

Una arquitectura de sistema multi-agente para la recuperación y presentación de información.

V. Julian, C. Carrascosa, J. Soler
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n
46020 Valencia
☎ 96 387 73 50 Fax: 96 387 73 59
E-mail: {vinglada, carrasco, jsoler}@ dsic.upv.es

Resumen

En los últimos años el uso del paradigma agente/sistema multi-agente ha crecido notablemente. Este paradigma se ha venido aplicando a diferentes áreas, como pueden ser control de procesos, robots móviles y recuperación de información. En este artículo presentamos una arquitectura de sistema multi-agente que permite la recuperación de información existente en la red. Presentamos una arquitectura abierta y genérica, donde una de sus principales características es la independencia del agente de la propia dinámica de la red.

Palabras Clave: Agentes de información, arquitectura de sistemas multi-agente, Internet, recuperación de información.

Abstract

Over the last few years the use of the agent/multi-agent paradigm has grown up sharply. This paradigm has been applied to different fields, such as control processes, mobile robots, information retrieval and so on. In this paper, we present a system architecture based on the agent and multi-agent paradigm that allows us to retrieve and manage any kind of information from the net. We present our architecture like a generic and open system architecture. One of its main features is the agents' independence of the network's dynamic. We explain here, what has been made and what we have planned to do in the future.

Key Words: information agents, multi-agent system architecture, Internet, information retrieval

1. Introducción.

En la última década los métodos para gestionar y organizar la información han variado notablemente. Hoy en día, Internet se ha convertido en el gran vehículo o contenedor de información. A medida que ésta ha ido evolucionando y creciendo se ha complicado la forma de poder localizar y consultar la información disponible en la propia red. La mayoría de herramientas disponibles actualmente para la localización y recuperación de información en Internet se centran más en la cantidad que en la calidad de la propia información. Es habitual en cualquier proceso de recuperación que un usuario se sienta desbordado por el elevado volumen de documentos o de referencias que se le suministra. Resulta, en este caso, evidente la necesidad de una cierta automatización en el proceso que facilite la labor de localización de información de calidad y que a su vez permita diferenciar entre ella desde el punto de vista subjetivo del propio usuario. En este punto, el empleo de técnicas inteligentes resulta adecuado para la mejora de los resultados obtenidos.

En este artículo, presentamos una aproximación basándonos en el empleo de los paradigmas de agente y sistemas multi-agente para la gestión de la información existente en la red. El empleo de este paradigma se presupone como uno de los mecanismos con más futuro para el desarrollo de software [14]. El artículo se divide de la siguiente forma: en el punto 2 se presenta el concepto de sistema multi-agente y se hace una visión de trabajos anteriores en el área de agentes de información, en el punto 3 se detalla la arquitectura de sistema propuesta. Finalmente en el punto 4 se indican las conclusiones y se plantean los trabajos futuros.

2. Agentes de información.

El paradigma de sistemas multi-agente constituye actualmente un área de creciente interés dentro de la Inteligencia Artificial, entre otras cosas, por ser aplicable a la resolución de problemas complejos no resueltos de manera satisfactoria mediante técnicas clásicas. Los sistemas multi-agente se enfrentan al problema de la necesidad de comunicación y cooperación entre agentes autónomos, es decir, entidades cuyo comportamiento es guiado por ellos mismos. En los últimos años diferentes aplicaciones basadas en el paradigma de agente han ido apareciendo intentando buscar fundamentalmente una solución al problema de la explosión de la información. Unido a otros factores, como es el espectacular aumento del uso de la red, ha provocado que Internet haya sido y sea un excelente campo de pruebas para el desarrollo y generación de agentes.

Podemos encontrar varios trabajos recientes en este campo como los que comentamos a continuación. Amalthea es un sistema en el que se implementa toda una arquitectura de agentes, donde se distinguen dos tipos, agentes dedicados al filtrado y agentes encargados de descubrir la información [7] y que evolucionan según un algoritmo genético. Otro ejemplo es Webmate, este agente realiza búsquedas para un usuario y le facilita ayuda inteligente en su proceso de consulta de información, mediante recomendaciones y sugerencias [2]. Otros ejemplos de sistemas existentes son Syskill & Webert [9], Personal WebWatcher [6].

