

# Sistema de Intermediación para Comercio Electrónico B2B basado en Servicios Web

Giner Alor Hernández <sup>1</sup>, José Oscar Olmedo Aguirre <sup>1</sup>

<sup>1</sup>Centro de Investigación y de Estudios Avanzados del IPN. Departamento de Ingeniería Eléctrica. Sección de Computación. Av. Instituto Politécnico Nacional 2508, Col San Pedro Zacatenco. 07360 México, D. F.  
e-mail: ginerador@computacion.cs.cinvestav.mx, oolmedo@delta.cs.cinvestav.mx

**Resumen.** La implementación de sistemas B2B (*Business-to-Business*) basados en la Web ha producido un beneficio considerable en el costo, la rapidez y la eficiencia de las operaciones comerciales de empresas con base tecnológica. Entre los problemas fundamentales de estos sistemas, se encuentra la integración, colaboración y operación eficiente de proveedores en la cadena de suministro. Para el desarrollo de una cadena de suministro de nivel global, organizaciones, consorcios y grandes empresas han desarrollado UDDI, un servicio de búsqueda y recuperación de información sobre negocios. Desafortunadamente, UDDI no define reglas para especificar y conducir los procesos administrativos necesarios para la integración de organizaciones ya que está enfocado principalmente al descubrimiento de servicios que ofrecen las empresas. Para resolver ésta falta de integración, en éste trabajo se propone un sistema de intermediación capaz de optimizar la cadena de suministro al descubrir e invocar de manera automatizada los servicios Web que ofrecen diversos proveedores. Además, se describe el prototipo experimental de un escenario B2B que ha sido desarrollado con el fin de demostrar la funcionalidad del sistema de intermediación.

## 1 Introducción

Los primeros sistemas de intermediación de comercio electrónico como MARI [1], OFFER [2], Kasbah [3], MAGMA [4] y Jango [5] se desarrollaron para llevar a cabo la integración de empresas mediante la creación de mercados virtuales con el fin de optimizar la cadena de suministro. Estos sistemas usan un modelo relativamente simple, en donde la única información disponible para los agentes es aquella que se encuentra públicamente disponible en la Web. Este modelo impone serias dificultades para recuperar información con contenido semántico debido a las limitaciones inherentes de la tecnología HTML que predomina en la Web. Puesto que no sería realista esperar un cambio global en la Web hacia documentos descritos con tecnologías más avanzadas como XML, la reorganización de empresas entorno a la intermediación podría dar lugar a mejores prácticas de comercio electrónico. Como solución se propone un sistema de intermediación que permita la integración dinámica de empresas, el descubrimiento y la invocación automatizada de sus servicios usando servicios Web.

## 2 Sistema de Intermediación

Un servicio Web es esencialmente la infraestructura de comunicaciones que permite invocar un método público ofrecido por un servidor. Mediante los servicios Web es posible llevar a cabo la integración

de organizaciones comerciales ya que los servidores son independientes de la plataforma operativa, siendo necesario conocer solamente la especificación del servicio Web para realizar su invocación. Dado que el sistema de intermediación está basado en servicios Web, el sistema puede verse como un servidor *proxy* por su interoperabilidad con otros sistemas o agentes de software en Internet. Al igual que un *proxy*, el sistema recibe solicitudes de clientes, redirigiendo las solicitudes a otros sistemas y enviando las respuestas a los clientes o a otro destino. Así, el sistema de intermediación recibe en un puerto específico las solicitudes de servicio escritas como documentos XML.

