

EL Protocolo Contract-Net en un Ambiente de Aprendizaje Colaborativo

Guillermo Aguirre, Berta García, Guillermo Leguizamón*

Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 San Luis, Argentina
{gaguirre, bgarcia, legui}@unsl.edu.ar

Resumen

EL avance de las nuevas tecnologías de comunicación ha permitido desarrollar plataformas de enseñanza/aprendizaje cuyo diseño tiene en cuenta la posibilidad de la comunicación, colaboración y coordinación entre los individuos que conforman un grupo dentro de una comunidad educativa *on-line*. La colaboración entre los miembros de este tipo de entornos es de gran importancia para el logro de los objetivos propuestos. La introducción de agentes inteligentes en un entorno colaborativo podría facilitar muchas de las tareas del grupo de aprendizaje. Sin embargo, el éxito de su inclusión dependerá en gran medida del modelo usado para realizar la interacción entre los usuarios y principalmente entre los agentes que conforman el sistema.

En este artículo presentamos una adaptación del modelo de *contract-net* para un ambiente colaborativo inteligente. Diferentes componentes del ambiente colaborativo son visualizados desde la perspectiva del modelo de *contract-net* haciendo las adecuaciones respectivas. La presentación del modelo adaptado incluye una breve discusión de los aspectos de su implementación a través de una plataforma de desarrollo de sistemas multi-agente conocido como JADE.

Palabras Claves: Aprendizaje colaborativo, agentes, *contract-net*, plataforma de desarrollo de sistemas multi-agentes.

1. Introducción

Las comunidades *on-line* [6] refieren a grupos con intereses comunes, objetivos compartidos, actividades y gobierno, grupos e individuos que colaboran para compartir recursos y satisfacer las necesidades propias y de otros. Entre los elementos que las componen podemos citar:

- Individuos: interactúan socialmente ya sea para satisfacer sus propias necesidades o realizar roles especiales tales como liderar o moderar.

*Integrantes del Proyecto Promocionado: Ambientes Colaborativos Inteligentes P3033, Aguirre y Leguizamón integrantes del LIDIC - UNSL

- Propósitos compartidos: interés, necesidad, intercambio de información o servicios que proveen un sentido a la comunidad.
- Normas: en la forma de suposiciones tácitas, rituales, protocolos, reglas, leyes que guíen las interacciones entre las personas.
- Sistemas de computación: soportan y mediatizan las interacciones sociales y facilitan el sentido de unidad.

En el caso particular de las comunidades educativas¹, se hace necesario, además, reunir características que permitan soportar necesidades especiales de los estudiantes [6]:

- Recursos: herramientas de comunicación usados para comunicarse con la clase, dentro de pequeños grupos, uno a uno o con el profesor; herramientas para realizar proyectos escritos en forma colaborativa, obtener retro-alimentación, chequear notas, etc.
- Guía: los profesores deberían guiar efectivamente a los estudiantes, llevarlos a usar los recursos creativamente, y asegurarse de recompensar sus esfuerzos.
- Retro-alimentación: puede tomar varias formas en distintos puntos del proceso de aprendizaje. Podría ser el reconocimiento de un profesor o de los pares por una tarea bien hecha, o la asignación de una nota, según corresponda. O podría ser la posibilidad de ayudar a otro estudiante. Algunos entornos proveen facilidades para los profesores, soportando mantenimiento de registros, valoración y retro-alimentación.
- Actividad placentera: cuando el aprendizaje es placentero es generalmente más significativo. Las características que permiten compartir, congeniar, confiar, soportar y colaborar y dejan de lado la agresión y los comportamientos egocéntricos hacen al aprendizaje una actividad motivante.

El creciente avance de las nuevas tecnologías de información ha permitido visualizar diversos campos para su utilización, particularmente aquellos en donde la comunicación, velocidad de acceso a los datos y trabajo colaborativos son factores importantes. Uno de estos campos está estrechamente vinculado a grupos de Aprendizaje Colaborativo Respaldo por Computadoras (ACRC), un caso particular de una comunidad *on-line*. Este enfoque se sostiene en la premisa de la disponibilidad de sistemas de computación que puedan respaldar y facilitar los procesos y dinámicas propias de grupos en situaciones donde éstas no puedan ser alcanzadas a través del contacto directo pero que tampoco pretendan reemplazar la comunicación directa o cara-a-cara. De esta manera un ACRC puede ser diseñado para ser usado por múltiples usuarios trabajando ya sea, en la misma computadora (en diferentes tiempos) o bien a través de computadoras remotas dispuestas en red con el objetivo de generar un flujo de datos entre los participantes como por ejemplo, ideas, documentos, información de acceso y principalmente, proveer de retro-alimentación en las actividades involucradas en la resolución de problemas comunes.

