

Técnicas de Inteligencia Artificial aplicadas a la Coreografía de Servicios Web

Artificial Intelligence applied to Web Services

Jorge E. Giraldo P., Ing., Jaime A. Guzmán L., MSc. y Demetrio A. Ovalle, PhD.
Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia - Sede Medellín, Colombia
jegiraldp,jaguzman,dovalle@unal.edu.co

Recibido para revisión 15 de Marzo de 2008, Aceptado 19 de MAyo de 2008, Versión final 28 de Mayo de 2008

Resumen—Este artículo presenta una aproximación al modelado de la Coreografía de Servicios Web basado en Restricciones con el fin de lograr cierto grado de automatización en su definición. Se plantea la definición tanto desde el punto de vista estructural como dinámico de los Componentes de la Coreografía de Servicios Web, con el fin de realizar su posterior representación en términos de un Problema de Verificación de Restricciones. Se presentan los resultados de rendimiento del proceso comparado con otras aproximaciones que demuestran que el modelo es apropiado para lograr automatización en la descripción del Proceso de Coreografía de Servicios.

Palabras Clave—Servicios Web, Coreografía de Servicios Web, Tecnologías Web, Verificación de Restricciones.

Abstract— This article presents an approach to modeling the Choreography the Web Services based constraints in order to achieve a certain degree of automation in its definition. This propose the definition from the point of view structural and dynamic compo-nents of the Web Services Choreography, with a view to their subsequent representation in terms of a Problem Verification Restrictions. In the work describe in this paper, experiments for analyse el performace compared with other approaches, our experiments show that the model is appropriate to achieve automation in the description Process Choreography Services.

Keywords—Web Services, Web Services Choreography, Web Technology, Constraint Satisfaction.

I. INTRODUCCIÓN

El desarrollo de software orientado a componentes se caracteriza por la tendencia de generar paquetes modulares de software, independientes de la plataforma denominados Servicios Web (SW). Su aporte principal es

proveer funcionalidades de negocio a otras aplicaciones a través de internet [1]. Bajo el enfoque web, los Servicios desempeñan distintos roles dentro del proceso de la lógica de un negocio (SOC) [2], la interacción y envío de información a través de estos se conoce como Coordinación de Servicios. Últimamente se emplean dos conceptos complementarios entre si que permiten diseñar una Coordinación de Servicios Web, estos son la Coreografía y Orquestación [3], siendo el primero de estos interés de este artículo.

En un proceso de Coreografía se intercambia información entre los servicios, para lograr un grado de automatización del proceso Coordinación, esta información requiere ser controlada, lo cual se puede hacer mediante Técnicas como la Planificación en Inteligencia Artificial [4], Técnicas de Búsquedas Heurísticas y en especial la Verificación de Restricciones [5]. Se pretende entonces modelar la Coreografía de Servicios Web con base en Restricciones definidas sobre su información.

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 hace una introducción a la temática de los Problemas de Verificación de Restricciones. Luego en la sección 4 se exponen los antecedentes a la problemática de la automatización de la definición de la Coreografía. En la Sección 5 se presenta la metodología de trabajo propuesta así como los resultados obtenidos. Por último en la sección 6 y 7 se encuentran las conclusiones y la bibliografía respectivamente.

II. COREOGRAFÍA DE SERVICIOS WEB

La Coreografía de SW se encarga de identificar la secuencia válida de mensajes entre servicios o procesos de negocios que interactúan entre si. Básicamente la Coreografía con respecto

a la composición de SW es un mecanismo de diseño que pretende definir un comportamiento global de ésta a partir de comportamientos individuales que se relacionan por medio del intercambio de información y que se rigen por reglas de comportamiento [3]. Este flujo de información en algunos casos se denomina Interacciones o Conversaciones que ya se han sido tratados por otros autores [6], [7].

III. PROBLEMAS DE VERIFICACIÓN DE RESTRICCIONES

Un PVR consiste de: Un conjunto de variables; por cada variable un dominio finito de posibles valores para las variables que aborda. Por último un conjunto de restricciones que determinan el valor que las variables pueden tomar simultáneamente [8]. Una Solución a un PVR es una asignación de valores a todas las variables de manera que satisfaga todas las restricciones. Realmente los objetivos que se persiguen con el uso de restricciones en la solución de problemas se centran en encontrar: una solución, sin preferencia alguna, encontrar todas las soluciones ó una óptima, o quizás una buena solución según una función objetivo definida.