## 3 Descubrimiento Automatizado de Servicios Web

El descubrimiento de los servicios Web se basa principalmente en encontrar los servicios que ofrecen las organizaciones para satisfacer ciertas necesidades. Para esto, es necesario utilizar un registro distribuido, conocido como UDDI (*Universal Description, Discovery and Integration*) [6], como mecanismo común para la publicación de las descripciones de los servicios Web. Sin embargo, UDDI presenta deficiencias como la dificultad de localizar a proveedores que ofrezcan productos y servicios que cumplan ciertas especificaciones como modelo, precio, tiempo de entrega, forma de pago y otras características técnicas. Esto se debe principalmente a que UDDI solo está enfocado a localizar servicios que los negocios ofrecen y no a

cómo estos servicios operan. Como una solución a esto, proponemos un repositorio central en donde se mantengan registrados y clasificados los negocios de las empresas, los procesos comerciales que llevan a cabo, los tipos de servicios que ofrecen y los productos que venden. Para llevar a cabo ésta clasificación de negocios, productos y servicios en el repositorio, se hace uso de ontologías ampliamente aceptadas como NAICS [7], UNSPSC [8] y RosettaNet [9]. Este repositorio puede considerarse como una extensión de UDDI en el sentido en que no solamente se registran los tipos de negocio, sino que además los tipos de servicios y productos que ofrecen las empresas. Por otra parte, el repositorio del sistema de intermediación no puede considerarse como complemento a UDDI, ya que no es interoperable con otros nodos registradores UDDI, como el de IBM [10,11], el de Microsoft [12], SAP [13] y Quick UDDI [14].

Para que una organización se registre en el repositorio, es necesario invocar el servicio Web `save_Business` y enviar en un documento XML los siguientes datos: `businessCode`, `name`, `description`, `personName`, `phone`, `email` y `discoveryURL`. El dato `businessCode` corresponde a la clasificación del negocio en la taxonomía NAICS. El sistema de intermediación registra el negocio, y devuelve un ticket (`businessKey`) con el cual, el sistema de intermediación identificará de manera única al negocio dentro del repositorio.

Después, para que una organización registre sus productos, es necesario invocar el servicio Web `save_Products` y enviar en un documento XML los siguientes datos: `businessKey` y `ProductsList`. El dato `ProductsList` es un arreglo de elementos `productCode`, en el cual se especifica la clasificación de los productos que ofrece el negocio en base a la taxonomía UNSPSC. El sistema registra la lista de productos y devuelve una lista de los productos que fueron registrados de forma exitosa.

Finalmente, para que una organización registre sus servicios, es necesario invocar en el intermediario al servicio Web `save_Services` y enviar en un documento XML los siguientes datos: `businessKey` y `ServicesList`. El dato `ServicesList` es un arreglo de elementos `serviceCode`, `description`, `accessPoint` y `overviewURL`. El elemento `serviceCode` corresponde a la clasificación del servicio en la taxonomía de RosettaNet. Los elementos `accessPoint` y `overviewURL` corresponden a las direcciones electrónicas (URLs), de dónde se debe invocar y dónde se encuentra la especificación del servicio, respectivamente. El sistema intenta registrar entonces la lista de servicios, devolviendo la lista de los servicios que fueron registrados de

forma exitosa. Estos 2 últimos elementos (`accessPoint` y `overviewURL`) son necesarios para llevar a cabo el proceso de invocación automatizada de los servicios Web que a continuación se describe.

#### 4 Invocación Automatizada de Servicios Web

Suponiendo que se tengan registrados en el repositorio los negocios, productos y sobre todo los tipos de servicios de diversas empresas, el sistema de intermediación es capaz de invocar los servicios Web. Para la invocación automatizada de servicios a partir de una solicitud, el intermediario consulta al repositorio, buscando los elementos `accessPoint` y `overviewURL` de aquéllos negocios que pueden satisfacer las condiciones de la solicitud. Los elementos resultantes de la consulta son direcciones electrónicas (URLs). Los elementos `accessPoint` indican las direcciones electrónicas dónde se deben invocar a los servicios Web. Mientras que los elementos `overviewURL` indican las direcciones electrónicas de documentos WSDL, un dialecto de XML que permite describir servicios Web. WSDL (*Web Services Description Language*) permite separar la descripción de la funcionalidad abstracta ofrecida por un servicio de los detalles concretos de una descripción de servicio, como "dónde" y "cómo" esa funcionalidad se ofrece [15,16]. La forma en que el sistema lleva a cabo la intermediación electrónica es mediante el análisis de los documentos WSDL. Con el análisis de dichos documentos, el sistema invoca de forma automática a los servicios Web, lo cual es una capacidad adicional que no se ofrece en UDDI.

