información intercambiada; por tanto, es aprovechada para su aplicación en mecanismos de solución como la verificación de restricciones, en donde la definición de restricciones sobre la información o propiedades de esta facilita su manipulación y, por ende, soluciones hacia su automatización.

Un análisis y formalización de las interacciones es necesario para determinar qué componentes pueden ser manejados en un posible modelo de restricciones a satisfacer. Las interacciones deben comprenderse dentro un grupo definido (petición, petición-respuesta) y así restringir expresividad y solución.

A esto se suma que un mecanismo apropiado para la formalización de las interacciones y los procesos que representan debe ser definido con el fin de maximizar el grado de manipulación orientado a lograr un grado de automatización en la descripción de una coreografía.

Como trabajo futuro, se pretende proponer una formalización de condiciones o restricciones sobre la estructura conversacional y declarativa de la coreografía de servicios web, a fin de lograr una manipulación automática de estas.

REFERENCIAS

- AIELLO M., and PAPAZOGLOU M., 2002. A Request Language for Web-Services Based on Planning and Constraint Satisfaction. Workshop on Technologies for E-Services (TES02).
- ANDREWS T., et.al., 2003. Business Process Execution Language for Web Services, version 1.1, Disponible en: <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel>.
- APPEL, K. and HAKEN, W., 1977. Every Planar Map is Four Colorable. Part ii. Reducibility. Illinois J. Math.
- BARBER, F., SALIDO, M., 2003. Introducción a la Programación de Restricciones. Revista Iberoamericana de Inteligencia Artificial AEPIA. N° 20.
- BARROS, A., DUMAS, M. and OAKS, P. 2005. A critical overview of the web services choreography description language. <http://www.bptrends.com>
- BARTÁK, R., 1999. Constraint Programming: In Pursuit of the Holy Grail, Week of Doctoral Students (WDS99), Prague, June.
- BENATALLAH, B., DUMAS, M., SHENG, Q.Z. and NGU, A. 2002. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. International IEEE Conference on Data Engineering, San Jose, USA, February.
- BERNERS-LEE, T. HENDLER, J and LASSILA, O, 2001. The Semantic Web. Scientific American.com. May, 2001.
- BERGSTRA, J., PONSE, A. and SMOLKA, A., Editors, 2001. Handbook of Process Algebra. Elsevier.
- BOOCH, G. 1998. The unified modeling language user guide.. Reading (Massachusetts). Adison Wesley 482pp.
- BROGI, A., CANAL, C., PIMENTEL, E., and VALLECILLO, A., 2004. Formalizing Web Service Coreographies. Electronic Notes in Theoretical Computer Science.
- BURDETT, D. and KAVANTZAS, N., 2003. Net Traveler: A Framework for Automatic Web Services Collaboration, Orchestration and Choreography in E-Government Information System. Universidad de Puerto Rico.
- BUSI, N., GORRIERI, R., GUIDI, R., LUCCHI, R. and ZAVATTARO, G., 2005. Towards a formal framework for choreography. IEEE Computer Society, Junio.
- CANAL, C., PIMENTEL, E. and TROYA, J. M., 2001. Compatibility and inheritance in software architectures, Science of Computer Programming.
- CHANNA, N., CAMPING, L., WASIM, A. and XIANGJUN, F., 2005. Constraint Satisfaction in Dynamic Web Services Composition. Asian Journal of Information Technology, 4 (10): 957-961.
- CERNUDA DEL RÍO, A., 2002. Sistema de Verificación de Componentes Software. Tesis Doctoral. Universidad de Oviedo, Departamento de Informática.
- CERNUDA DEL RÍO, A., 2002. Static Validation of BusinessProcess Compatibility in Web Services Choreographies. Universidad de Oviedo, Departamento de Informática.
- CHOW, K., PERETT, M., 1997. Airport Counter Allocation Using Constraint Logic Programming. Practical Application of Constraint Technology.
- CONSOLE, L., TORASSO, P., 1991. A Spectrum of Logical Definitions of Model-based en Diagnosis. Computacional Intelligence.
- DECHTER, R. 2003. Constraint Processing. Morgan Kauffman.
- FALTINGS, B., and YOKOO, M., 2005. Introduction: Special Issue on Distributed Constraint Satisfaction. Artificial Intelligence.
- FOSTER, H., 2003. LTSA-BPEL4WS Tool. Department of Computing, Imperial College London. Available to download at <http://www.doc.ic.ac.uk/ltsa/bpel4ws>.
- FOSTER, H., UCHITEL, S. and MAGEE J., 2004. Compatibility Verification for Web Services Choreography. Imperial College London.
- FOSTER, H., UCHITEL, S., MAGEE, J. and KRAMER, J. 2006. Model-Based Analysis of Obligations in Web Service Choreography. Imperial College London.

