

Mecanismos de mediación entre agentes inteligentes y servicios web

Cecilia Gil, Andrea Lorenzo, Guillermo Aguirre y Marcelo Errecalde¹

LIDIC. Departamento de Informática. Universidad Nacional de San Luis
Ejército de los Andes 950. (D5700HHW) - San Luis - Argentina.
{cecilia.gil15, andrealorenzogil}@gmail.com, {gaguirre, merreca}@unsl.edu.ar

Resumen Los *servicios Web* y los *agentes inteligentes* se han constituido hoy en día en dos tecnologías fundamentales para el desarrollo de aplicaciones industriales y basadas en la Web. Si bien son ampliamente reconocidas las potenciales ventajas de la interoperabilidad de ambas tecnologías, llevar a cabo tal integración no es una tarea sencilla si consideramos que cada una de ellas ha sido desarrollada tomando en cuenta diferentes estándares y especificaciones. En este trabajo, proponemos un enfoque que en lugar de considerar estas funcionalidades de integración como una facilidad disponible a los agentes individuales, las traslada al contexto multiagente mediante el uso de mecanismos de mediación y sus protocolos asociados. La factibilidad de la propuesta es determinada mediante la implementación de WSIBA, un sistema que implementa un protocolo particular de mediación (*brokering*) y un ejemplo sencillo donde se muestra de qué manera WSIBA permite el acceso a los agentes de servicios Web para acceder a información del tiempo y conversiones de temperaturas.

Keywords: sistemas multi-agente, servicios web, métodos de mediación

1. Introducción

El avance tecnológico en las comunicaciones y la evolución de nuevas plataformas para el desarrollo del software, representa un desafío para la integración de las nuevas técnicas de desarrollo de sistemas con las aplicaciones ya existentes. Esta integración debe superar una serie de diferencias para llegar a ser efectiva ya que las nuevas tecnologías se diferencian de las existentes en varios aspectos, como ser, el lenguaje de programación, el lenguaje y los protocolos de comunicación, la plataforma de desarrollo, etc. Para superar estas diferencias es necesario recurrir a métodos estandarizados y bien definidos que posibiliten la interacción entre distintas aplicaciones desarrolladas con diferentes tecnologías.

En este contexto, dos tecnologías que han ganado un creciente interés en los últimos años son los *servicios Web* (en inglés *Web services* y de ahora en más WS) y los *agentes inteligentes* (de ahora en más AI). Cada una de ellas, ha mostrado importantes desarrollos en la definición de estándares, protocolos y aplicaciones industriales, los que se han desarrollado mayoritariamente en forma independiente una de la otra.

A partir de los WS, los millones de personas que acceden a la Web para compartir información y servicios *online*, pueden ver a la Internet ya no sólo como una red de transmisión de contenido, sino que ésta se convierte en una verdadera red de ejecución de computaciones, procesamiento de transacciones comerciales y aplicaciones de

negocio. Estas tareas se tornan cada vez más habituales, lo que hace imposible que un usuario en forma manual determine, en tiempo y con eficiencia, el o los servicios requeridos para satisfacer una necesidad concreta. Por otro lado, la tecnología de AIs ha sido estudiada y perfeccionada desde hace varios años, con aplicaciones en distintas áreas de interés, desde el desarrollo de sistemas para edificios inteligentes, dispositivos móviles, conducción autónoma de vehículos, cuidado de enfermos y ancianos, sistemas recomendadores e interfaces de usuario inteligentes y control de tráfico aéreo y aplicaciones espaciales, entre otras. Sin embargo, el área donde mayores esfuerzos en investigación se están realizando es el de desarrollo de sistemas inteligentes para aplicaciones Web. En estos casos, los AI y los SMA se plantean como herramientas fundamentales para abordar los problemas de heterogeneidad de datos, distribución y dinamismo que la Web presenta, en áreas como Web Mining, Semantic Web y Web Intelligence [6,8,12].