Este trabajo propone la problemática referida a la automatización de la definición del proceso de Coreografía de Servicios., ya que en la composición la definición de componentes y de sus interacciones se hace por separado y de manera manual, generando pérdida del control al incrementarse el número de servicios en los procesos de negocios.

IV. ANTECEDENTES

Los problemas de la Coreografía han sido abordados entre otros por: (McKinley, 1999) [9], (Bergstra, 2001) [10] (Benatallah, 2002) [11], (Burdett, 2003) [12], (Foster 2004) [13] y (Brogi, 2004) [14]. Estas propuestas se caracterizan por presentar una visión monolítica de la Coreografía en donde la poca expresividad en la información que la comprende impide que sea posible su manipulación automática. Una de las primeras aproximaciones al manejo de las restricciones en el ámbito de los SW es la propuesta hecha por (Aiello y Papazoglu, 2002) [15], no obstante su solución se basa en la representación y solución de la composición de servicios a partir de las peticiones hechas por el usuario bajo un ambiente de la Planificación en IA. Su relación con la Coreografía radica en que se enfatizan las peticiones basadas en restricciones como un intercambio de mensajes; estas peticiones posibilitan una descripción global del proceso de Coreografía.

Posteriormente y basado en la solución anterior, en [16] las restricciones son empleadas para la representación del problema planteado por el usuario, mediante la representación de estados de transición es posible la definición de los estados finales que de una forma u otra representan los objetivos a cumplir. En [17] se propone la aplicación de la Verificación de Restricciones al problema de la Coreografía Automática de SW. En este trabajo las restricciones son definidas mediante una Ontología de dominio, representadas en clases, atributos, instancias y relaciones, entre otras; también se propone una arquitectura compuesta por un repositorio de servicios descritos en OWL-S

en asocio con un mecanismo de descubrimiento de SW, el cual consiste en una interfaz de usuario para el manejo del repositorio con el cual es posible realizar consultas sobre las propiedades funcionales de los SW ubicados allí. Otro componente de la arquitectura es el optimizador de restricciones, diseñado para asegurar que el descubrimiento de los SW sea acorde a las peticiones hechas por el cliente, junto con el optimizador de restricciones trabaja el módulo de estimación de costo, aquí se tiene en cuenta los siguientes aspectos: tiempo de respuesta, compatibilidad, costo económico y confiabilidad, en sí este análisis gira en torno a la Calidad del Servicio.

V. METODOLOGÍA PROPUESTA Y RESULTADOS

Como primera medida se identifican y se modelan los componentes de la Coreografía (Servicios, operaciones, Roles y mensajes); con base en esta información se hace una representación en términos de un Problema de Verificación de Restricciones y luego se emplean mecanismos de programación para su manipulación automática.

A. Componentes de la Coreografía

La Coreografía de SW se caracteriza por los siguientes componentes: Servicios y los mensajes intercambiados entre estos, los Servicios representan funciones dentro del proceso de Coreografía, los mensajes son información intercambiada entre servicios. Los conceptos y componentes aquí presentados se enfocan en la definición de la Coreografía desde un punto de vista estructural y dinámico, la primera se encarga de la definición de los componentes y sus características, la parte dinámica garantiza un apropiado intercambio de información durante la composición, es decir garantiza un flujo de información representado por medio de mensajes.

Servicio Web

Un Servicio Web está definido a partir de sus entradas, salidas y la clasificación funcional; esta última es una descripción de la funcionalidad que ofrece el servicio. Las entradas y las salidas de los servicios brindan información acerca de los tipos de datos a manejar.

Operaciones:

Las operaciones definen las tareas que realiza un determinado Servicio Web. Las operaciones acarrean variables, que son los argumentos de invocación del servicio, así mismo una operación bajo un esquema conversacional soporta unos patrones de interacción; estos últimos se encargan del flujo de información intercambiada por los mensajes que se envían los servicios entre sí. Cada operación contiene un tipo de operación que indica si esta retorna o no valores, así mismo si espera que le lleguen algunos datos para su ejecución.

Mensaje:

La información intercambiada por los servicios se hace a través de mensajes, siempre existe un emisor y un receptor de un mensaje, el contenido de este se especifica por medio de las operaciones de los servicios asociados

Rol:

Como se había indicado, dentro de un Proceso de Negocio existen funcionalidades que se representan por Roles y pueden ser asociados Servicios Web capaces de realizar dichas funcionalidades. La idea es determinar el Servicio apropiado para ser asociado al Rol de acuerdo a los requerimientos especiales de intercambio de información propuesto por el Proceso de Negocio y que cobijan a los Roles.