Es importante destacar que los roles de los individuos en un entorno colaborativo no deberían permanecer estáticos a lo largo de tiempo. Por el contrario, la dinámica en los cambios de roles puede favorecer la tarea de aprendizaje en los grupos de estudio en los que las capacidades de los integrantes no son similares, y cada uno se destaca en algo particular. La determinación continua de los posibles roles de los individuos dentro del grupo es de gran importancia para el éxito de este tipo de entornos.

¹En este contexto se refiere implícitamente a comunidades educativas *on-line*.

El proceso de determinación de dichos roles puede ser alcanzado a partir de la información obtenida en forma explícita (vía formularios) o implícita, es decir, haciendo un seguimiento del comportamiento del individuo en relación a la calidad y grado de interacción de éste en el grupo de aprendizaje. Dado que un individuo puede asumir diferentes roles según la situación, deberá existir en el sistema un mecanismo que permita identificar aquellos individuos que puedan facilitar un recurso cuando otro del grupo lo necesite. Un recurso muy importante a considerar es el conocimiento que un individuo pueda tener sobre un determinado tema y que otros puedan requerirle a los efectos de avanzar en la tarea de aprendizaje. Desde esta perspectiva surge la motivación y objetivo del presente trabajo de incorporar un mecanismo de contratos dentro del sistema para el requerimiento y obtención de algún tipo de ayuda² entre los miembros del grupo. En este contexto, el protocolo de red de contratos (*contract net protocol*) provee de un marco adecuado para ser usado por un sistema multi-agentes dentro de una plataforma ACRC.

El presente trabajo está organizado de la siguiente manera: la próxima sección introduce brevemente el ambiente colaborativo y aquellas características que posibilitan la inclusión de un sistema multi-agente. La sección 3 presenta el modelo de *contract-net* y su adaptación al ambiente colaborativo. En la sección 4 se discuten algunos aspectos relacionados con la implementación del modelo propuesto. Finalmente, en la sección 5, se analiza la presente propuesta y se evalúan posibles líneas de investigación en relación al grado de avance actual.

2. Ambiente Colaborativo

El trabajo en grupos colaborativos puede estar dado en ámbitos de trabajo o educación. En ambos casos se desarrollan tareas conjuntas para lograr un fin, la diferencia reside en el contexto y propósito: trabajo o aprendizaje. Para aplicar el aprendizaje colaborativo en ambientes informáticos se precisan entornos que soporten y posibiliten compartir datos, recursos, mensajes, etc; así como mecanismos que permitan las interacciones en tiempo real necesarias para que el aprendizaje se produzca en forma efectiva. El uso de la computadora como un dispositivo mediador permite a los usuarios comunicarse, cooperar y colaborar en actividades (síncronas y asíncronas) conjuntas a través de una red, proporcionando instrumentos para la construcción del conocimiento. El aprendizaje se ve favorecido en los grupos de estudio en los que las capacidades de los integrantes no son similares, y cada uno se destaca en algo particular. En grupos heterogéneos la ayuda proporcionada por el sujeto “más capaz” en determinado saber debería ser visible a los demás integrantes de la interacción. De manera tal que el sujeto que aprende pueda precisar el nivel de autonomía/dependencia alcanzados en un momento dado. La visión del sujeto de su propio proceso de aprendizaje y la mirada autorizada de quien está a cargo de guiar este proceso deberían ser parte de la devolución del ambiente colaborativo, ya sea éste producido por un agente de software o un agente humano. ¿Cómo conformar equipos colaborativos que trabajen en entornos de aprendizaje virtuales? La organización de un grupo o curso es realizada por el profesor a partir de las características de sus alumnos: fortalezas, debilidades, preferencias; es decir, a partir del conocimiento de los ritmos de aprendizaje de un grupo de trabajo para poder guiarlo, cuidando se dé la diversidad y el aporte genuino de todos los que interactúan en él. Es común que en este tipo de ambientes virtuales, el ingreso se realice a través de un proceso de inscripción explícito. Sin embargo, ante la imposibilidad de un diálogo cara a cara en este tipo de entornos, es frecuente la utilización de cuestionarios que permitan la caracterización del usuario. Un cuestionario inicial produce cierto rechazo, suele resultar muy extenso, y por otra parte, generalmente es respondido desde

²El tipo de ayuda considerada en este artículo está vinculada a cualquier aporte que permita la consecución de las tareas asignadas.

