

# ARQUITECTURA DISTRIBUIDA DE AGENTES PARA ENLACE DE INTERESES

Erick Muñiz T.                      José G. Escamilla                      Mario A. García E.  
emuniz@cia.mty.itesm.mx    jescamil@campus.ruv.itesm.mx    magarcia@cia.mty.itesm.mx  
Instituto Tecnológico y de Estudios Superiores de Monterrey  
Centro de Inteligencia Artificial  
Av. Eugenio Garza Sada 2501 sur Edificio CETEC torre sur  
64849 Monterrey N.L. Méx.  
Tel. (52 - 8)358 - 2000 Ext. 5136  
Fax (52 - 8) 328 - 4189

## RESUMEN

El objetivo de este trabajo es proponer una arquitectura que permita la creación de un entorno descentralizado de agentes que realicen la tarea de vinculación entre personas. Entre las principales ventajas están la difusión de trabajos de investigación, el aumento de la comunicación entre científicos y la integración de proyectos a distancia. Se presenta en este artículo una introducción, el desarrollo de dos arquitecturas para este problema, una arquitectura completamente distribuida y otra parcialmente distribuida. Se expone además los resultados obtenidos y la discusión, por último se realizan las conclusiones

## 1. INTRODUCCIÓN

Debido a la gran cantidad de información que se encuentra disponible en la super carretera de la información, los usuarios necesitan herramientas que les ayuden a filtrar esa información y les permita encontrar artículos de su interés [1].

Los sistemas de comunicación en Internet no han sido diseñados para la colaboración en trabajos a distancia [2]. Sólo unas cuantas personas se dan tiempo para escribir un artículo y difundirlo por correo electrónico, grupos de noticias o por medio de una pagina Web que generalmente es visitada por unas cuantas personas [3]. Esto ocasiona que dos personas quienes están trabajando sobre el mismo problema, o comparten un mismo interés, pueden no llegar a conocer los resultados de sus investigaciones [2].

Los agentes inteligentes son entidades

autónomas que actúan racional e intencionadamente con respecto a sus propias metas y al estado actual de su conocimiento[4]. Los agentes inteligentes ejecutan un conjunto de tareas de manera autónoma para sus usuarios en una forma segura y personalizada [3]. Estos agentes pueden ser programados por el usuario o usar técnicas de aprendizaje para descubrir cómo el usuario realiza las tareas y automatizarlas gradualmente. Ejemplos de este tipo de agentes incluyen programas de filtrado de correo electrónico, los cuales aprenden a evaluar cual correo es importante y cual no; programas de calendarización de juntas o concertación de citas, los cuales aprenden cuando y con quien establecer una cita y que tan flexibles deben ser en la negociación [2].

Uno de los campos de investigación sobre agentes inteligentes es el filtrado de información ya que el crecimiento constante en la cantidad de información que un usuario puede acceder a través de Internet hace que el proceso de búsqueda en un área de interés específica puede ser una tarea desgastante, tediosa y que toma demasiado tiempo [5;6].

**1.1. Motivación.** La motivación para la realización de este trabajo es el proponer una arquitectura de agentes que fortalezca la vinculación entre investigadores en áreas afines de forma que conozcan sus investigaciones y resultados de forma automática y en el momento en que ellos deseen compartirlos, sin necesidad de esperar hasta que sean publicados en algún congreso o publicación.