Definición de una Coreografía:

La Coreografía se modela teniendo en cuenta los componentes estructurales como dinámicos, los primeros describen los Servicios y sus Operaciones que van a ser integrados. Por su parte la dinámica de la Coreografía hace referencia al control en el flujo de información, definir componentes que manipulen información del flujo de esta en el Proceso de Negocio. De acuerdo a lo anterior la Coreografía de Servicios Web tiene como punto de partida la definición de un Proceso de Negocio basado en un flujo de información entre Roles específicos. Cada uno de estos roles contiene características e información que representan el papel que desempeña dentro del proceso de negocio.

La información expuesta en la Figura 1 representa un diagrama de clases UML que expone la relación de los componentes que permiten la descripción de la Coreografía. Se ilustra que un Servicio Web está asociado con un solo Rol. Por su parte un Rol está definido a partir de Restricciones sobre la información que este brinda. Por último un servicio puede tener una o muchas operaciones asociadas a este y estas son incluidas en mensajes como parte de la información que este acarrea.

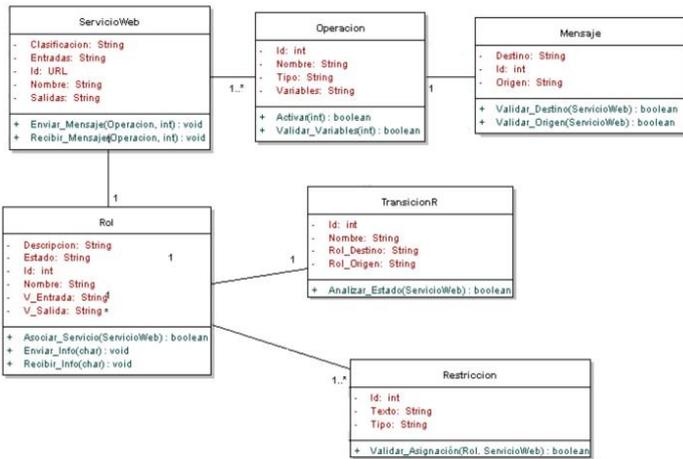


Figura 1. Modelamiento de la coreografía

La mayoría de las propuestas encontradas en la revisión bibliográfica manejan a la Coreografía como la integración de varios componentes, omitiendo el orden de intercambio de información mediante mensajes, sin embargo existen propuestas que intercambian información, pero esta es definida totalmente de manera manual, lo que en ambientes reales no sería viable, ya que se necesita que el proceso de Coreografía sea automático. De igual forma en estas propuestas de investigación el punto

de partida de la Coreografía no presenta un objetivo en común el cual se trate de cumplir, este es determinado manualmente; para el caso de esta tesis se pretende partir de un pro-ceso de negocio abstracto basado en intercambio de información entre roles específicos.

Los Estados de los Roles facilitan un control automático sobre el proceso de Coreografía, así pues cuando el proceso se encuentre en un estado significa que allí en ese estado se debe analizar, detectar sus restricciones y posteriormente realizar una búsqueda que permita encontrar aquel servicio apropiado que satisficelas. Una vez se tiene información sobre los Roles y sus Restricciones, puede darse paso a un razonamiento lógico sobre estas, que pueden ser basados en Reglas de Producción.

Estas reglas dan apoyo sustancial al proceso de búsqueda pues permiten omitir valores o servicios no apropiados para su asignación en el proceso de la Coreografía. Sin embargo se debe definir cual de los algoritmos para soluciones Problemas de Verificación de Restricciones es el más apropiado para dar solución a un problema de definición de la Coreografía de Servicios Web modelado de esta manera. Para ello en la sección siguiente se presenta una comparación de los algoritmos de solución de problemas basados en restricciones.

B. Coreografía de Servicios Web como un Problema de Verificación de Restricciones

Un Problema de Verificación de restricciones básicamente consiste en un conjunto finito de variables, un dominio de valores para cada variable y un conjunto de restricciones que acotan la combinación de valores que las variables puedan tomar. Así el objetivo es encontrar un valor para cada variable de manera que se satisficgan todas las restricciones del problema.