el “deber ser” —es decir, lo que se responde en función de las expectativas que el otro tiene de mí. Esto generalmente se traduce en una determinación incorrecta, que además será asociada al usuario de aquí en más, y mientras permanezca en el ambiente. Esta información podría completarse realizando un seguimiento de las actividades en el tiempo que transcurre el estudiante dentro de la plataforma, de esa manera tendríamos un perfil dinámico, en constante redefinición. Los agentes de software deberían monitorear el trabajo de cada uno de los participantes. Esta información podría ser utilizada por el profesor para redefinir los equipos e intercambiar los roles de los integrantes de modo tal que favorezcan el trabajo colaborativo y posibilite: valorar el aporte y opinión de cada uno de los integrantes; comunicar las opiniones, ideas, sentimientos y convicciones propias con claridad y eficacia; fomentar el trabajo en equipo asumiendo diversos roles (coordinador, secretario, etc.); seleccionar los recursos que permitan un mejor desarrollo de la tarea propuesta en un ambiente interactivo y creativo; evaluar las características de la tarea para determinar si realmente estimulan el trabajo colaborativo; introducir los temas que se abordarán y los problemas que se deben resolver con claridad y precisión, cuidando mantener el interés de los alumnos; determinar si los alumnos poseen los conocimientos informáticos necesarios para realizar las tareas; asumir un rol de guía, que no sólo cuestione sino también estimule el desempeño; entregar orientación e información oportuna, resaltando principios y conceptos relevantes; estimular prácticas de interacción; ayudar a los alumnos a realizar una reflexión meta-cognitiva del trabajo realizado, con el fin de discutir cada una de las etapas del proceso y así optimizar trabajos futuros.

Por su parte los alumnos deberían organizarse de forma que todos los integrantes puedan participar activa y equitativamente para: trabajar en equipo en pos un objetivo común; fijar objetivos a corto y largo plazo, revisando el cumplimiento de los mismos para identificar los cambios que se deben realizar con el fin de optimizar sus acciones; hacer la parte del trabajo que corresponde a cada uno y poner a disposición de todos los miembros del grupo el material correspondiente; interactuar en forma sincrónica intercambiando información, razonamientos y puntos de vista para que exista retroalimentación entre los miembros del equipo; tener la posibilidad de comunicarse y poner a disposición inter grupos tanto lo experimentado en las distintas etapas del proceso, como los resultados obtenidos.

El ambiente de aprendizaje colaborativo en desarrollo [4, 5] está dirigido a grupos de alumnos-docentes involucrados en un proceso de enseñanza-aprendizaje (un curso regular) en el que profesores y alumnos podrán acceder sincrónica o asincrónicamente para realizar distintas actividades. Proveerá herramientas tales como: pizarra, foros, correo electrónico, chat, etc; y permitirá compartir objetos tales como: bibliografía, documentos, enlaces, etc.

Cada usuario (profesor o alumno) realizará el ingreso al ambiente identificándose con clave y contraseña. A partir del proceso de registro el usuario obtendrá una dirección de correo electrónico, generada automáticamente por el ambiente, a través de la que se gestionará la comunicación de lo que corresponda exclusivamente a ese curso. El profesor podrá realizar acciones con respecto a un curso: dar de alta o baja, gestionar información académica: fijar fechas, solicitar listado de alumnos, publicar calificaciones, ingresar/modificar programa y correlatividades, consultar desempeño académico de los alumnos. Cada curso estará organizado en módulos, en los que el profesor podrá ingresar el material de estudio, prácticos, entre otras cosas. Las actividades a realizar por el alumno podrán ser individuales o grupales, de acuerdo a las necesidades y a lo sugerido por el profesor. Además, dentro del ambiente colaborativo está previsto un sistema multi-agente que principalmente estará destinado a administrar los intereses de los usuarios a los que representa.

Por diferentes circunstancias, durante la realización de alguna tarea, un alumno que esté trabajando dentro del ambiente colaborativo puede necesitar ayuda. A partir de una petición explícita de ayuda, el ambiente debería poder asistir o guiar al alumno a superar esa circunstancia. La ayuda podría provenir del ambiente mismo, de otro alumno o del profesor. ¿Cómo determinar qué tipo de ayuda

necesita y quién puede proveerla? Muchas de las plataformas existentes en la actualidad proveen de mecanismos para el registro de las actividades de los alumnos, aunque no siempre existen procesos que automaticen el análisis de las mismas y lo pongan a disposición del docente para futuras intervenciones. Dado que realizar el análisis de forma manual insumiría un tiempo considerablemente alto, el uso de agentes inteligentes permitiría la automatización de estos procesos, identificando por un lado la necesidad a partir de la demanda de ayuda explícita, y por otro quienes están en condiciones de satisfacerla.