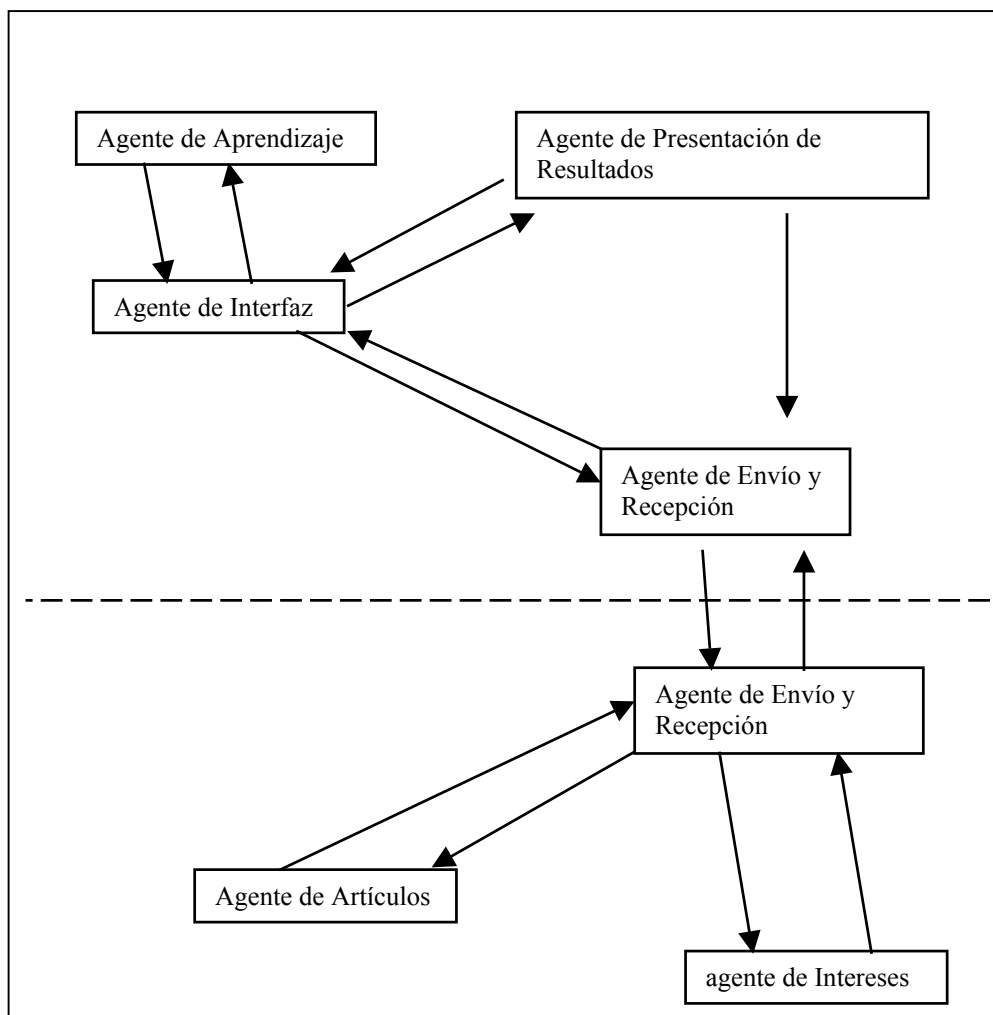


Fig.1. Arquitectura completamente distribuida

**1.2. Objetivo.** Proponer una arquitectura que permita la creación de un entorno descentralizado de agentes que realicen la tarea de vinculación entre personas o agentes con intereses afines. Entre las principales ventajas están la difusión de trabajos de investigación, el aumento de la comunicación y la integración de proyectos a distancia por medio del uso de tecnologías de filtrado de información.

## 2. DESARROLLO

En esta sección se presentan dos arquitecturas que pueden ser implementadas en esta aplicación, una arquitectura completamente distribuida y una arquitectura parcialmente distribuida.

### 2.1 Arquitectura completamente distribuida.

En principio se propone una arquitectura completamente distribuida la cual se muestra en la fig. 1. En esta arquitectura se tienen 7 agentes divididos en 4 agentes para el servicio del usuario y 3 agentes para el filtrado de la información

En esta primera aproximación se propone un agente para cada una de las actividades a realizarse. Dentro de los agentes que prestan servicios al usuario se proponen los siguientes:

- Agente de Interfaz. Este agente es el encargado de la interacción con el usuario y será el coordinador de los demás agentes de servicio al usuario.

- Agente de Aprendizaje. La tarea de este agente es aprender las áreas de interés del usuario revisando los artículos y páginas Web a compartir. Cada vez que el usuario desee compartir un nuevo artículo o un nuevo marcador (bookmark), el Agente de Interfaz activará al Agente de Aprendizaje para que realice su tarea.

- Agente de Envío y Recepción. Como su nombre lo indica este agente es el encargado de enviar los artículos que se desean compartir y recibir los artículos que lleguen de otros agentes.

- Agente de Presentación de Resultados. Este agente se activa cuando el Agente de Envío y Recepción avisa al Agente de Interfaz que un artículo ha llegado. El objetivo del Agente de Presentación de Resultados es obtener un resumen del artículo y enviarlo al Agente de Interfaz para ser desplegado.

Dentro de los agentes para el filtrado se tienen los siguientes:

- Agente de Envío y Recepción. Cumple con las mismas funciones del Agente de Envío y Recepción del usuario.

- Agente de Artículos. Este agente es encargado de almacenar y distribuir los artículos que los distintos usuarios han querido compartir.

- Agente de Intereses. La función de este agente es almacenar y distribuir los intereses de los distintos usuarios.