Dada la relevancia que los WSs y los AIs han ido ganando a lo largo del tiempo, no es de extrañar que comenzaran a surgir distintas propuestas para su integración, de manera tal de obtener las mejores características y ventajas de ambas tecnologías. Éste no es sin embargo, un aspecto trivial si consideramos que ambos enfoques han sido desarrollados con diferentes estándares y especificaciones, como el caso de las propuestas de FIPA¹ para los AIs y W3C² para los WSs.

Por otro lado, los *sistemas multiagente* (SMA) se caracterizan por la complejidad y flexibilidad de las interacciones de sus agentes constituyentes, que quedan expresadas en distintos *protocolos* provistos por mecanismos tales como la *votación*, *negociación*, *argumentación*, *mediación*, etc. En todos estos casos, los protocolos especifican y formalizan la forma en que los agentes deben llevar a cabo sus interacciones para obtener un *resultado social* particular.

Nuestra propuesta consiste en trasladar la integración de AIs y WS de la visión del agente individual al contexto multiagente. Para ello, en lugar de ver esta integración como una simple facilidad provista por una componente de la plataforma de agentes, la implementamos mediante un *mecanismo de mediación* y un protocolo particular de este tipo de mecanismos (*brokering*) que permite ofrecer una visión de SMA de esta integración, con varias ventajas y potencialidad de extensión en el futuro.

La utilidad de nuestra propuesta, se visualiza claramente en aquellos sistemas multiagente que consultan o utilizan funcionalidades brindadas por los servicios web para cumplir con sus objetivos. Para que dichos sistemas se comuniquen, es necesario un protocolo de comunicación que permita traducir los requerimientos entre ambas partes de modo que exista una comunicación efectiva. En este contexto, nuestra hipótesis de trabajo es que el protocolo *brokering* como recurso para conseguir la mediación entre agentes y servicios web puede jugar un rol importante en esta integración. La propuesta es implementada en el sistema WSIBA, donde existen agentes especializados que se encargan de la comunicación y administración de los requerimientos. El agente “Broker” actúa como ente centralizador de los requerimientos que los agentes dedicados le solicitan o consultan, y luego establece la comunicación con el proveedor del servicio correspondiente. El enfoque consiste en aplicar un mecanismo de mediación a partir de estándares de comunicación como los promovidos por FIPA, respecto a la utilización de lenguajes y protocolos de comunicación como ACL (Agent Communication Language) y el *Brokering* [5] para facilitar la integración de las partes.

¹ <http://www.fipa.org>

² World Wide Web Consortium. <http://www.w3.org>

La organización del trabajo es la siguiente: en la Sección 2 se caracterizan las tecnologías de agentes y servicios web continuándose en la sección siguiente con la descripción del rol de la mediación entre aplicaciones en general y mediante el protocolo brokering en particular. La Sección 4 explica cómo se empleó el protocolo brokering de mediación en el diseño e implementación de WSIBA. La última sección presenta una breve discusión sobre las principales conclusiones obtenidas y potenciales trabajos futuros.

2. Agentes y servicios Web

Los AIs constituyen un paradigma interesante para el análisis, diseño e implementación de entidades que razonan y actúan de forma autónoma y flexible. Éstos exhiben características y comportamientos usualmente asociados a los humanos como el aprendizaje, la comunicación con otros agentes y la toma de decisiones en base a las circunstancias particulares que presenta el ambiente, intentando maximizar su desempeño.