Para llevar a cabo el proceso de invocación automática, el sistema analiza los documentos WSDL de cada negocio encontrado en la consulta, extrae la información necesaria de los servicios Web de los negocios y construye las solicitudes de invocación a los servicios Web de los negocios. Estos documentos XML están empaquetados en SOAP. El protocolo SOAP permite el intercambio libre y estructurado de mensajes a través de Internet, definiendo un mecanismo estándar de comunicación e intercambio de datos entre aplicaciones [16,17]. Una vez construidos los documentos XML empaquetados bajo SOAP, se envían los mensajes a los diferentes servicios Web por medio de Java WSDP (*Java Web Service*

*Developer Pack*) que permite construir, probar y desplegar servicios y aplicaciones Web. Después de enviados los mensajes, se reciben las respuestas de los diferentes servicios Web que fueron invocados. El sistema de intermediación

analiza éstas respuestas, extrae la información que le es útil y construye un documento XML el cual será la respuesta a la solicitud hecha al sistema por una organización a través de un servicio Web provisto por el sistema de intermediación, como por ejemplo: `get_ProductPrice`, `get_ProductQuantity`, `get_ProductProviders`, por mencionar solo algunos. El intermediario envía entonces la respuesta al URL donde le fue hecha la solicitud. Para esto, el sistema empaqueta el documento XML bajo SOAP y por medio de Java WSDP envía dicho documento.

Para ilustrar la funcionalidad del sistema de intermediación, se describe a continuación un escenario de una cadena de suministro que integra a varios proveedores que ya ha sido implementado en nuestro prototipo experimental.

## 5 Escenario B2B

El caso de estudio describe como el sistema facilita el descubrimiento de información de empresas permitiéndoles integrarse dinámicamente a una cadena de suministro.

Suponga el siguiente escenario:

- 1) Se tienen 3 compañías (X, Y y Z) las cuales son proveedores de Tarjetas Aceleradoras de Video, cuyo código en la taxonomía UNSPSC es 43172004.
- 2) Se tiene a una compañía W, la cual tiene una cadena de producción de ensamblaje de computadoras y necesita reabastecerse de Tarjetas Aceleradoras de Video.
- 3) Las 5 compañías han adoptado las taxonomías UNSPSC, NAICS y RosettaNet.
- 4) Ninguna de las compañías sabe de la existencia de las otras

En éste escenario, ¿cómo puede la compañía W buscar y encontrar los servicios Web de las empresas X, Y y Z, de tal forma que mantenga el reabastecimiento oportuno de suministros?

Al registrarse las empresas X, Y y Z en el repositorio deben especificar el tipo de su negocio y los productos y servicios que ofrecen. Así, la compañía W puede realizar una solicitud al sistema de intermediación mediante la invocación de algún servicio Web, como por ejemplo `get_ProductPrice`, `get_ProductQuantity`, `get_ProductDeliveryTime` o `get_ProductProviders`.

Para invocar estos servicios, solamente es necesario enviar el dato `productCode` cuyo valor es 43172004, de acuerdo a la clasificación del producto en la taxonomía UNSPSC. Después, el

sistema procesa la solicitud, formulando consultas al repositorio y construyendo solicitudes para la invocación de los servicios Web de las organizaciones encontradas en las respuestas de las consultas hechas. Posteriormente, el sistema de intermediación envía dichas solicitudes, obtiene respuestas a éstas solicitudes, extrae la información requerida y construye un documento XML como respuesta. Este documento XML lo envía empaquetado bajo SOAP a la compañía W. La respuesta contiene información correspondiente del producto (según el servicio Web invocado) y el elemento `discoveryURL`, que corresponde a la dirección electrónica de los proveedores que ofrecen tal producto, es decir, las direcciones de las empresas X, Y y Z. Mediante los servicios Web que provee el sistema de intermediación, la compañía W puede saber el precio, el tiempo de entrega o la cantidad disponible en el inventario de los almacenes de los proveedores X, Y y Z.