- GWEN, S. BOURDEAUX, L. and SCHAEERF, M., 2004. Describing and reasoning on web services using process algebra. In proc. 2nd Internacional Conference on Web Services. Published by IEEE, July, page(s) 43-50.
- HONGLI, Y., XIANGPENG, Z., ZONGYAN, Q., GEGUANG, P. and SHULING, W., 2006. A formal Model for web services choreography description language (WS-CDL). School of Mathematical Science. Peking University.
- IBM, 2001. Web Services Flow Language. Accedida en Noviembre 2005. www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.
- JANSEN, E., YANG, H. and HELAL, S., 2005. Automated Verification of Proper Choreography Implementation. Universidad de Florida.
- KAVANTZAS, N., BURDET, D. and RITZINGER, G. 2004. Web Services Choreography Description Languages Version 1.0. W3C Working Draft. Diciembre.
- KIMA, K., WOO KIMA, K., HWANGB, H., C. Ko., 2004. Operator-scheduling Using a Constraint Satisfaction Technique in Port Container Terminals. *Computers & Industrial Engineering* 46.
- KUNO, H. et.al., 2001. *Conversations + Interfaces = Business Logic*. Springer.
- LARROSA, J. and MESEGUE P., 2003. Algoritmos para Satisfacción de Restricciones. *Revista Iberoamericana de Inteligencia Artificial AEPIA*. N° 20.
- LAZOVIC, A., AIELLO, M. and GENNARI, R., 2005. Encoding request to web services composition as constraints. In Int. Conf. on Principles and Practice of Constraint Programming (CP2005), LNCS 3709, pages 782- 786. Springer.
- MARTIN, D. and BURSTEIN, M. 2001. OWL-S: Semantic Markup for Web Services. Disponible en <http://www.daml.org/services/owl-s/1.1>
- MCKINLEY, P., MALENFANT, A. and ARANGO, J. 1999. Pavilion: A Middleware Framework for Collaborative Web-based Applications. ACM SIGGROUP conference on Supporting group work, November.
- MEDERITH, G. and BJORG, S., 2003. Contracts and types. *Communications of the ACM*, 46(10):41-47.
- MENDLING, J., HAFNER, M. 2005. From Inter-Organizational Workfows to Process Execution: Generating BPEL from WS-CDL.
- MICROSOFT CORPORATION, 2001. Xlang Web Services for Business Process Design. Accedida en Noviembre 2005. www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm.
- MILNER, R., 1989. *Communication and Concurrency*. Prentice Hall.
- MORRIS, P., 1993. The Breakout Method for Escaping from Local Minima. Morgan Kauffman, AAAI.
- PAPAZOGLOU, M. and GEORGAKOPOULOS, D., 2003. Service Oriented Computing. *Communications of the ACM*, October.
- PAPAZOGLOU, M. and YANG, J. 2002. Design Methodology for Web Services and Business Processes. 3rd VLDB-TES Workshop, Hong-Kong.
- PARROW, J, 2001. An introduction to the p-calculus. *Handbook of process algebra*, Elsevier 479-543.
- PELTZ, C, 2003. *Web Services Orchestration and Choreography*. Hewlett-Packard Company. IEEE Computer Society.
- PIECHOWIAK, S., RODRIGUEZ, J., 2005. The Localización and Correction of Errors in Models: A Constraint-Based Approach. *Applied Inteligence* 23, Diciembre.
- PUHLMANN, F. and WESKE, M., 2005. Using the p-calculus for formalizing workflow patterns. *BMP2005*, volume 3649 of *Lectures Notes in Computer Science*, Berlin, Springer-Verlag, 156-168.
- RAO, J., 2003. A Survey of Automated Web Services Composition Methods. Department of computer and information science. Norwegian University of science and technology.
- RUSSELL, R. and NORVIG, P. 2004 *Inteligencia Artificial. Un enfoque Moderno*. Segunda Edición. Pearson Education, S.A, Madrid.
- SZYPERSKI, C., 1997. *Component Software – Beyond Object-Oriented Programming*. Addison-Wesley.
- VALLECILLO, A., HERNÁNDEZ, J. and TROYA, J., 2000. New issues in object interoperability. *Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg.

- W3C. 2001. Web Service Choreography Interface (WSCI) version 1.0. Disponible en <http://www.w3.org/2003/01/wscwg-charter>.
- W3C, 2002. Web Services Conversation Language (WSCL) 1.0. Disponible en <http://www.w3.org/TR/2002/NOTE-wscl10-20020304/>.
- WOMBACHER, A., FANKHAUSER, P. and MAHLEKO, B., 2004. Matchmaking for Business Processes Based on Choreographies. 14 International Journal of Web Services Research, diciembre.
- YANG, J., and PAPAZOGLOU, M., 2002. Web Components: A Substrate for Web Services Reuse and Composition. 14 Conferencia Internacional sobre Ingeniería de Sistemas de Información. CaiSE02.