La Coreografía de Servicios Web define el orden en el cual un grupo de servicios intercambia in-formación a través de mensajes esto con el fin de describir desde un punto de vista global; donde se tiene en cuenta aspectos estructurales como dinámicos. Por tanto y ya que el Modelo de Verificación de Restricciones se alimenta de un proceso abstracto de conversaciones entre Roles, se quiere satisfacer las restricciones que se han definido en cada uno de estos por medio instancias de Servicios Web. De acuerdo con lo anterior, se definen los siguientes elementos que hacen posible ver la Coreografía de Servicios Web como un Problema de Verificación de Restricciones; estos son las Variables, un Dominio para cada valor y un conjunto de Restricciones.

Variables

Las variables del Problema de Verificación de Restricciones son aquellos elementos a los cuales se les debe asignar un valor específico proveniente del Dominio de valores asociado a cada uno de las variables.

Variable Id_Rol : El identificador único de un determinado rol se define como una variable del sistema verificación de restricciones, el valor específico que se busca es una descripción del servicio apropiado, según el número de restricciones que este satisfaga.

Variable $V_{Entrada}$: Se define que un Rol contiene en su definición el atributo variables de entrada representadas por tuplas de estas. Se definen como variables del sistema pues los valores a asociar son las entradas de las operaciones que ofrecen cada uno de los Servicios.

Variable V_{Salida} : Al igual que la variable anterior un Rol contiene en su definición un atributo encargado de presentar las tuplas de variables que este emite hacia otros, este atributo son las variables de salida. Los valores que se asocian a estas variables son las salidas de las operaciones de cada uno de los servicios.

Variable Tipo_Operación: La variable Tipo_Operación es información a tener en cuenta debido a que si se encuentra un servicio donde su operación tenga el mismo tipo, es posible que este servicio pueda ser asociado al Rol.

Variable Num_Variables: El número de variables manejadas por el Rol viene dada en las tuplas de estas, por tanto se debe buscar un Servicio donde sus Operaciones contengan el mismo número de Variables solicitadas por el Rol.

Dominios de los valores

Por cada una de las variables se debe determinar un valor para estas, los valores están regidos por unos dominios y estos son:

Dominio de Id_{Roles} : Los valores específicos que pueden asociarse a los identificadores únicos de los roles son los identificadores de los servicios representados en direcciones URL.

Dominio Operaciones: Este dominio abarca los valores específicos para los variables referentes a las operaciones, estas son: Tipo de Operación, $V_{Entrada}$, V_{Salida} y $Num_{Operaciones}$, la información gira en torno a la información disponible por parte de las operaciones por tanto, este dominio en general contiene las descripciones de los servicios que van a ser analizados.

Restricciones

Las Restricciones son expresiones que condicionan la asignación de un valor a una variable, este valor pertenece a un dominio específico. Se busca entonces definir restricciones que busquen equivalencia de información entre la que otorgan el Proceso de Negocio y las descripciones de los Servicios Web. Las Restricciones que se definen son: Restricciones de Tipo de Operación, de Número de Variables, de Variables de Entrada, de Salida y de Existencia del Servicio.

La primera restricción expresa que el Tipo de Operación propuesto por el Proceso de Negocio debe ser el mismo al que manifiesta una operación de un Servicio. La segunda define que el total de variables manejadas por el Proceso sea al del servicio analizado. La tercera y cuarta Restricción se definen sobre la información de las Variables de entrada y de salida respectivamente. Por último se encuentra la Restricción donde se plantea que para todos los Roles debe existir un valor asociado, esto garantiza que se asocie un Servicio a Rol,

dependiendo de la información que ofrece según atributos.

Para dar solución al problema planteado se realizó un estudio comparativo de los algoritmos de búsqueda apropiados para los problemas de Verificación de Restricciones. Con el fin de determinar cual de los algoritmos es apropiado para su aplicación en problemas bajo ambientes web se llevó a cabo un experimento relacionado con subastas electrónicas, donde es posible tener uno o varios agentes de bolsa comparando productos y a su vez otros vendiéndolos. Es así que a medida que se incrementan los agentes se incrementan las Restricciones pues existe más información en el Proceso.