El desarrollo de los *sistemas multiagentes* (SMA) por su parte, extiende la visión de los AIs al contexto de los sistemas inteligentes distribuidos, centrándose en el comportamiento *social* de los agentes y a los mecanismos y protocolos que permiten patrones de interacción complejos y flexibles de la sociedad de agentes como un todo. Así por ejemplo, podemos mencionar mecanismos de interacción bien conocidos en los SMAs como la votación, negociación, mediación, etc. Cada uno de estos mecanismos, proveen sus propios *protocolos de interacción* asociados, que permiten a los agentes intercambiar mensajes siguiendo una secuencia determinada entre los participantes de una conversación. A modo de ejemplo, se puede mencionar la *cuenta de Borda* y la *regla de Black* en los mecanismos de votación, la *subasta inglesa* y la de *Vickrey* en los mecanismos de subastas, los protocolos de *brokering* y *matchmaking* en los mecanismos de mediación, etc. En todos estos casos, los SMAs constituyen una solución interesante a los problemas que surgen cuando los datos y las fuentes de información se encuentran distribuidos geográficamente ya que pueden interconectarse e interoperar con otros sistemas sin tener que cambiar o reescribir el software existente.

JADE (por *Java Agent DEvelopment framework*) [1] es un software libre y *open source* implementado en Java que provee una API simple y amigable, proporcionando un marco de trabajo adecuado para el desarrollo de agentes de software. Es una plataforma que provee funcionalidades básicas para el desarrollo de SMA y aplicaciones que se ajusten a las especificaciones FIPA para agentes. Estas especificaciones son un conjunto de estándares que tienen por objeto promover la comunicación e integración de los agentes con sistemas heterogéneos y los servicios que ellos pueden prestar, dejando abierto los detalles de implementación y la arquitectura interna.

Al iniciarse la plataforma JADE, se lanzan automáticamente junto con el contenedor principal dos agentes especializados: *AMS* (*Agent Management System* - agente administrador del sistema) el cual representa la autoridad en la plataforma ya que administra el ciclo de vida de los agentes y provee el servicio de nombre, garantizando que cada agente tenga uno único. El *DF* (*Directory Facilitator* - Facilitador de Directorio) es un agente que provee el servicio de páginas amarillas. Los agentes que desean publicar sus servicios se registran en él y también pueden buscar los servicios que ellos necesitan. El agente *Sniffer* permite ver los mensajes y la comunicación entre agentes. Usa una técnica similar a los diagramas de secuencias de UML.

Una de las características más importantes de los agentes JADE es su capacidad de

comunicarse. El modelo de comunicación adoptado es el intercambio asíncrono de mensajes o actos comunicativos. Cada agente tiene una bandeja de entrada donde ingresan los mensajes enviados por otros agentes. Los mensajes que intercambian los agentes JADE tienen un formato especificado por el lenguaje estándar definido por FIPA e incluye algunos campos, los cuales son usados para administrar varias conversaciones al mismo tiempo y permiten la utilización de protocolos.

El WSIG es una componente agregada de JADE, desarrollada para conectar transparentemente la plataforma con los WS, provyendo los mecanismos de transformación apropiados y asegurando la integración sin tener que cambiar las especificaciones existentes de ambas tecnologías. Actúa como intermediario entre ambas aplicaciones, cuando un agente quiere comunicarse con un servicio web, lo hace mediante el WSIG quien administra y transforma los requerimientos recibidos, para luego enviarlos a los WS solicitados. Cuando el servicio web retorna los resultados se los envía al WSIG, quien traduce esta respuesta y se la reenvía al agente solicitante. Estos servicios se prestan a través del *Agente Gateway*.