**2.2. Arquitectura parcialmente distribuida.** En la fig. 2 se muestra una arquitectura parcialmente distribuida, en esta se proponen 4 agentes, de los cuales 2 serán para servicio al usuario y dos para filtrar los artículos.

Para el servicio al usuario se proponen los siguientes agentes:

- Agente de Usuario. Este agente es el encargado de la interacción entre el usuario y el ambiente, además de coordinar a los demás agentes de servicio al usuario. Entre sus actividades principales están la recepción y envío de artículos de y hacia los agentes de filtrado, también avisa al usuario de que un nuevo artículo ha llegado y activa al Agente de Aprendizaje cuando el usuario desea compartir un nuevo artículo.

- Agente de Aprendizaje. Este agente aprenderá las áreas de interés del usuario revisando los

artículos y las páginas Web a compartir. Cada vez que el usuario comparta un nuevo artículo o un nuevo apuntador(bookmark), el Agente de Usuario activará al Agente de Aprendizaje para que realice su tarea. Una vez que el Agente de Usuario confirma el interés, el Agente de Aprendizaje envía el interés al Agente de Intereses.

Para el filtrado de artículos se proponen 2 agentes:

- Agente de Artículos. Se encarga de almacenar y distribuir los artículos que los distintos usuarios han querido compartir.

- Agente de Intereses. Este agente es encargado de almacenar y distribuir los intereses y las direcciones de los distintos usuarios.

### 3. RESULTADOS Y DISCUSIÓN

Aquí se presentan las ventajas y desventajas de la implementación de estas arquitecturas, además de un experimento para estimar el gasto de memoria de los agentes de servicio al usuario en las dos arquitecturas.

Al utilizar una arquitectura completamente distribuida para la solución de este problema se tienen las siguientes ventajas:

- Se relaja la solución del problema,
- Se distribuyen las actividades,
- Se pueden realizar las actividades del conjunto de agentes en forma paralela,
- Se crean agentes capacitados para una tarea específica,
- Se tiene un sistema multiagente flexible ya que sólo se requiere modificar un agente para una tarea específica, si se desea que el sistema multiagente realice otra tarea sólo se crea un nuevo agente,
- Se tiene un agente por cada una de las tareas a realizar.

Sin embargo, se observan las siguientes desventajas:

- Las tareas para esta aplicación no son descentralizadas por naturaleza pues de los 7 agentes propuestos, 4 agentes realizan sus tareas en la computadora del usuario, esto hace que el desempeño de la computadora del usuario

disminuya al tener a 4 agentes corriendo todo el tiempo en la misma computadora,

- Las tareas no necesariamente tienen que realizarse en paralelo,
- El tráfico de mensajes en la red aumenta considerablemente pues para cada actividad se requiere del envío de mensajes entre todos los agentes y
- El flujo de mensajes es muy alto e ineficiente pues se tienen 4 agentes en la misma computadora. Para establecer una comunicación entre ellos es necesario enviar sus mensajes por la red hasta donde esté localizado el ruteador de mensajes (Agent Message Router). Una vez que los mensajes están en el ruteador son regresados a la misma computadora en donde se encuentra el agente para quien iba dirigido el mensaje.

Utilizando una arquitectura parcialmente distribuida se tienen las siguientes ventajas con respecto a la arquitectura anterior:

- El número de mensajes que se envían a agentes que corren en la misma computadora disminuye en proporción de 7 a 2 mensajes al eliminar a 2 agentes,
- Se propone aprovechar la característica de comunicación para crear agentes que por sí mismos puedan comunicarse con otros agentes sin necesidad de tener un Agente de Envío y Recepción como intermediario,
- El tráfico de mensajes en la red disminuye considerablemente pues en la arquitectura distribuida se requieren en promedio 24 mensajes para que un agente reciba un artículo de otro agente con intereses comunes, en cambio con la arquitectura parcialmente distribuida se requieren en promedio 16,
- El desempeño de la computadora del usuario no se ve tan disminuido al correr sólo 2 procesos en lugar de 4 que se proponían anteriormente y
- Al tener capacidad de poder comunicarse con otros agentes sin intermediario, se tiene una distribución más natural en la que un agente es el encargado de interactuar con el ambiente y otro de aprender los intereses del usuario.