El ejercicio se inicia con veinte posibles Servicios Web a analizar, equivalentes a las instancias o nombres de los agentes de bolsa; las Restricciones se definen como que ningún servicio sea igual a otro. El problema se configura como sigue: la primera configuración se compone de 7 variables y 9 restricciones ($V=7$ $R=9$), es decir 7 servicios y 9 restricciones entre estos, la segunda de 14 variables y 19 restricciones ($V=14$ $R=19$) y al tercera de 21 variables y 29 restricciones ($V=21$ $R=29$), relativamente son pocos los componentes de cada problema, pero la búsqueda de la solución sufre una explosión combinatoria que los convierten en problemas poco tratables, por tanto el algoritmo debe soportar dichas situaciones.

Para efectos de medición del tiempo se realizaron los experimentos en un equipo de cómputo con las siguientes características: Procesador Pentium 4 de 3 GHz y 1GB de memoria, empleando la API de Java especial para el manejo de algoritmos de Verificación de Restricciones, esta es la JCL (Java Constraints Library) [20]. Los puntos a evaluar en cada una de las configuraciones del problema ($V=7$ $R=9$), ($V=14$ $R=19$), ($V=21$ $R=29$) son el número variables asignadas, el número de soluciones encontradas y el tiempo en milisegundos que toma para encontrar todas las soluciones. Los algoritmos a evaluar son: Backtracking (BT), Arco-consistencia (AC), Backmarking (BM), Backjumping (BJ), Forward Checking (FC), Backjumping dirigido por conflictos (BJ DC) y Backjumping basado en grafos (BJ BG).

En la Tabla 1 se sintetizan los resultados para ello se muestra una lista de chequeo con las características que cumple cada uno de las técnicas de solución. Para su calificación se considera apropiado (A) o no apropiado (NA), en el caso de medidas específicas se presentan su equivalencia. En la tabla se tienen en cuenta los siguientes aspectos:

Instanciar una variable: Se denomina del mismo modo hacer una asignación; esto quiere decir dar-le un valor a una variable que se sepa que satisface una restricción. Un algoritmo se considera apropiado si que frente a problemas incrementales y cambiantes procura realizar el menor número de asignaciones.

Descubrimiento de soluciones: Después determinar el proceso de búsqueda es posible que halla encontrado más de una combinación donde efectivamente se satisfagan las restricciones. Así pues es posible que sea esto favorable al momento de tener varias opciones por elegir, pero la idea es

tomar aquel servicio más apropiado dependiendo del nivel de interacción que acareé el proceso. Número de soluciones encontradas bajo ambientes cambiantes como la web.

Tiempo de ejecución: Se mide en milisegundos, determina el tiempo en el cual encuentra las soluciones existentes para el problema o también cuando determina que no existe solución alguna. Es una medida para eficiencia del algoritmo.

Manejo de información: Determina si el algoritmo elimina valores del dominio definido para un grupo de variables. Este aspecto es importante, pues a medida que no sea necesario omitir valores del dominio, sino considerarlos para funciones requeridas dentro de mismo proceso garantiza la integridad de los datos.

Tabla 1. Resumen de la diferencia de medias entre los diferentes grupos

Algoritmo	Instanciación de Variables	Descubrimiento de soluciones	Manejo de información	Tiempo de ejecución
BT	NA	A	A	NA
AC	NA	A	NA	NA
BJ Dirigido/Conflictos	A	NA	A	A
BJ Basado/Grafos	A	A	A	NA
BM	NA	A	A	NA
BJ	NA	A	A	NA
FC	A	A	A	A

De los algoritmos de búsqueda el recomendado para el manejo automático de la Coreografía de Servicios Web basada en restricciones es el Forward Checking - FC - pues este hace un chequeo hacia adelante permitiendo que los valores subsecuentes y anteriores puedan ser tenidos en cuenta y así generar etiquetas de marcado que permitan regresar a estos. Aunque el FC declara dentro de su procedimiento una eliminación de valores del dominio que sean inconsistentes, esta eliminación es temporal, permitiendo posteriormente emplear los valores.

C. Resultado Obtenidos, Comparación con otros Modelos

La validación realizada al modelo planteado se basó en algunos aspectos orientados a medir el rendimiento del proceso de Definición Coreografía de Servicios, estos son: Escalabilidad, Complejidad en el proceso, Número de asignaciones, Falsos positivos, Falsos Negativos.

En la Figura 2 se ilustra el criterio de Escalabilidad, donde se varía el número de Servicios y se mide el tiempo del proceso de Coreografía, los Servicios varían de 3 en 3 desde 6 hasta 12 servicios. El eje x representa el número de servicios y el eje y el tiempo en milisegundos. De la figura se puede concluir que el comportamiento del sistema es apropiado a medida que se incrementan los servicios.