3. Contract-net en el Ambiente Colaborativo

Considerando las características del entorno colaborativo descrito, una de las responsabilidades que han sido asignadas al sistema multi-agente es facilitar el encuentro entre pares de manera que ellos mismos se presten ayuda mutuamente. Esa situación de la vida real es traducida dentro del sistema multi-agente del ambiente colaborativo como la caracterización de dos grupos de agentes bien destacados: aquellos que buscan ayuda y aquellos que la ofrecen. La situación se presenta como muy prometedora pero ¿cómo establecer la conexión entre los agentes de cada grupo para maximizar los beneficios para el conjunto?. La búsqueda de soluciones para este tipo de problemas nos lleva a pensar en una alternativa distribuida que considere las características propias de cada agente.

3.1. Contrato de ayuda.

Haciendo algunos cambios es posible encontrar mucha semejanza entre esta situación y la que motivó a Smith [7] a proponer el protocolo de red de contratos (*contract net protocol*) para resolver problemas intrínsecamente distribuidos. En su artículo, Smith presenta el *problema de conexión* en donde se busca la forma lograr que nodos con tareas a ser ejecutadas encuentren los nodos ociosos más apropiados para que ejecuten aquellas tareas. El objetivo perseguido por este protocolo es establecer una relación entre el cliente, denominado *manager* y un prestador de servicio, el *contractor*; esto se consigue a través de un requerimiento de ofertas y una evaluación de las propuestas recibidas. El camino para establecer una relación se puede dividir en cuatro pasos (ver figura 1):

- El primer paso, involucra el requerimiento de ofertas. El *manager* envía una descripción de la tarea a los posibles oferentes.
- Los potenciales *contractors* elaboran sus ofertas (segundo paso).
- Durante el tercer paso, el *manager* recibe y evalúa las ofertas y luego selecciona una de ellas para acordar un contrato.
- En el último paso el agente que hizo la oferta ganadora, cierra el contrato confirmando que está dispuesto a cumplir con lo acordado.

En el artículo de Smith un nodo de la red de contratos está formado por tres procesadores y una base de datos local; los tres procesadores son: un procesador de comunicación, encargado de la comunicación de bajo nivel, un procesador de tarea, encargado de realizar la tarea específica del nodo y un procesador de contrato, encargado de los mensajes del protocolo de alto nivel y administrador de los recursos del nodo. Nuestro enfoque considera que la funcionalidad de un nodo ha de ser desempeñada por un agente de software, mientras que la tarea de cada procesador dentro del nodo,

corresponde a una tarea específica³ del agente. (Ver la sección 4 para una descripción más detallada de la implementación de estas tareas dentro del agente).

Cuando un agente inicia un requerimiento de ayuda, el primer paso es encontrar cuales son los agentes más adecuados para satisfacer ese requerimiento. La administración del sistema multi-agente no debe permitir que sólo unos pocos agentes consigan acaparar todas las recompensas, dado que se podría traducir en que ciertos agentes se vean sobrecargados con consultas, sin dar la posibilidad que otros agentes brinden también ayuda. Esto permitiría balancear las cargas e incentivar la participación dentro del ambiente colaborativo.

En el sistema multi-agente, los agentes podrán asumir el rol de *manager* o el de *contractor*. Un agente en el rol de *manager* será aquel que iniciará una conversación para encontrar los agentes más adecuados que puedan ayudarlo. Un agente *contractor* será responsable de brindar ayuda, recibirá diversos pedidos de ayuda y responderá si los requerimientos pueden ser satisfechos.

Los roles de cada uno no serán establecidos de antemano, ni de manera estática tal cual se establece en el protocolo original propuesto por Smith [7]. El *manager* recibirá las respuestas a su requerimiento y las evaluará, respondiendo a la propuesta más adecuada, cerrando así un contrato de colaboración. El protocolo considera que un contrato inicial puede generar nuevos contratos en los cuales los agentes *contractors* jugarán el rol de *manager* contratando a su vez a otros *contractors* que le ayuden con el contrato inicial.

Con el objeto de poder realizar los contratos dentro del sistema multi-agente, es necesario que cada agente pueda crear y mantener un modelo aproximado del usuario al que representa, en el que esté representado principalmente aquello relacionado a sus conocimientos como también, lo referente a su relación con el resto de los usuarios en el aspecto social [1]. Desde el punto de vista de sus conocimientos, se puede hacer un perfil a partir de un cuestionario sobre la temática en cuestión e ir enriqueciendo esa información básica considerando el desempeño que el usuario haya conseguido al desarrollar los diferentes trabajos individuales (*homeworks*) entre otros. El perfil social se concentrará principalmente en cual ha sido su grado de colaboración con sus pares y la calidad de la misma.