Los aspectos más interesantes de este tipo de agentes son la funcionalidad específica que se les da y el carácter personal. En lo referente a la funcionalidad, la imposibilidad de gestionar todo tipo de información suministrada por la red ha provocado que los agentes se especialicen en determinados temas ("DO I CARE" [12] se encarga de encontrar cambios significativos en páginas anteriormente visitadas, CiteSeer [1] se encarga de la búsqueda de artículos de investigación). Y en cuanto al carácter personal, los agentes asocian su comportamiento a los gustos de los usuarios a los que sirven.

3. Arquitectura propuesta.

Nuestra arquitectura multi-agente se nutre de algunas de las características de los sistemas ya desarrollados, como el método de aprendizaje o su marcado carácter personalizado. Sin embargo, se incluyen ciertos elementos que diferencian nuestro trabajo de lo visto hasta el momento. Nuestra arquitectura se puede ver como una arquitectura de dos niveles (tal y como se observa en la figura 1), por un lado a un primer nivel, el conjunto de agentes personales (a los cuales denominaremos agentes James) cada uno de ellos asistiendo a un usuario concreto. Comunicados con ellos ya sea a través de red local o de la propia Internet, tendremos al conjunto de agentes especialistas, 2º nivel de la arquitectura, pudiendo estar cada uno de ellos en una localización diferente. Un agente especialista desarrollará su función aplicando su conocimiento en la materia en la cual es experto, teniendo la posibilidad de conectarse para ello a las fuentes que sean necesarias por medio de la red.

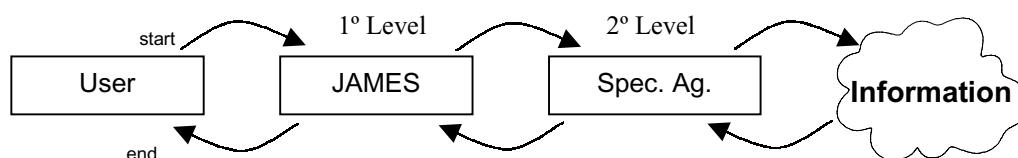


Figura 1. –Evolución de una petición de servicio.

Se ha intentado dotar al agente James de nuestra arquitectura de total independencia respecto a la propia dinámica inherente de la red, consiguiendo de este modo que el agente no tenga que ser modificado continuamente. Por otro lado, el servicio suministrado al usuario por un agente James no estará limitado, si no que dependerá del conjunto de agentes especialistas con los que el agente negocie. Con este último aspecto lo que se persigue es no limitar las actividades de un agente James, pudiendo éste evolucionar a medida que surgen nuevos servicios.

A grandes rasgos, la dinámica del sistema es la siguiente: cuando un agente James necesita la realización de algún tipo de servicio para satisfacer de esa forma a su usuario, deberá contactar (a través de un mecanismo de negociación) con aquel agente especializado que pueda facilitarle el trabajo. La figura 2 muestra la arquitectura básica del sistema propuesto, en ella podemos observar el conjunto de agentes James cada uno de ellos asignado a un usuario concreto y su comunicación con los diferentes agentes especialistas vía red (en principio se considera una conexión vía red local pero podría ser que los agentes especialistas estuviesen en localizaciones remotas).