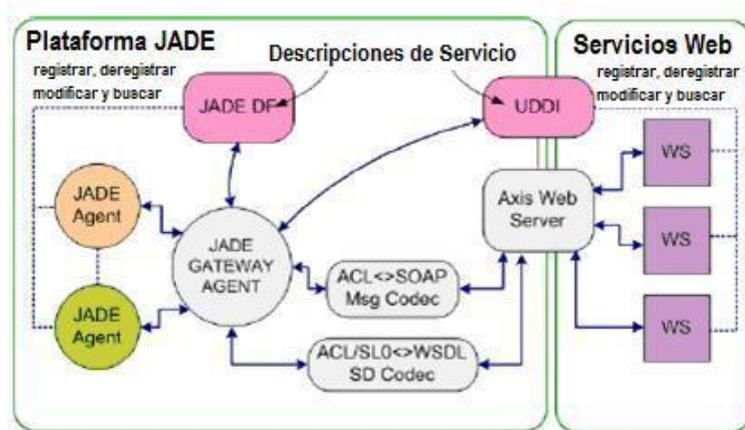


Figura 1. Componente WSIG

Existen numerosas definiciones para los WS y, tomando como referencia las más reconocidas [2] [15], es posible decir que los servicios web *son funcionalidades o procedimientos remotos provistos por organizaciones o entidades para el intercambio de información a través de la Web, los cuales son desarrollados bajo ciertos estándares que posibilitan la comunicación con los sistemas o aplicaciones de software que necesitan accederlos*. Los WS combinan una serie de recursos para lograr un mecanismo de interacción estándar entre diferentes aplicaciones, haciendo posible que la información circule entre ellas sin requerir ningún tipo particular de middleware, ni lenguaje de programación o sistema operativo [13].

Algunos recursos empleados por los WS son: el lenguaje para el intercambio de datos en la Web *XML* (en inglés, eXtended Markup Language). Tanto el formato de los mensajes, como las reglas de codificación y las convenciones para representar las llamadas

a los procedimientos y sus respuestas se especifican en *SOAP* (en inglés, Simple Object Access Protocol). Para enviar los mensajes SOAP es necesario un protocolo de transporte como HTTP o similar. Las características del servicio web se describen mediante *WSDL* (en inglés, Web Services Description Language) que es un tipo de documento XML. Finalmente, *UDDI* (en inglés, Universal Description Discovery and Integration) es un servicio de directorio independiente donde los proveedores pueden registrar sus servicios web y los solicitantes pueden buscar y encontrar los servicios que necesitan.

3. Mediación entre aplicaciones

Como se explicó en la introducción, el enfoque adoptado consiste en extender las facilidades de integración de AIs y WS provistas por componentes de JADE como WSIG, a un contexto multiagente basado en un mecanismo y protocolo de mediación.

Para ello, se utilizó un protocolo de mediación específico (*brokering*) que, en base a requerir que cada parte se ajuste a dicho protocolo de comunicación, permite superar las diferencias existentes entre las partes. Estas diferencias comienzan con la interpretación sintáctica, ya que cada tecnología emplea su propio lenguaje y continúan al establecer la correspondencia semántica entre las prestaciones que se brindan.

Protocolo Brokering. Una solución para procesar requerimientos entre distintos agentes que no comparten el mismo lenguaje o entre WS y agentes, es usar el *Protocolo Brokering* [3][9][10] el cual regula el funcionamiento de un mediador para facilitar la interacción entre las partes. Por ejemplo, si dos agentes quieren comunicarse pero no comparten un lenguaje en común, el Broker puede hacer la traducción; si los agentes no confían entre ellos, éste puede participar como un intermediario confiable. En este protocolo se distinguen tres participantes:

- Agente Solicitante: solicita información o servicios al Broker.
- Agente Broker: conoce a los proveedores y sus servicios.
- Proveedores: anuncian sus servicios al Broker, reciben requerimientos, los procesan y envían las respuestas al Broker.

El protocolo se divide en, *Protocolo de Anuncio* y *Protocolo de Mediación* [14].

- En el Protocolo de Anuncio, el Broker recibe desde los proveedores los servicios disponibles.
- En el Protocolo de Mediación, el solicitante envía una petición al Broker, el cual procesa el requerimiento y decide si puede acordar o rechazar dicha petición. En caso de acordar la solicitud, seleccionará aquellos proveedores que puedan realizar la tarea solicitada e iniciará una comunicación con ellos, luego estos responderán a las peticiones enviadas y el Broker retornará los resultados obtenidos al solicitante.