3.2. La adaptación del protocolo de red de contratos.

A continuación son descriptos los mensajes necesarios para la adaptación del protocolo y la forma en que procederán las distintos agentes involucrados en un contrato para brindar y conseguir ayuda. La figura 1 describe⁴ el intercambio de mensajes para llegar a un contrato. Para simplificar la descripción del comportamiento de los agentes consideramos que un agente X cumplirá el rol de *manager* y otro agente Y será el *contractor*, en la realidad puede darse que un agente esté participando de distintas conversaciones cumpliendo diferentes roles. Dentro de X se consideran dos módulos (Módulo A y Módulo B), uno encargado de generar el anuncio de necesidad y procesar los ofrecimientos y el otro responsable de cerrar el contrato.

El objetivo del modelo adaptado al ambiente colaborativo es lograr poner en contacto a los agentes que representan a los usuarios que necesitan ayuda y los agentes de aquellos que están en condiciones de brindarla.

³En términos de la nomenclatura utilizada en la plataforma de desarrollo es lo que se conoce como *comportamiento* (behaviour).

⁴La convención utilizada es la que se utiliza en [3] y denominada BRIC.

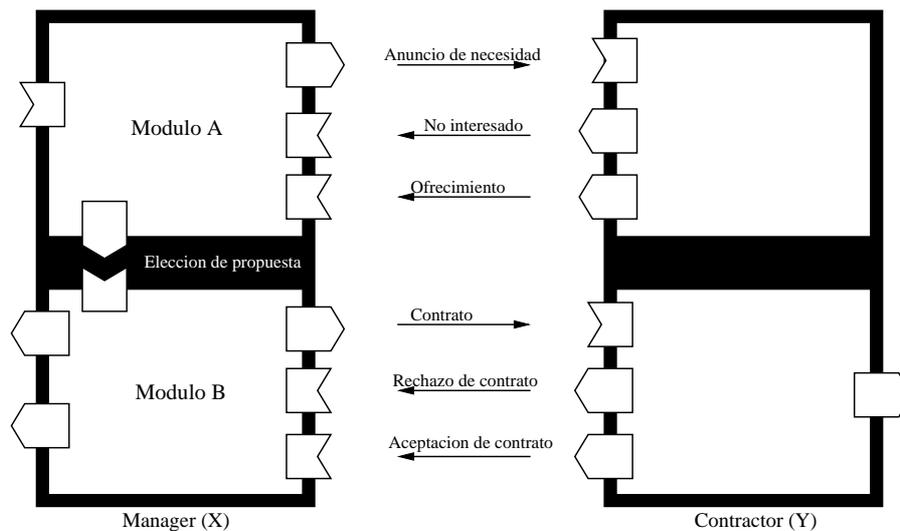


Figura 1: Adaptación del Protocolo contract net

3.2.1. Anuncio de necesidad.

Un agente que necesite ayuda iniciará una negociación de contrato emitiendo un mensaje de necesidad de ayuda. Este agente jugará el papel de *manager*. El mensaje podría ser destinado a todos los agentes o bien a ciertos agentes seleccionados a partir de información social relevante. Una decisión que debe ser tomada en este punto es si conviene difundir el anuncio de necesidad o bien enviarlo a un agente determinado para establecer un contrato directo sin considerar otras ofertas.

Los mensajes de este tipo presentan cuatro campos. Uno de ellos es la especificación de selección, que es una lista de criterios a ser satisfechos para poder brindar ayuda. El campo de identificación combinado con el campo anterior servirán para establecer el conjunto de agentes candidatos para salir en ayuda del *manager*. Un tercer campo especifica la síntesis de la ayuda que permite que los agentes hagan su propia ponderación del pedido en relación a otros posibles pedidos que deban ser considerados al mismo tiempo. La especificación de la oferta orienta a los oferentes acerca de cual es la información relevante que será considerada de su propuesta. Finalmente un campo que establece el límite de validez que tiene el anuncio.

3.2.2. Procesamiento de un anuncio de necesidad.

Cada agente mantiene una lista de los anuncios que aun no han vencido y para los cuales el agente cumple las pre-condiciones. Esta lista es ordenada según un criterio local propio de cada agente, teniendo en cuenta las particularidades de cada caso. Una serie de procedimientos son realizados una vez que se recibe el anuncio: Primero es necesario ubicar el anuncio recién recibido en el *ranking* de los anuncios ya considerados, por defecto el último anuncio recibido estará al tope del *ranking*. Luego cuando es posible generar una oferta, se considera uno de los anuncios de necesidad de los que son mantenidos y se responde con una oferta o bien con un rechazo del anuncio. Cuando se acepta, el mensaje con la oferta presenta un campo denominado síntesis del agente en donde se describe la capacidad del agente para ayudar en lo relacionado con la necesidad anunciada.