La complejidad es medida en la Figura 3, para ello se varían el número de Restricciones, donde su incremento varía de 3 en 3 partiendo desde 6 hasta 18 Restricciones. El eje x representa el número de Restricciones y el eje y el tiempo en milisegundos.

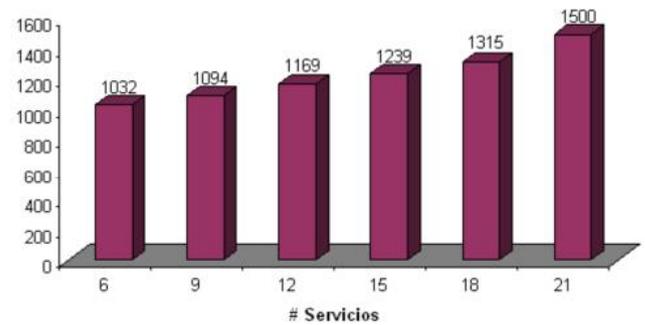


Figura 2. Escalabilidad

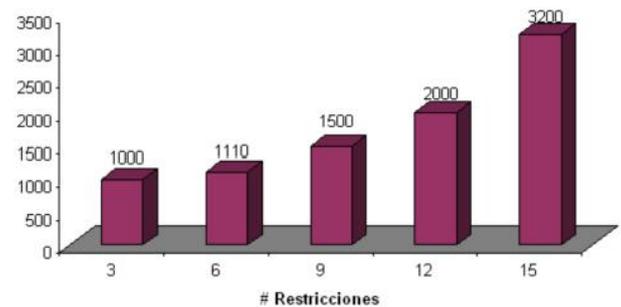


Figura 3. Complejidad

De la figura anterior se concluye que la complejidad se incrementa desde el punto de vista del tiempo cuando se incluyen más de 6 Restricciones en cada uno de los Roles, para ello es recomendable por ahora emplear solamente 4 Restricciones.

Por su parte en la Figura 4 se exponen los Falsos Positivos así como los Negativos. Para los Positivos se observa que se alcanzan a asociar servicios que no tienen mucha relevancia para la solución de la Coreografía y es directamente proporcional al incremento del número de servicios. Por el contrario los Falsos Negativos se mantiene de cierta manera constantes a medida que incrementan los servicios.

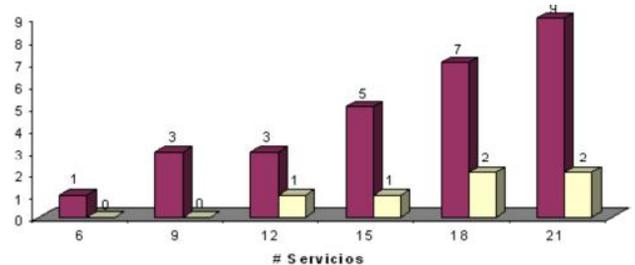


Figura 4. Falsos Positivos y Falsos Negativos

El algoritmo de Forward Checking apoyado con el uso de reglas de razonamiento agiliza la asociación de servicios a Roles que requieran la funcionalidad que estos ofrecen. Sin embargo es necesario optimizar la recuperación de servicios, ya que el aspecto de los falsos positivos y negativos es un riesgo que puede ser mitigado mediante el apoyo de mecanismos de razonamiento semántico.

VI. CONCLUSIONES

La conclusión principal de este trabajo de investigación es que el Modelo de Verificación de Restricciones, ofrece mejores prestaciones para que la definición de un proceso de Coreografía se pueda automatizar, esto teniendo en cuenta únicamente el descubrimiento del servicio, omitiendo la calidad de la información.

De las mediciones hechas se puede demostrar que la complejidad demora un poco más de tiempo con respecto a otros modelos, no obstante su precisión mejora, en referencia a los Falsos Negativos y Falsos positivos. Lo anterior concuerda con el hecho que se tuvo en cuenta el descubrimiento más no la calidad de la información del servicio.

El intercambio de mensajes es clave para el modelamiento de la Coordinación de servicios, por lo cual la Coreografía de Servicios Web debe ser descrita desde un punto de vista tanto estructural como dinámico. La Vista Estructural debe encargarse de la definición de los Servicios y sus operaciones, por su parte la Vista Dinámica, permite el control del flujo de la Información por medio del intercambio de mensajes y el uso de los Estados de Roles.