En la Figura 2 se muestran los pasos del Protocolo de Mediación, en el paso (1) el solicitante envía el requerimiento al Broker y espera por una respuesta mientras el Broker usa sus habilidades de búsqueda para localizar un proveedor que pueda responder a la consulta. Una vez que se localizó al proveedor, el Broker reformula la consulta a ese proveedor (2) y queda a la espera de una respuesta. El proveedor recibe la consulta (3), la procesa y le contesta al Broker. Finalmente en el paso (4) el Broker le responde al solicitante los resultados obtenidos.

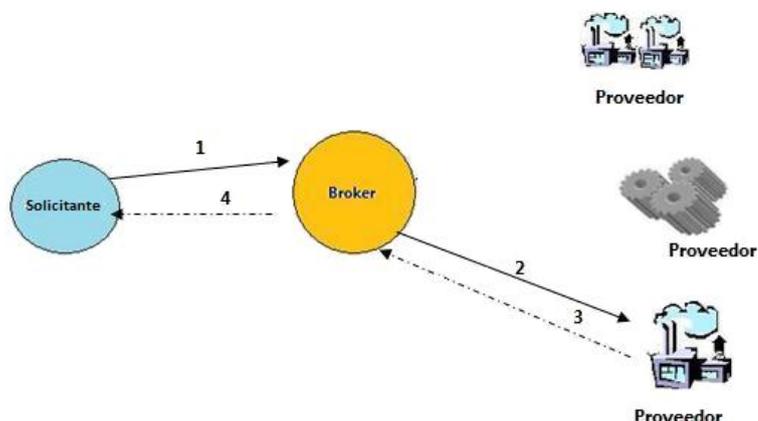


Figura 2. Protocolo de Mediación del protocolo Brokering

Existen distintas implementaciones del modo en que los resultados son retornados al solicitante: si hay varios proveedores que le responden al Broker, éste puede reenviarle todas las respuestas o antes puede filtrar y seleccionar la que considere que es la mejor. La elección depende del dominio en particular y las facultades e inteligencia del agente Broker para procesar los requerimientos.

El Broker es el encargado de realizar las tareas de razonamiento más complejas al tener que interpretar los anuncios de los WS de cada proveedor, analizar las consultas de los agentes solicitantes para encontrar los proveedores adecuados, invocarlos e interactuar con ellos para obtener los resultados que luego tendrá que procesar para brindarle una respuesta al agente solicitante.

Algunas aplicaciones que usan este protocolo son: el comercio electrónico, al responder a los requerimientos de los clientes con ciertas características; en la búsqueda de información, al tener que interactuar con distintas bases de datos y fuentes de conocimiento para responder las consultas; como agentes de asistencia personal, al crear perfiles en base a la información provista por el usuario [7].

Protocolo de Interacción FIPA Brokering. El uso de agentes Broker simplifica la interacción entre los agentes en un sistema multi-agente. La tarea del Broker como mediador tiene dos aspectos: el uso eficiente de la información provista por el solicitante al Broker y la correspondencia de los mensajes entre solicitante y proveedor.

La Figura 3 muestra mediante un diagrama UML, el flujo de mensajes ACL que intercambian los agentes Iniciador, Broker y los proveedores de servicio. En esta secuencia, el Iniciador envía al Broker un mensaje *proxy*, con una expresión referencial que especifica los agentes destinatarios, el acto comunicativo y un conjunto de condiciones proxy como el número máximo de agentes que recibirán el mensaje [4]. El Broker procesa el requerimiento y toma la decisión de acordar o rechazar, se lo comunica al Iniciador mediante *agree* o *refuse* respectivamente. Si rechaza la solicitud, se termina la interacción con el Iniciador.

