Agentes Móviles Inteligentes para la Web Semántica

Cristian M. Mateos

Instituto de Sistemas Tandil (ISISTAN) - Facultad de Ciencias Exactas - UNCPBA
Campus Universitario - Paraje Arroyo Seco
(B7001BBO) Tandil, Bs. As., Argentina
Comisión de Investigaciones Científicas (CIC)
E-mail: cmateos@exa.unicen.edu.ar

1. Introducción

La Web de hoy ha sido diseñada principalmente para uso e interpretación por parte del humano [1]. Sin embargo, existe la necesidad de automatizar la interoperabilidad de los servicios que ofrece la Web, especialmente en aplicaciones *Business to Business* y de comercio electrónico. Generalmente, esta interoperabilidad se realiza a través de programas específicos que localizan y extraen información Web e invocan servicios accesibles vía Web. Esta alternativa resulta poco útil ya que depende del formato en el cual la información está representada (generalmente código HTML que contiene datos y presentación mezclados) y de las interfaces de los servicios (CGI, RMI, Corba, SOAP, etc.). Con el propósito de dar solución a estos problemas, las nuevas tecnologías relacionadas a la Web están apuntando a crear una Web Semántica, donde los recursos - información y servicios ofrecidos - sean descriptos de una forma no ambigua e interpretable por las computadoras.

En el escenario de la Web consistiendo de sitios que proveen contenido altamente dinámico, usuarios móviles utilizando conexiones no confiables y pequeños dispositivos tales como PDAs y teléfonos celulares, los agentes móviles tendrán un rol fundamental [2]. Un agente móvil es un programa que representa un usuario en una red de computadoras y es capaz de migrar de forma autónoma entre los sitios para realizar alguna tarea en favor de dicho usuario [3]. Esta característica es particularmente interesante cuando un agente hace uso esporádico de un recurso compartido valioso. Además, la eficiencia puede ser mejorada moviendo agentes a un sitio para consultar grandes repositorios en forma local y regresar con los resultados, evitando así múltiples interacciones con los datos a través de vínculos de red sujetos a demoras e interrupciones.

Los agentes móviles exhiben una serie de características que los hacen ideales para explotar el potencial de las redes actuales, debido a que cuentan con las propiedades de un agente convencional (reactividad, percepción, deliberación, etc.) más movilidad, que es la capacidad de transportarse entre los diferentes sitios de una red [4]. Algunas de las ventajas que ofrece el uso de agentes móviles se describen en [5,6].

Desafortunadamente, el potencial que ha demostrado la tecnología de agentes móviles se ha visto obstaculizado por las dificultades que éstos presentan a la hora de entender e interactuar con datos no estructurados. La inhabilidad de los agentes móviles de entender los conceptos presentes en una página Web o de manejar la semántica en la respuesta de un servicio Web, conllevan a la necesidad de una Web semántica en donde su contenido es descrito de acuerdo a una semántica precisa.

El presente artículo se organiza como sigue: en la siguiente sección se describirá el estado del arte en cuanto a la tecnología relacionada a la Web Semántica. En la sección 3 se describe Movilog, una plataforma de programación de agentes móviles inteligentes basados en lógica. La sección 4 presenta una posible integración de Movilog con servicios Web y servicios Web semánticos. Finalmente, en la sección 5 se exponen las conclusiones.

2. Estado del arte

A diferencia de la Web de actual, los servicios Web [8] - programas y dispositivos accesibles vía Web - pueden ser vistos como un conjunto de programas interactuando a través de una red de computadoras, sin interacción humana durante la transacción. Para que los programas puedan intercambiar datos, es necesario definir los protocolos de comunicación, la sintaxis con la cual los datos serán transferidos y el punto de la red con el cuál se establecerá la comunicación. Tales definiciones de servicios deben ser hechas de una manera rigurosa, preferentemente mediante un lenguaje interpretable por la computadora con una semántica bien definida, en contraposición con los imprecisos lenguajes naturales.