3.2.3. Procesamiento de ofertas.

La información relacionada a las conversaciones que han sido comenzadas pero aun no han finalizado, se mantienen dentro del *manager* en una cola hasta que éstas terminen. Se mantiene también una lista de los ofrecimientos de ayuda recibidos, ordenada según un *ranking*. Si el *manager* detecta que un ofrecimiento cumple con los requisitos esperados, puede otorgar un contrato al oferente correspondiente. Si por otro lado se alcanza el tiempo de expiración y no se ha escogido ningún ofrecimiento, las acciones posibles son: acordar con la oferta que resulte más aceptable, transmitir otro anuncio de necesidad o si no ha llegado ninguna propuesta aceptable continuar esperando durante un intervalo adicional de tiempo.

El oferente que efectuó la oferta elegida es informado que ha sido escogido, por medio de un *anuncio de acuerdo* generado por el *manager*. Este tipo de mensajes poseen un campo de especificación de necesidad en donde se transmiten los datos necesarios para comenzar con la ayuda o cualquier dato que el oferente hubiese requerido.

Una vez que el *anuncio de acuerdo* ha sido establecido con algún agente se inicia una etapa de traspaso de información entre el *manager* y el *contractor*, el intercambio se hace a través de tres estructuras de datos:

- “contrato”, contiene el nombre asignado al contrato, quien es el *manager*, la lista de los agentes a los que se les ha enviado información relacionada con este contrato, la lista de agentes que están trabajando sobre contratos relacionados, una referencia a una instancia de la próxima estructura de datos, los resultados alcanzados y una lista de subcontratos relacionados.
- “necesidad”, contiene los siguientes campos: tipo de ayuda que se está brindando, una breve descripción y el detalle de las alternativas que han sido tomadas para llevar adelante el contrato.
- “subcontrato”, contiene los siguientes campos: nombre del contrato, identificación del *contractor*, las listas de los agentes precedentes y subcontratados para hacer el contrato correspondiente.

Un reporte es usado por el *contractor* para informar al *manager* una parte de la ayuda (reporte intermedio) o la parte final de la ayuda (reporte final). Los reportes intermedios pueden ser útiles para que el *manager* analice si la ayuda está bien orientada o requiere algún cambio de orientación. En estos casos, el *contractor* debe esperar la confirmación del *manager* para continuar con su contrato. El *manager* puede utilizar el mensaje terminación para que el *contractor* cancele su ayuda para un contrato determinado y finalizar con todos los subcontratos que estuviesen asociados.

Además de las situaciones descriptas, que sucederán la mayoría de las veces, hay otras circunstancias especiales que pueden darse dentro del protocolo. Por ejemplo “oferta de respuesta inmediata”, la que será usada para reducir el tiempo que un *manager* debe esperar para recolectar todas las ofertas. Un “contrato directo”, permite evitar la etapa de negociación. Los mensajes de “requerimiento” e “información” permiten el intercambio de información sin necesidad de mayores controles una vez que un contrato ha sido acordado. Otro posible mensaje es el de “agente disponible”, usado para anunciar que un agente actualmente se encuentra sin brindar ayuda.

4. Aspectos de la Implementación

En esta sección se describe la plataforma usada para la implementación del sistema multi-agente la cual ha sido escogida por las facilidades que brinda y por su adaptación a diferentes sistemas ope-

rativos. JADE (Java Agent DEvelopment Framework) es una plataforma de software implementada en JAVA que facilita la creación, prueba y depuración de sistemas multiagentes. Este proyecto fue desarrollado en CSELT S.p.A. en Torino, Italia [2]. Esta plataforma de agentes (agent platform AP) respeta los estándares establecidos por la Foundation for Intelligent Physical Agents (FIPA). Se puede decir que trata con todos aquellos detalles que no son propios de un agente particular sino aquellos que son independientes de las aplicaciones, tales como parsing, codificación y transporte de mensajes, o el ciclo de vida del agente. A fin de simplificar la tarea de desarrollar un sistema multi-agente (SMA), JADE ofrece el servicio de pasaje de mensajes de una manera eficiente y flexible; cada agente accede a su cola privada de mensajes. Todos los mensajes entre agentes son codificados en lenguaje ACL (agent communication language).

Una plataforma JADE consiste de contenedores de agentes, distribuidos sobre una red de computadoras, y tres agentes especiales destinados a tareas administrativas: AMS (Agent Management System), DF (Directory Facilitator), y el ACC (Agent Communication Channel), que están agrupados bajo el contenedor Principal. A su vez, las distintas plataformas podrían estar conectadas entre sí.

Los agentes especiales son descritos a continuación: Directory Facilitator es un tipo especial de agente que provee el servicio de “páginas amarillas” a los otros agentes. Los agentes pueden registrar sus servicios con el DF, o consultarlo acerca de información sobre otros agentes. Alrededor de un DF se genera un dominio, una organización lógica de agentes con fines específicos.

AMS provee un servicio de nombres de agentes ya que mantiene un índice de los nombres de todos los agentes registrados en la AP. El índice incluye el nombre unívoco del agente y su dirección de transporte asociada dentro de la AP.