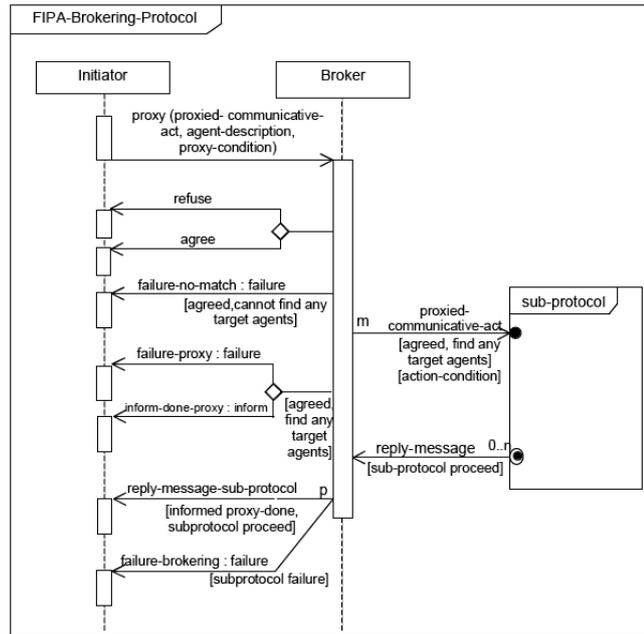


Figura 3. Protocolo de Interacción FIPA Brokering [5]

Si el Broker decidió acordar la solicitud, envía un mensaje al DF para ver si existe algún proveedor de servicio que realice la operación solicitada según la descripción recibida en el mensaje proxy. Si el DF encuentra alguno, devuelve una lista con los proveedores. Aquí comienzan m interacciones que resultarán en una lista de n proveedores. Cada interacción inicia un nuevo sub protocolo, para ello el Broker registra algunos parámetros ACL como el identificador de la conversación (*conversation-id*), emisor (*reply-with*), entre otros. Si no se encuentra ningún proveedor, el Broker retorna el acto comunicativo *failure-no-match* y termina la interacción ³.

A medida que los proveedores van respondiendo, el Broker reenvía al Iniciador las respuestas que recibe. En otros casos, el Broker puede reunir todas las respuestas, combinarlas y enviar una respuesta en un único mensaje. Cuando el Broker recibe como respuestas de los proveedores fallos o inconsistencias, puede determinar si resuelve estas situaciones internamente o le reenvía las respuestas al Iniciador.

Cualquier interacción que forme parte de este protocolo se identifica por un parámetro global único asignado por el Iniciador. Los agentes involucrados en la interacción almacenan este identificador en todos los mensajes ACL de la conversación, lo que permite a cada agente administrar sus actividades y estrategias de comunicación [5].

El protocolo FIPA Brokering, solo implementa el protocolo de mediación detallado para el protocolo Brokering en el párrafo anterior. Al comparar las Figuras 2 y 3, se observa que el rol del Solicitante de la Figura 3 es realizado por el agente Iniciador. El envío del mensaje *proxy* desde el Iniciador al Broker corresponde al paso (1) de la

³ Para un detalle completo de los actos comunicativos FIPA ver [4].

Figura 2. Mientras que el envío de este mensaje desde el Broker a los n proveedores se corresponde al paso (2). Las respuestas de los proveedores se corresponden al paso (3) y el envío de las respuestas del Broker al Iniciador se corresponde al paso (4). Si bien, el Protocolo de Anuncio en la secuencia de la Figura 3 no se muestra explícitamente, los proveedores registran en el UDDI los servicios que ellos proveen. Como se vió en la Sección 2, el agente Gateway actúa como intermediario para el acceso y la comunicación entre los agentes y los proveedores de los servicios web.

4. WSIBA

Las ideas presentadas se emplearon en el desarrollo de WSIBA (por *Web Services Integrated with Broker and Agents*).

Características de la Aplicación. Esta aplicación está compuesta por dos agentes solicitantes, un agente Broker y el agente Gateway de JADE que se adaptó especialmente para que pueda interactuar con el agente Broker cuando este necesite comunicarse con algún servicio web. Los WS utilizados como ejemplo para realizar la conexión con la aplicación JADE, son **Global Weather**⁴ y **Temperature Conversions**⁵, los mismos fueron seleccionados ya que se encuentran disponibles constantemente en Internet y no se requiere de algún costo adicional para poder utilizarlos.