WSDL (Web Services Description Language) [9] es un lenguaje basado en XML para describir servicios Web como un conjunto de puntos en una red que operan con mensajes. Una descripción de servicio WSDL contiene una definición abstracta de un conjunto de operaciones y mensajes, un *binding* a un protocolo de comunicación concreto para dichas operaciones y mensajes, y una especificación del punto específico de la red para el *binding*.

A partir de una especificación WSDL un programa puede determinar los servicios que provee un servidor y como invocar y utilizar estos servicios, independientemente del protocolo de la red o el lenguaje de programación involucrado. Como complemento a WSDL, el WWW Consortium ha desarrollado la especificación UDDI (Universal Description, Discovery and Integration) [10]. UDDI provee un mecanismo para la publicación y búsqueda de descripciones de servicios escritas en WSDL o cualquier otro mecanismo de descripción de servicios.

Mediante el uso de las facilidades de UDDI, potenciales proveedores de servicios tales como empresas registran individualmente la información acerca de los servicios Web que ofrecen para ser usados por otras empresas. UDDI cuenta con nodos, denominados registros, que replican datos basados en una política determinada. Una vez que un proveedor se registra en un nodo UDDI, los datos son automáticamente compartidos con otros nodos y quedan disponibles a cualquiera que necesite determinar qué servicios Web son expuestos por un proveedor dado.

Otros lenguajes de descripción de contenido Web más orientados hacia la realización de una Web Semántica basada sólo en comunicación entre agentes lo constituyen el Resource Description Framework (RDF) [11] y el DARPA Agent Markup Language + Ontology Inference Layer (DAML + OIL) [12]. RDF permite ligar atributos a los recursos accesibles vía Web y a relacionar estos recursos. DAML + OIL extiende RDF con capacidades extras, permitiendo definir clases, objetos, propiedades y relaciones entre estos. Sin embargo, el problema de estos lenguajes es la necesidad de que los proveedores de servicios Web compartan una ontología estándar. Algunos avances hacia la creación de una ontología de servicios es DAML-S [1].

Recientemente, las compañías de infraestructura del E-bussiness han comenzado a anunciar plataformas que soportan cierto nivel de automatización de servicios Web. Algunos ejemplos de tales productos incluyen el e-speak de Hewlett-Packard, una plataforma para la descripción, registración y búsqueda dinámica de servicios Web; el dominio .NET y las herramientas BizTalk de Microsoft; el Framework de Servicios Dinámicos de Oracle; el Framework de Aplicaciones para E-Bussiness de IBM; y el Ambiente de Red Abierto de Sun. Algunos sitios, tales como Amazon y Google, han publicado servicios Web para la búsqueda de información mediante claves de búsqueda.

3. Movilog

Movilog [7,16] es una plataforma para la construcción de agentes móviles inteligentes, siguiendo un modelo de *movilidad fuerte* [4], en donde el estado de ejecución es transportado junto con un agente durante la migración del mismo, de forma transparente al programador del agente.

Movilog se materializa como una extensión del *framework* JavaLog [13,14], un lenguaje multi-paradigma que integra JAVA y Prolog implementado en JAVA. Movilog implementa un modelo de movilidad fuerte incorporando a JavaLog primitivas proactivas de movilidad fuerte, posibilitando la construcción de agentes móviles basados en lógica, denominados Brainlets. El motor de inferencia de Movilog está capacitado para restaurar el estado de ejecución de un Brainlet que llega a un determinado sitio a partir del estado que éste tenía antes de la migración.

Para proveer movilidad a través de diferentes sitios, cada sitio Web perteneciente a una red Movilog debe ser extendido con un MARlet (Mobile Agent Resource). Un MARlet extiende el mecanismo de *servlets* JAVA [15] encapsulando un motor de inferencia Movilog y proporcionando servicios para acceder a éste último. De esta forma, un MARlet representa un ambiente de ejecución para los Brainlets. Adicionalmente, un MARlet está capacitado para proveer servicios inteligentes bajo demanda, tales como agregar, borrar, activar o desactivar módulos lógicos, o llevar a cabo consultas lógicas. En este sentido, un MARlet puede también ser usado para proveer servicios de inferencia a agentes y aplicaciones Web externas.