Como trabajos futuros en el área se debe profundizar en los mecanismos de interacción y su interoperatividad semántica. Para ello es necesario buscar aplicaciones donde sea posible intercambiar información de manera masiva y así se evalúe la calidad de la información intercambiada

REFERENCIAS

- [1] B. Srivastava y J. Koehler. Web Services Composition – Current Solutions and Open Problems. IBM research laboratory. 2003.
- [2] Papazoglou, M. Service oriented computing: Concepts, Characteristics and Directions. Tilburg University, IN-FOLAB. Dept. of Information System and Manegement. 2003.
- [3] Peltz, C. Web Services Orchestration and Choreography. Hewlett-Packard Company. IEEE Computer Society. 2003.
- [4] Peer, J. Web services Composition as AI planning, A Survey. Second revised version, March 22, 2005.
- [5] Rich, E. y Knight, K. Inteligencia artificial. Segunda edición. McGrawHill. 1994.
- [6] Hull, R., Benedikt, M., Christophides, D. y Su, J. E-services: A look behind the curtain. In Proceedings of ACM Symposium on Principles of Database Systems. ACM, 2003.
- [7] Chen, L., Wassermann, B., Emmerich, W. y Foster, H. Service Orchestration with BPEL, IEEE International Conference on Software Engineering (ICSE 2006), Shanghai, China, May 2006.
- [8] Russell, R. and Norvig, P. Inteligencia Artificial. Un enfoque Moderno. Segunda Edición. Pearson Education, S.A, Madrid. 2004.
- [9] McKinley, P., Malenfant, A. y Arango, J. Pavilion: a middleware framework for collaborative Web-based applications. Proceedings of the international ACM SIGGROUP conference on Supporting group work, November 1999.
- [10] Bergstra, J., Ponse, A. and Smolka, A., Editors. Hand-book of Process Algebra. Elsevier. 2001.
- [11] Benatallah, B., Dumas, M., Sheng, Q.Z. and Ngu, “A. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services”. International IEEE Conference on Data Engineering, San Jose, USA. 2002.
- [12] Burdett, D. and Kavantzaz, N. “Net Traveler: A Framework for Automatic Web Services Collaboration, Orchestration and Choreography in E-Government Information System”. Universidad de Puerto Rico. 2003.
- [13] Foster, H., Uchitel, S. and J. Magee. Compatibility Verification for Web Services Choreography. IEEE ICWS 2004, San Diego, CA. 2004.
- [14] Brogui, A. and Canal, C. “Formalizing Web Service Coreographies”. Notes in Theoretical Computer Science 105, pages 73-94. 2004.
- [15] Aiello, M., Papazoglou, M. A Request Language for Web-Services Based on Planning and Constraint Satisfaction. Workshop on Technologies for E-Services (TES02), 2002
- [16] Lazovik, A., Aiello, M. And Gennari, R. Choreographies: using Constraints to Satisfy Service Requests. In IEEE Web Services-based Systems and Applications (WEBSA at ICIW), 2006.
- [17] Channa, N., Camping, L., Wasim, A. and Xiangjun, F. Constraint Satisfaction in Dynamic Web Services Composition. Asian Journal of Information Technology, 4 (10): 957-961, 2005.
- [18] Aggarwal, R., Verma, K., Miller, J. and Milnor, W. Constraint Driven Web Service Composition in METEOR-S. Services Computing, 2004 IEEE International Conference on (SCC'04), 2004.

Jorge Eliécer Giraldo Plaza: Ingeniero de Sistemas, Universidad Francisco José de Caldas, 2004. Magister en Ingeniería-Ingeniería de Sistemas de la Universidad Nacional de Colombia, sede Medellín, 2008.

Jaime Alberto Guzmán Luna: Ingeniero Civil, Universidad Nacional de Colombia, sede Medellín, 1990. Magister en Ingeniería-Ingeniería de Sistemas de la Universidad Nacional de Colombia, sede Medellín, 1997. Especialista en Comunicación Educativa, Universidad de Pamplona, 2000.

Demetrio Arturo Ovalle Carranza: Ingeniero de sistemas y computación, Universidad de los Andes. Magister en Informática, Institut National Polytechnique de Grenoble, INPG, Francia, 1987. Doctor en Informática, Université Joseph Fourier, UJF, Francia, 1991.