Una vez que decida cuál o cuáles son los proveedores que satisfacen sus necesidades comerciales, la compañía W puede solicitar al sistema de intermediación un pedido, mediante el servicio Web `get_ProductsOrder`. Para invocar este servicio es necesario enviar los datos `businessKey` y `ProductsList`. El dato `ProductsList` es una estructura (array) con los elementos `productCode` y `quantity`, que corresponde al producto y la cantidad a ordenar. El sistema toma el pedido, lo almacena en el repositorio y devuelve un ticket (`orderKey`), el cual identifica de manera única al pedido. Para procesar el pedido, es necesario solicitar confirmación al sistema de intermediación. Esto se realiza mediante la invocación al servicio Web `confirm_ProductsOrder`. Para invocar este servicio, solo se envía el dato `orderKey` y el sistema procesa dicho pedido de manera análoga a una solicitud `get_ProductPrice`, `get_ProductQuantity`, `get_ProductDeliveryTime` o `get_ProductProviders`. De esta forma, usando el sistema de intermediación la compañía W resuelve el problema de reabastecimiento del producto A. En la Fig. 1 se describe la funcionalidad y organización del sistema de intermediación.

En la cadena de suministro, el sistema de intermediación es de gran ayuda a la compañía W en la búsqueda, localización e invocación de los servicios Web. Ya que generalmente la compañía W intenta maximizar su cadena de producción, es necesario que su línea de producción no se detenga para lo cual necesita el reabastecimiento oportuno de productos.

Finalmente, cabe mencionar que el sistema de intermediación utiliza un modelo basado en reglas. Aunque actualmente las reglas están programadas



- Information Systems, Vol-ume 7, Number 4, pp. 315-341.
3. A. Chavez y P. Maes. "Kasbah: An agent marketplace for buying and selling goods." In First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, pp 75-90,1996.
  4. M. Tsvetovatyy, M. Gini, B. Mobasher, and Z. Wieckowski. MAGMA: An agent-based virtual market for electronic commerce. *Journal of Applied Artificial Intelligence*, (6), 1997..
  5. R. Guttman y P. Maes. "Agent-mediated integrative negotiation for retail electronic commerce." In Pablo Noriega and Carlos Sierra, editors, *Proceedings of the 1st International Workshop on Agent Mediated Electronic Commerce (AMET-98)*, volume 1571 of LNAI, pages 70-90, Berlin, May 1999.
  6. UDDI, UDDI Version 3.0, Published Specification, July 19, 2002.
  7. North American Industry Classification System, NAICS Homepage, <http://www.naics.com/>.
  8. United Nations Standard Products and Services Code, UNSPSC Homepage, <http://www.unspsc.org/>.
  9. RosettaNet, RosettaNet Homepage, <http://www.rosettanet.org/>
  10. IBM UDDI Business Production Registry, <https://uddi.ibm.com/ubr/registry.html>
  11. IBM UDDI Business Test Registry, <https://uddi.ibm.com/testregistry/registry.html>
  12. Microsoft UDDI Business Registry Node, <http://uddi.microsoft.com/default.aspx>
  13. SAP Home, <http://www.sap.com>
  14. Quick UDDI Registration Service, qUDDI Home, <http://www.quddi.com>
  15. Web Services Description Language Specification, WSDL Home, <http://www.w3.org/TR/wsd12>
  16. Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi and Sanjiva Weerawarana, "Unraveling the Web Services Web An Introduction to SOAP, WSDL, and UDDI". *IEEE Internet Computing*. April 2002. Pages. 86-93
  17. Simple Object Access Protocol Specification, SOAP Home, <http://www.w3.org/TR/SOAP/>
  18. Frank Leymann, *Web Services Flow Language (WSFL 1.0)*. IBM Academy of Technology, IBM Software Group. May 2001
  19. Kurt Christopher, *Implementing OAGIS within the Microsoft Biztalk Framework*. Biztalk and the Open Applications Group. June 14, 2000.
- Ebxml, Ebxml Business Process and Business Information Analysis Overview. Ebxml Business Process Team. May 2001.