Además de proveer primitivas típicas de movilidad fuerte a los Brainlets, el aspecto más novedoso de Movilog es la incorporación de la noción de *movilidad reactiva por fallas* [7], mecanismo aún no explotado por las herramientas de construcción de agentes móviles existentes en la actualidad. Mediante este mecanismo, cuando ciertos predicados especiales previamente declarados en el código de un Brainlet fallan, Movilog mueve transparentemente el estado de ejecución del Brainlet a otro sitio que tiene definiciones para ese predicado. Al arribar al nuevo sitio, el Brainlet reanuda su ejecución a partir del estado de ejecución anterior a la migración, pero haciendo uso de los predicados locales. Tales predicados especiales, en la terminología Movilog, son conocidos como los *protocolos* del Brainlet, y son los que definen el formato de las cláusulas por las cuáles el Brainlet migrará a otro sitio en caso de falla de las mismas.

El soporte de la movilidad reactiva por fallas en Movilog está dado por un cierto número de agentes estacionarios distribuidos a lo largo de la red, denominados Protocol Name Servers (PNS). Estos agentes proveen un mecanismo inteligente para migrar Brainlets automáticamente basado en sus requerimientos de recursos.

4. Integrando Movilog a la Web Semántica

Con el propósito de aprovechar las ventajas ofrecidas por los agentes móviles para la construcción de aplicaciones distribuidas citadas anteriormente, y el enorme aprovechamiento de recursos y servicios Web que propone la Web semántica, se propone extender Movilog para su integración con servicios Web.

Con el fin de materializar esta integración, se espera extender los agentes PNS mencionados en la sección anterior con la capacidad de consultar ciertos registros UDDI, interpretar documentos WSDL y mapear cláusulas Prolog a y desde servicios Web. Cuando un agente PNS detecte una falla en cierto predicado de un Brainlet, consultará un registro UDDI que le proporcionará los servicios Web que coinciden con el protocolo del predicado que causó la falla. Cuando el Brainlet trate efectivamente de accederal recurso Web, un agente PNS determinará si debe viajar al sitio remoto o no, dependiendo de ciertos factores como la carga de la red, la proximidad al sitio, el tamaño del Brainlet, etc.

Por ejemplo, considérese el servicio Web de búsqueda de información que provee Amazon, denominado *KeyWordSearchRequest*. La descripción WSDL de dicho servicio especifica sus parámetros y tipos, por ejemplo, la lista de palabras a buscar y la cantidad máxima de páginas de la respuesta. Además, el WSDL incluye una descripción del método para invocar a ese servicio. En este caso particular, el servicio debe ser invocado a través de una petición HTTP codificada mediante el protocolo SOAP al URL http://soap.amazon.com/onca/soap. Conociendo estos detalles,

podría ser posible para Movilog invocar *KeyWordSearchRequest* u otro servicio Web. El programador es quien se encargaría de especificar, con reglas Prolog sencillas, la forma en que se mapeará un determinado protocolo a la definición de un servicio Web específico.

El problema que presenta la alternativa anterior es que no tiene en cuenta la semántica de los servicios Web. Por ejemplo, el programador debe asegurar que un determinado protocolo se mapee a más de una definición de servicio Web de búsqueda de información, como puede ser *KeyWordSearchRequest* del sitio Amazon o *doGoogleSearch* del sitio Google. Una solución a este problema es la utilización de una descripción interpretable por los agentes de los conceptos involucrados en los servicios Web. Por ejemplo, un agente que conozca conceptos genéricos tales como *servicio* o *búsqueda* y materializaciones particulares de estos conceptos en implementaciones tales como *AmazonKeyWordSearchService* o *GoogleShopSearchService* podría estar capacitado para realizar una búsqueda en los sitios de Amazon y Google, sin necesidad de que el programador especifique ambas alternativas.

Con el fin de automatizar las tareas antes descriptas, Movilog debe entender el significado de los serivicios Web. Para hacer esto posible, se extenderá Movilog con manejo de ontologías, posiblemente utilizando lenguajes de descripción de contenido Web existentes, tales como RDF y DAML + OIL.