Para efectuar las pruebas del gasto de memoria de los agentes de servicio al usuario se realizó un sistema prototipo, el cual tiene las siguientes características: el sistema consta del Agente de Usuario y el Agente de Aprendizaje. El Agente de Usuario pide el interés al Agente de

Aprendizaje y en el Agente de Aprendizaje se ingresa el interés por medio del teclado. Estos agentes se implementaron utilizando el lenguaje JAVA y los paquetes contenidos en JATLite[7] que facilitan el desarrollo de agentes que utilizan KQML como protocolo de intercambio de mensajes.

La arquitectura completamente distribuida es simulada con la inclusión de otros dos Agentes de Usuario por lo que se supone que el Agente de Presentación y el Agente de Envío y Recepción son de tamaño similar al Agente de Usuario al igual que los mensajes que se envían.

En la gráfica 1 se muestra la comparación del uso de memoria virtual y el porcentaje de memoria física cargada, en ésta se observa claramente que al ejecutar los 4 agentes el uso de memoria aumenta en 7MB de memoria. Además conforme van ejecutándose más agentes, el tiempo de respuesta de la computadora se incrementa, lo que puede afectar el desempeño global del sistema.

#### 4. CONCLUSIONES

Para la implementación de ambas arquitecturas se dividieron las actividades a realizar en dos principales, servicio al usuario y filtrado, esto ocasionó resultados contrastantes entre estas dos divisiones.

En las actividades de servicio al usuario se observó que el uso de una arquitectura completamente distribuida puede hacer que los agentes sólo realicen una tarea muy particular y con esto no sobrecargarlos. Sin embargo, para esta aplicación se observa que utilizar esta arquitectura no es óptimo pues aunque las actividades pueden ser realizadas por varios agentes, es mejor agrupar las tareas para que varias tareas puedan ser realizadas por un solo agente pues, por el tipo de tareas que realizan, estos agentes deben de residir en la misma computadora.

Además el uso de un mayor número de agentes hace aumentar el gasto de memoria, de recursos de la computadora, tráfico en la red, y mal uso de la funcionalidad de los agentes pues no se aprovecha el hecho de que un agente está capacitado para enviar y recibir mensajes.

En lo que se refiere a actividades de filtrado se encontró que las tareas si son completamente separadas y pueden ejecutarse en paralelo, razón por la que en las dos arquitecturas persisten el Agente de Artículos y el Agente de Intereses. Si el número de usuarios aumenta se pueden incrementar el número de estos agentes. Por cada centro de investigación se utilizará un Agente de Artículos y un Agente de Intereses. Por último, el tener un agente para lo relacionado con los artículos y un agente para los intereses permite la flexibilidad de los filtros pues si después se quisieran compartir otro tipo de archivos tales como videos, música, etc, o noticias sólo se tendría que crear un agente especializado para

realizar esta actividad, el cual se comunicaría únicamente con el Agente de Intereses y a partir de ahí distribuiría sus archivos.

Es importante tomar en cuenta la naturaleza de las actividades y la necesidad de efectuar dichas actividades en paralelo para decidir sobre una arquitectura a utilizar.

Los resultados de este trabajo permitirán utilizar una arquitectura parcialmente distribuida en la elaboración del sistema que finalmente lleve a cabo la vinculación entre personas con intereses comunes.