ACC se encarga de llevar mensajes entre plataformas de agentes, conecta a todos los agentes dentro de una AP con otras plataformas. A fin de cumplir con los estándares de FIPA el protocolo IIOP (Internet Inter-orb Protocol) es utilizado por ACC, otros protocolos podrían también ser provistos.

Las principales características de un contenedor básico de agentes son: requiere una máquina virtual JAVA para proveer un marco completo en el que los agentes ejecutan de manera concurrente. Es un ambiente multi-thread, compuesto de un thread para cada agente. controla el ciclo de vida de sus agentes, pudiendo comenzar suspender, reactivar o finalizar los agentes que lo componen. En su interior existe un despachador de mensajes, a través del cual ingresan los mensajes al contenedor y son derivados a sus agentes destinatarios; en el caso del envío de mensajes hacia otro contenedor, el despachador mantiene toda la información necesaria para ubicar al agente destinatario, garantizando el envío de mensajes mediante el protocolo más adecuado.

El intercambio de mensajes entre los agentes de la plataforma es responsabilidad del sub-sistema de comunicación entre agentes. Cada AP presenta un contenedor especial, llamado Front-end, que mantiene información para la administración de toda la AP. Esta información es: un registro que permite la invocación de métodos remotos en JAVA (Remote Method Invocation – RMI), una tabla de los contenedores dentro de la AP y una tabla global de todos los agentes dentro de la AP. Una vez que la plataforma ha comenzado a funcionar, los distintos contenedores que sean iniciados, deben registrarse en la tabla de contenedores. A medida que nuevos agentes se incorporan en los contenedores, se registran en una tabla propia de cada contenedor y en la tabla global mantenida dentro del contenedor front-end.

Dentro de un contenedor, los mensajes que tienen como destino algún agente en otro contenedor, son dirigidos a partir de las direcciones conservadas dentro de la tabla local, para evitar el acceso a la tabla global. En ocasiones es necesario acceder a información más actualizada directamente desde la tabla global.

Para llevar adelante las tareas asignadas a un agente, cada uno sigue un Modelo de ejecución. A fin de asegurar la completa autonomía de cada agente, es necesario que tenga un thread de control

interno que gobierne su ejecución. En situaciones comunes un agente puede encontrarse en más de una conversación con otros agentes por lo que debe poder desarrollar varias tareas al mismo tiempo; en este sentido, JADE usa la abstracción de Comportamiento para modelar las tareas que un agente es capaz de llevar a cabo, es decir que los agentes instancian determinados comportamientos de acuerdo a sus necesidades y capacidades.

Los agentes específicos son creados a partir de la clase Agente, extendiendo esta clase y las tareas particulares que debe hacer, a partir de la clase Behaviour. Las clases derivadas de Behaviour son instanciadas y agregadas dentro del agente. De esta forma un agente JADE es una clase que extiende la clase Agente y hereda así un comportamiento que le permite desenvolverse en la AP para hacer tareas como registrarse, configurarse, permitir su administración remota, etc. Además hereda una serie de métodos que le permiten hacer su tarea específica, como ser enviar y recibir mensajes, usar protocolos de interacción estándares, registrarse en varios dominios, etc.

Las tareas de administración del SMA, pueden ser hechas por medio de la interfaz gráfica del agente RMA.