5. Conclusiones

La tecnología de agentes móviles inteligentes representa una de las áreas de investigación que presenta más desafíos debido a los diferentes factores y tecnologías involucradas en su desarrollo. La movilidad fuerte y los mecanismos de inferencia son, indudablemente, dos importantes características que una plataforma debiera proveer. Movilog representa un avance en esta dirección. La principal contribución de Movilog es el concepto de *movilidad reactiva por fallas*, que junto a la utilización de Prolog facilita el desarrollo de agentes móviles. En este sentido, la integración de Movilog con servicios Web y mecanismos de manejo de ontologías de los conceptos asociados a éstos servicios representa una poderosa herramienta para explotar la movilidad de agentes y los recursos presentes en la Web.

6. Referencias

1. S. McIlraith, T. C. Son y H. Zeng: "Semantic Web Services". En *IEEE Intelligent Systems*, Special Issue on the Semantic Web, 16(2):46-53, 2001.
2. J. Hendler: "Agents on the Semantic Web". En *IEEE Intelligent Systems*, 16(2):30-26, 2001.
3. N. Karmik y A. R. Tripathi: "Design Issues in Mobile Agent Programming Systems". En *IEEE Concurrency*, Vol. 6, Nro. 3, Julio-Septiembre de 1998.
4. A. Fuggetta, G. P. Picco, G. Vigna: "Understanding Code Mobility". En *IEEE Transactions on Software Engineering*, 24(5):342-361, 1998.
5. D. Milojicic, F. Douglis y R. Wheeler (Eds.): "Mobility - Processes, Computers and Agents". Addison-Wesley, 1999.
6. D. B. Lange, M. Oshima: "Programming and Deploying Java Mobile Agents with Aglets". Addison-Wesley, Septiembre de 1998.
7. A. Zunino, M. Campo y C. Mateos: "Simplifying Mobile Agent Development Through Reactive Mobility by Failure". En Guilherme Bittecourt (Ed.), *XVI Brazilian Symposium on Artificial Intelligence*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Noviembre 2002. En prensa.

8. S. J. Vaughan-Nichols: "Web Services: Beyond the Hype". En *Computer*, 35(2):18-21, Febrero del 2002.
9. E. Christensen, F. Curbera, G. Meredith y S. Weerawarana: "Web Services Description Language (WSDL) 1.1". *W3C Note, World Wide Web Consortium*, Marzo del 2001. URL=http://www.w3.org/TR/wsdl.
10. UDDI Technical White Paper, Septiembre del 2001. URL=http://www.uddi.org.
11. O. Lassila y R. Swick: "Resource Description Framework (RDF) Model And Syntax Specification", *W3C Recommendation, World Wide Web Consortium*, Febrero de 1999. URL=http://www.w3.org/TR/REC-rdf-syntax.
12. I. Horrocks, F. van Harmelen y P. Patel-Schneider (Eds.): "The DAML + OIL Language Specification", Marzo del 2001. Sitio web del programa *DARPA Agent Markup Language (DAML)*, URL=http://www.daml.org.
13. A. Amandi, A. Zunino y R. Iturregui: "Multi-paradigm Languages Supporting Multi-agent Development". En *Multi-agent System Engineering*. F. J. Garijo y M. Boman (Eds.). Volumen 1647 de *Lecture Notes in Computer Science*, páginas 128-139. Springer-Verlag, Berlin - Heidelberg - New York, 1999.
14. A. Zunino, L. Berdún y A. Amandi: "JavaLog: Un Lenguaje para la Programación de Agentes". *Revista Iberoamericana de Inteligencia Artificial*, (13):94-99, 2001.
15. J. Hunter y W. Crawford: "Java Servlet Programming". O'Reilly and Associates Inc., 1998.
16. C. Mateos: "Brainlets: Una Arquitectura de Software para el Soporte de Movilidad Reactiva en Sistemas de Agentes Móviles". Trabajo Final de la carrera de Ingeniería de Sistemas, Universidad del Centro de la Provincia de Buenos Aires, Julio del 2002.