El Protocolo Brokering se inicia cuando los agentes solicitantes hacen peticiones al agente Broker por algún servicio en particular. Luego el Broker se comunica con el agente Gateway para que transforme los mensajes e invoque a los servicios web solicitados, quedando a la espera por una respuesta. En esta implementación, los sub protocolos de interacción con los proveedores son iniciados por el agente Gateway, quien realiza los procesos de traducción; en la Sección 2 se describió como éste actúa de intermediario para que los WS y los agentes puedan entenderse. Una vez que los WS retornan los resultados al Gateway, este le reenvía los resultados al Broker quien luego le enviará la respuesta al solicitante. De esta manera se cumple la secuencia completa del protocolo Brokering entre el solicitante, el broker y el proveedor, como se muestra en la Figura 3.

Funcionalidades de la Aplicación. Al ejecutar el sistema se lanzan automáticamente la plataforma JADE y el agente Gateway, luego se realizan las registraciones de los servicios web en el UDDI. Se crean los agentes solicitantes y el agente Broker, el cual estará activo esperando por los requerimientos de algún agente solicitante.

El servicio Temperature Conversions provee varias funciones de conversión de temperatura ambiental: Por ejemplo, si selecciona la operación *Celsius a Fahrenheit*, el usuario deberá ingresar la temperatura en grados Celsius que necesita que se convierta a grados Fahrenheit. Una vez que el usuario ingresó el valor y envía la petición, el solicitante le envía al Broker un mensaje con la operación y el valor ingresado, esta acción es la que inicia el protocolo Brokering, tal como se indica en la Figura 4⁶. El Broker con esta información envía un mensaje al DF para ver si existe algún proveedor de servicio que realice la operación solicitada, si el DF encuentra alguno, le envía una lista con estos proveedores. Aquí el Broker le responde al solicitante informándole que acepta

⁴ <http://www.webservicex.net/ws/WSDetails.aspx?CATID=12&WSID=56>

⁵ <http://webservices.daehosting.com/services/TemperatureConversions.wso>

⁶ La secuencia completa del protocolo brokering, es suministrada por el agente Sniffer

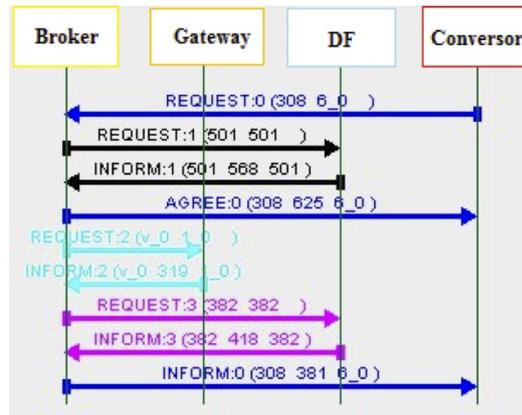


Figura 4. Secuencia de mensajes del Protocolo Brokering

el requerimiento y selecciona un proveedor del mensaje recibido desde el DF, crea un mensaje con la operación solicitada y se lo envía al Gateway quien lo traducirá en un mensaje SOAP y lo enviará al servicio web correspondiente utilizando la información guardada en el registro UDDI, en este punto se inician los m sub protocolos como se mostraba en la Figura 3 para el Protocolo FIPA Brokering en la sección anterior. Cuando el Gateway recibe la respuesta del servicio web, realiza la traducción correspondiente y le envía el mensaje al Broker, quien lo procesa y genera un mensaje con el resultado de la operación y se la envía al solicitante (en este punto finaliza el protocolo Brokering), quien luego muestra por pantalla la información al usuario, como se muestra en la Figura 5. En el caso de que el usuario elija otra de las operaciones disponibles o se



Figura 5. Respuesta de la operación Celsius a Fahrenheit

utilicen las funcionalidades del agente solicitante *Clima*, se realizarán los mismos pasos entre los agentes Solicitante, Broker, DF y Gateway, solo se modificará el contenido y los parámetros de los mensajes que se intercambian.