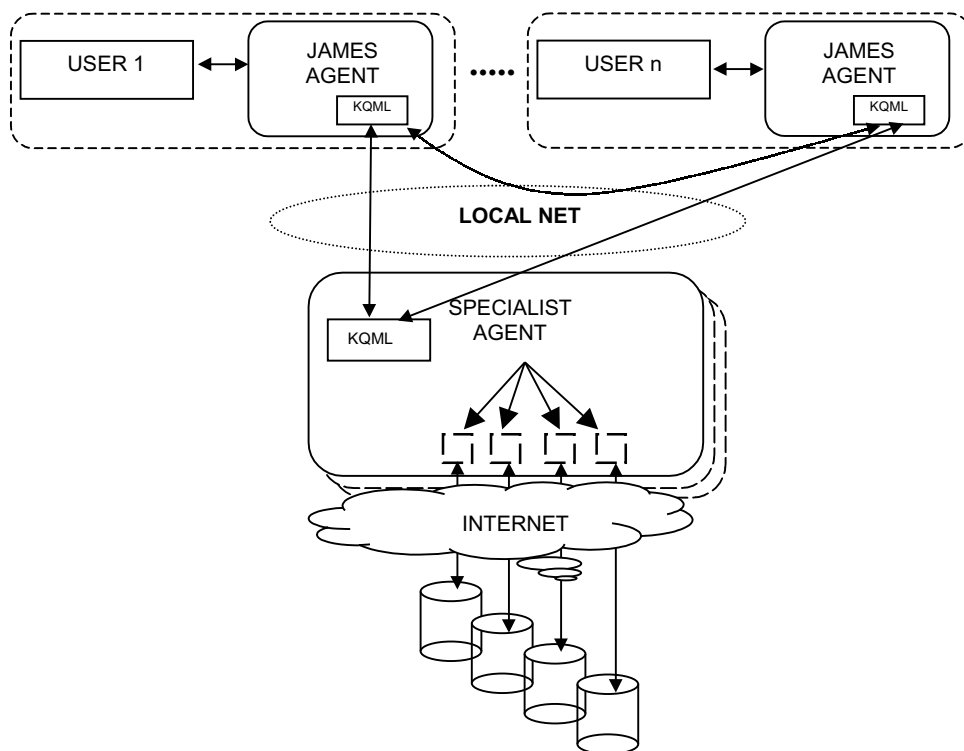


Figura 2. Arquitectura del sistema.

Aunque en principio presentamos una arquitectura compuesta únicamente por dos tipos de agentes, la idea es su desarrollo desde un punto de vista totalmente abierto, de forma que la inclusión o comunicación de otros agentes sea posible. La complejidad del entorno donde están situados estos agentes conlleva que el lenguaje de comunicación entre estos no se limite a un mero pase de mensajes. Los agentes pueden estar ejecutándose en redes lejanas, e incluso tener una representación del conocimiento diferente, que se deberá de trasladar a un término entendible por ellos. Hay que añadir que el sistema de agentes ira creciendo con nuevos agentes especialistas capaces de realizar servicios impensables a priori. Es necesario pues dotar al conjunto de un lenguaje que nos permita coordinar estos agentes.

EL lenguaje de comunicación escogido es KQML (Knowledge Query and Manipulation Language) [3]. El núcleo de KQML es un conjunto de "performatives" o acciones preestablecidas basadas en los principios de la teoría de habla (Speech Act Theory) [11] y que indican qué tipo de acciones se supone que debe de hacer el receptor de un mensaje. La complejidad de este conjunto permite que KQML soporte diferentes tipos de comunicación, tanto síncrona como asíncrona. Es necesario introducir un tipo especial de agente que se encargará de conocer cuales son los agentes disponibles y sus servicios, en definitiva pondrá en contacto al consumidor (james) y a los proveedores.

En los siguientes dos subpuntos pasamos a explicar con un poco más de detalle las características de los dos niveles de nuestra arquitectura, esto es, en que consiste un agente James y un agente especialista.

3.1 Agente personal (James)

El agente personal (JAMES) puede definirse como un agente interfaz de aprendizaje que según la definición dada en [5] es un programa que emplea técnicas de aprendizaje para asistir a un usuario que trate con una aplicación informática en particular. Para realizar dicha tarea el agente se divide en 4 módulos:

- Módulo de aprendizaje, este módulo se encarga de disponer de una visión de el entorno la cual puede irse modificando en función de la experiencia adquirida por el propio agente ya sea bien por interacción con el usuario, ya sea por la propia comunicación con otros agentes, este módulo lo podemos dividir en dos submódulos:
 - Submódulo del usuario, encargado de crear y mantener una representación del modelo de un usuario.
 - Submódulo de aprendizaje independiente del usuario, utilizado para mejorar la labor de obtención y filtrado de información atendiendo a criterios independientes del usuario, como por ejemplo la eficiencia de los agentes especialistas consultados.
- Módulo de filtrado, se encarga de filtrar la información, en principio de cualquier tipo, que le llega quedándose únicamente con la pertinente según el modelo de su usuario.
- Módulo de recuperación, se encarga de toda la labor de gestión de la información, es por tanto el módulo donde distinguiremos los diferentes servicios que James ofrece a su usuario.
- Módulo de comunicación exterior, encargado de la comunicación con cualquier otra entidad. En un primer momento estas entidades pueden ser otros agentes James o agentes especialistas.

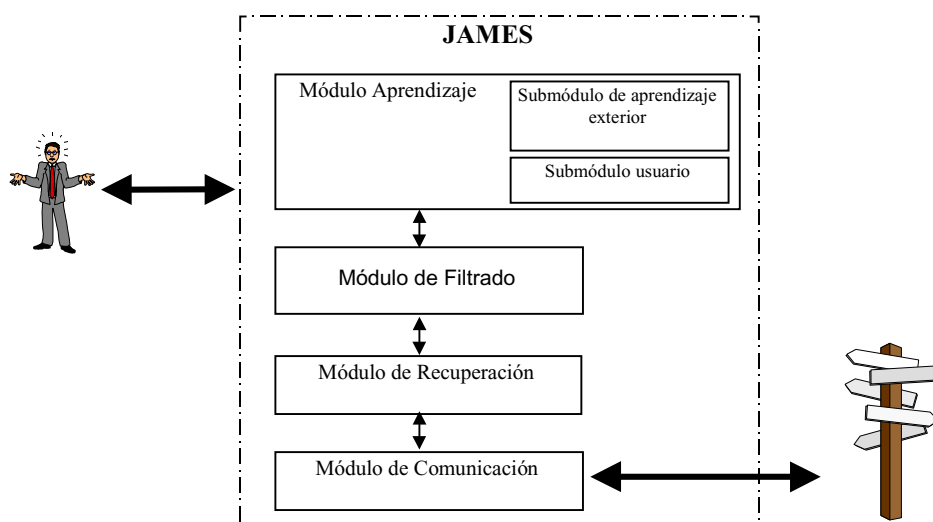


Figura 3. Arquitectura del agente mayordomo JAMES

En James quedan representadas las diferentes fases que caracterizan a un agente, esto es, una fase de percepción de su entorno (el entorno estaría representado por el usuario y por el conjunto de agentes con los que se comunica) donde el agente James mantiene una visión actualizada de su entorno y obtiene las peticiones del usuario. Por otra parte, existe una fase deliberativa donde dicho agente, en función de la labor a desarrollar, debe realizar un proceso de petición de servicios a agentes especialistas y de selección y filtrado de información sobre las respuestas obtenidas de los agentes especialistas consultados. Finalmente, una fase de actuación donde el agente personal responde a su entorno (el usuario) contestando a la petición realizada.

Se ha considerado el posible incremento de forma dinámica de los servicios, esta modificación se realiza a través del agente especialista, el cual será expuesto en el punto siguiente. La aparición de un nuevo agente de este tipo supone el lanzamiento de un mensaje

de oferta de servicio, que al llegarle a un agente JAMES provoca que éste decida si el servicio le interesa o bien directamente se lo pregunte al usuario. Si lo acepta y tras un proceso de negociación, el agente especialista suministrará a JAMES el código a incorporar, añadiendo este último una nueva opción en su menú de servicios.