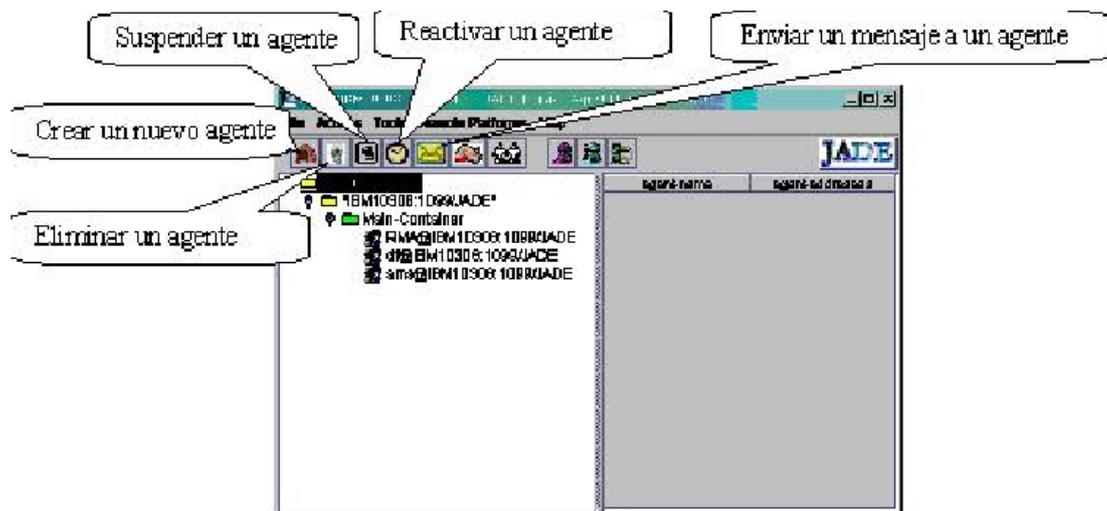


Figura 2: Interfaz gráfica de JADE

Desde la barra de herramientas es posible crear, eliminar o suspender un agente. Es posible enviar un mensaje a un agente en particular y también migrarlo de plataforma.

Otras facilidades son provistas por el agente Sniffer. Cuando un usuario decide usmear (sniffer) a un agente o grupo de agentes, el seguimiento de todos los mensajes que el agente envíe o reciba podrá ser hecho mediante la interfaz gráfica del agente; es posible también ver el contenido de cada uno de esos mensajes o guardarlo en archivo. El sniffer realiza su tarea en conjunto con el agente AMS, por cada mensaje que llega o sale del agente observado, se genera otro mensaje destinado al sniffer, teniendo como contenido el mensaje original; de esta manera a través de la interfaz gráfica del sniffer el usuario puede seguir los mensajes de/para un agente en particular.

El agente DF posee una interfaz gráfica que permite asociarse (to federate) con otros DFs y controlar (por ejemplo: registrar, quitar del registro, modificar o buscar por la descripción de los agentes) la red completa de DFs federados. Un DF se puede federar con otro simplemente registrándose en él. Desde esta interfaz es posible ver la lista de agentes registrados bajo un determinado DF. Permite ver el resultado de una búsqueda por descripciones dentro de un DF. También es posible visualizar la red de DF federados, en cuales DF está registrado un DF particular y que DFs están registrados bajo un DF. Las interfaces gráficas del sniffer y DF pueden ser comenzadas desde la ventana del agente RMA.

JADE facilita el desarrollo y depuración de sistemas multi-agente como el requerido dentro de nuestro ambiente colaborativo. La codificación de los agentes puede ser realizada por ejemplo, sin considerar los detalles del manejo de mensajes destinados a otros agentes ya que la plataforma lo resuelve. Con las facilidades disponibles para seguir el intercambio de mensajes, es mucho más sencillo verificar que se cumpla con la definición de las distintas conversaciones en las que se ven involucrados los agentes.

5. Conclusiones

En este trabajo se ha presentado una propuesta para la conformación de grupos colaborativos entre los miembros de una comunidad educativa *on-line*. En el marco de un ambiente colaborativo inteligente, que actualmente está en desarrollo, una de las tareas que se ha presentado como muy desafiante, es poder brindar a los usuarios facilidades para obtener ayuda de sus pares. En este sentido nuestra propuesta es organizar mediante una red de contratos a los agentes que representan a los usuarios dentro del ambiente. Esta organización que es una adaptación del protocolo *contract-net*, permite que en forma dinámica los usuarios puedan desempeñar diferentes roles durante el desarrollo de experiencias de aprendizaje colaborativo. Las consideraciones presentadas acerca de la plataforma JADE, muestran la factibilidad de la implementación del protocolo *contract-net* adaptado al ambiente colaborativo.

Referencias

- [1] G. Aguirre, M. Errecalde, and G. Leguizamón. Agentes y aprendizaje de máquina en optimización, control y entornos de enseñanza. In *Resúmenes del WICC'02*, 2002.
- [2] F. Bellifemine, A. Poggi, and G. Rimassa. Jade- a fipa-complaint agent framework. In *Proceedings of PAAM'99*, pages 97–108, 1999.
- [3] Jacques Ferber. *Multi-Agent Systems. An introduction to distributed artificial intelligence*. Addison-Wesley, 1999.
- [4] G. Leguizamón, M. Lucero, G. Aguirre, M. Chiarani, I. Pianucci, L. Manzur, and B. García. Nuevos entornos de aprendizaje. los ambientes colaborativos inteligentes. In *Resúmenes del WICC'02*, 2002.
- [5] G. Leguizamón, M. Lucero, G. Aguirre, M. Chiarani, I. Pianucci, L. Manzur, and B. García. Una propuesta para ambientes colaborativos inteligentes. In *Resúmenes del WICC'02*, 2002.
- [6] Jenny Preece. *Online Communities: Designing Usability and Supporting Sociability*. John Wiley, Publisher, 2001.
- [7] Reid G. Smith. The contract net protocol: high-level communication and control in a distributed problem solver. In *Readings in Distributed Artificial Intelligence*, pages 357–366, 1988.