5. Conclusiones y trabajos futuros

La propuesta presentada en este trabajo traslada la función de integración entre WS y AIs del contexto del agente individual al contexto multiagente. Éste no constituye

sólo un aspecto conceptual, sino que tiene también distintas implicancias prácticas. En primer lugar, el protocolo de brokering especifica clara y formalmente (de acuerdo a las especificaciones de FIPA) los pasos necesarios para establecer la integración de ambas tecnologías. Por otro lado, se delega en el agente mediador la logística y responsabilidad de la interconexión de ambas partes de una manera más flexible, pudiéndose adoptar distintos estándares de comunicación, distintos protocolos de mediación, etc. De igual manera, el agente broker puede ser extendido con más inteligencia en su rol de mediador a lo largo del tiempo, dotándolo de capacidades de aprendizaje para determinar la efectividad, costo y *reputación* de los distintos WS, etc.

Como trabajos futuros, se plantea la extensión de WSIBA para el trabajo con un número mayor de AIs y WS para medir su desempeño en un entorno más exigente. Otra manera de continuar con la integración de AIs y WS consiste en que los agentes elaboren planes que incluyan el uso de WS. Si bien la composición de WS en planes se está investigando desde ya hace varios años, el uso de mediación y el protocolo de brokering puede ser una alternativa interesante para lograr dicha composición. Por último, una extensión obvia de este trabajo, es la incorporación de los mecanismos implementados en un agente Web para una aplicación específica. En este sentido, el próximo paso será integrar en un AI desarrollado en JADE, distintas facilidades (que estarán disponibles via WS) que se están desarrollando en nuestro grupo de investigación para la determinación de calidad de información en la Web, en el contexto de un proyecto FP7 de la comunidad europea [11].

Referencias

1. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-agent Systems with JADE. John Wiley and Sons, LTD (2007)
2. Chase, N.: Understanding web services specifications: SOAP. IBM Corporation (2006)
3. Decker, K., Williamson, M., Sycara, K.: Matchmaking and brokering (1996)
4. FIPA: Communicative Act Library Specification (2002)
5. FIPA: Brokering Interaction Protocol Specification (2003)
6. H. Chen, T. Finin, A.J., Chakraborty, D.: Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing* 8(6), 69–79 (2004)
7. Jonker, C.M., Treur, J.: Information broker agents in intelligent websites. Vrije Universiteit Amsterdam, Department of Artificial Intelligence (1996)
8. Kosala, R., Blockeel, H.: Web mining research. *SIGKDD Explorations* 2, 1–15 (2000)
9. Kuokka, D., Harada, L.: On using kqml for matchmaking. In: *ICMAS*. pp. 239–245 (1995)
10. Labrou, Y., Finin, T.: A semantics approach for kqml. In: *In Proceedings of CIKM'94*. pp. 447–455. ACM Press (1994)
11. Lex, E., Voelske, M., Errecalde, M., Ferretti, E., Cagnina, L., Horn, C., Stein, B., Granitzer, M.: Measuring the quality of web content using factual information. In: *Proc. 2nd WICOW/AIRWeb Workshop Web Quality*. pp. 7–10. ACM, NY, USA (2012)
12. N. Zhong, J.L., Yao, Y.: Envisioning intelligent information technologies through the prism of web intelligence. *Commun. ACM* 50(3), 89–94 (2007)
13. Overeinder, B., Verkaik, P., Brazier, F.: Web service access management for integration with agent systems (2008)
14. Paolucci, M., Soudry, J., Srinivasan, N.: A broker for OWL-S web services. Robotics Institute, Carnegie Mellon University (2005)
15. Richards, D., van Splunter, S., et al.: Composing ws using an agent factory (2003)