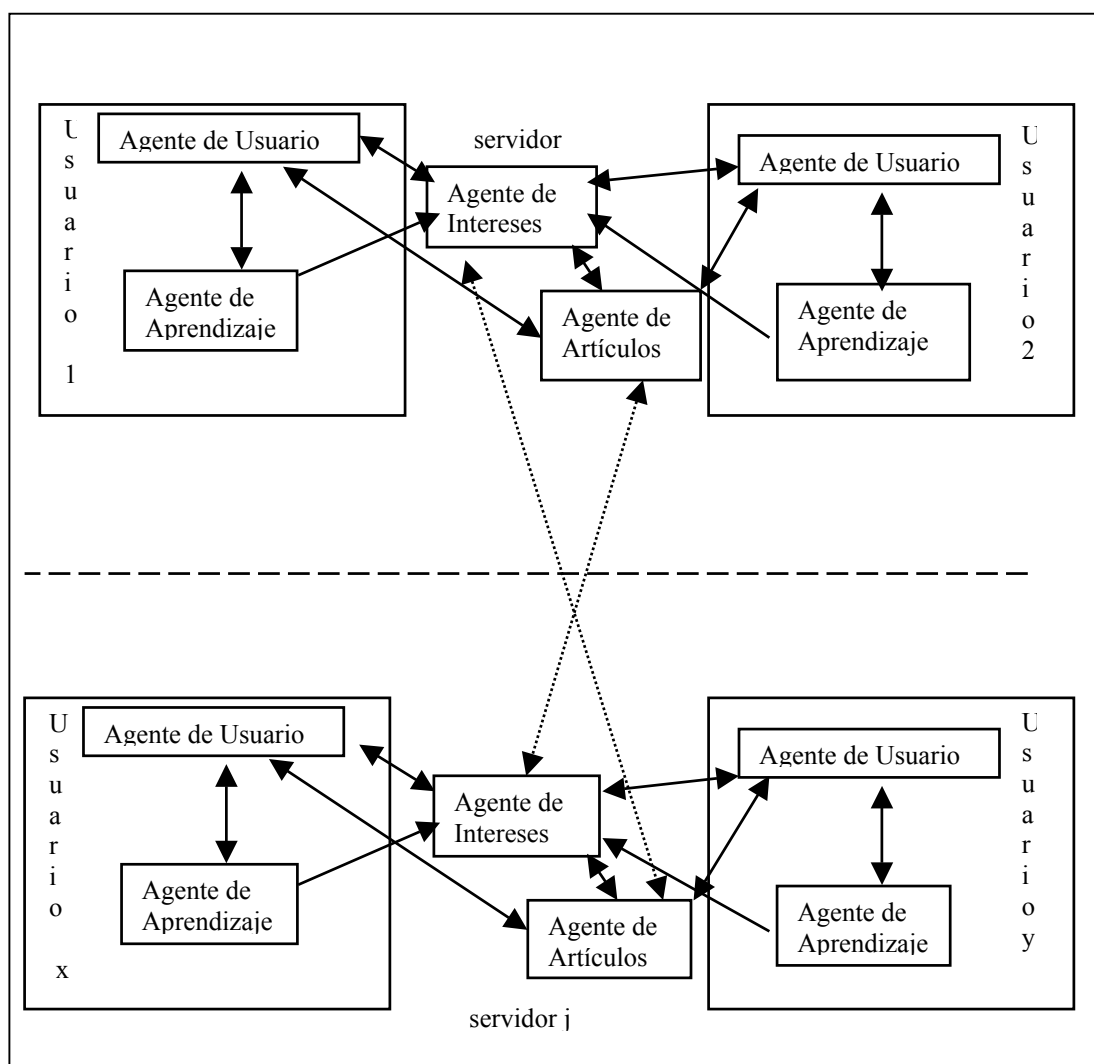
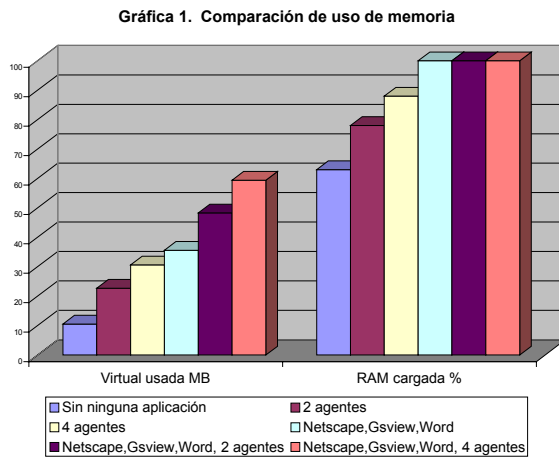


Fig. 2. Arquitectura parcialmente distribuida



[6]. García B. Alfredo 1996. "Sistema Multiagente para la Búsqueda de Información en Internet" Tesis para obtener el grado de M.C. Instituto Tecnológico y de Estudios Superiores de Monterrey. Monterrey N.L. Méx.

[7]. JATLite. <http://java.stanford.edu>

## 5. REFERENCIAS

[1]. Bradshaw Jeffrey, 1997. "Software Agents" AAAI Press / The MIT Press

[2]. Foner Leonard, 1996 "A Multi-Agent Referral System for Matchmaking", presented at The First International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, April '96. <http://agents.www.media.mit.edu/groups/agents/research.html>

[3]. Foner Leonard 1996 "A Security Architecture for Multi-Agent Matchmaking", presented at The Second International Conference on Multi-Agents Systems at Keihanna Plaza, Kansai Science City, Japan, December '96. <http://agents.www.media.mit.edu/groups/agents/research.html>

[4]. Demazeau Yves, Müller Jean-Pierre 1989 "Decentralized Artificial Intelligence" Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Cambridge, England. August 16-18, 1989

[5]. BussinessWeek, 1997. "A Way Out Of The Web Maze" <http://www.bussinessweek.com/1997/08/b35151.htm>