3.2 Agentes especialistas

Los agentes especialistas se encuentran en el 2º nivel de la arquitectura multi-agente presentada. Sin un contacto directo con los usuarios, su cometido principal es servir las peticiones realizadas por los agentes personales. Para ello existirá como mínimo un agente especializado por cada uno de los servicios que el sistema oferta y que no realizan directamente los agentes personales. Un claro ejemplo de los servicios que aportan este tipo de agentes es el de búsqueda de información. Así, un agente especialista en búsquedas de información será un servidor inteligente de recopilación de información relacionada. Otros agentes especialistas podrían encargarse por ejemplo de la gestión inteligente de una base de datos, o del correo electrónico, etc.

Al igual que en el caso del agente James, un agente especialista también puede ser caracterizado como una entidad con tres etapas o fases, una fase de percepción (peticiones de servicios), una fase de desarrollo del servicio y una fase de actuación (respuesta a la petición realizada).

En un principio, en el sistema no está limitada la cantidad de agentes especialistas, pudiéndose encargar de diferentes servicios o bien ofrecer el mismo servicio y competir con el resto por ser los destinatarios de las peticiones de servicio por parte de los agentes mayordomo.

4. Conclusiones

Presentamos una arquitectura de un sistema multi-agente que nos permite integrar dentro de un mismo agente personal diferentes funcionalidades asociadas a la recuperación y filtrado de información en Internet y que a la vez variará de manera dinámica según el conjunto de agentes especialistas disponibles en cada momento.

Actualmente se está desarrollando un prototipo de la arquitectura presentada formado por un agente James y un agente especialista cuyo cometido es la búsqueda de referencias en Internet. El lenguaje de programación empleado es Java y como se ha comentado anteriormente, la comunicación entre el agente James y el agente especialista se realiza vía KQML. Los objetivos inmediatos son la puesta en marcha de nuevos agentes especialistas que compitan con el ya existente por la realización de servicios. Otro tema también pendiente es la ampliación de actividades por parte del agente James, que evidentemente, será tratado a medida que nuevos agente especilistas se hayan desarrollado y estén operativos para su empleo en el sistema.

Referencias

- [1] K. Bollacker, S. Lawrence, C.L. Giles, CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications, Agents'98, 2nd International ACM Conference on Autonomous Agents, p.116, 1998.
- [2] Chen, Liren; Sycara, Katia. Webmate: A personal Agent for Browsing and Searching, 1997
- [3] Finin, T; Fritzon, R; Mckay, D; McEntire, R. "KQML as an Agent Communication Language". Proc 3rd Int. Conf. On Information and Knowledge Management, 1994.

- [4] Huhns, Michael N.; Singh, Munindar P. Readings in Agents. Chapter 1, pages 1-24. ISBN 1-55860-495-2
- [5] Maes, Pattie; Kozierok, Robyn (1993): Learning Interface Agents. In: Proceedings of AAAI'93 Conference Washington, D.C. July 1993. 459-465.
- [6] Mladenic, D. Personal WebWatcher: Implementation and Design, Technical Report IJS-DP-7472, Department for Intelligent Systems, J.Stefan Institute, October, 1996.
- [7] Moukas, Alexandros. Amalthea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem. Proceedings of the Conference on Practical Application of Intelligent Agents & Multi-Agent Technology, London, 1996.
- [8] Nwana, H.S. Software Agents: An Overview. Intelligent Systems Research AA&T, BT Laboratories, Ipswich, United Kingdom, 1996.
- [9] Pazzani, Michael; Muramatsu, Jack; Billsus, Daniel. Syskill & Webert: Identifying interesting web sites. In AAAI conference, Portland, 1996
- [10] Salton, G.; McGill, M.J. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983
- [11] Searle, J. "Speech Acts". Cambridge University Press, 1969.
- [12] Brian Starr, Mark S. Ackerman, and Michael Pazzani. Do-I-Care: A Collaborative Web Agent, Proceedings of the ACM CHI'96 Conference, April, 1996.
- [13] M. Wooldridge and N. R. Jennings, editors. Intelligent Agents --- Theories, Architectures, and Languages. Volume 890 of Lecture Notes in Artificial Intelligence, Springer-Verlag, January 1995. ISBN 3-540-58855-8.
- [14] M. Wooldridge. Agents and software engineering. In AI*IA Notizie XI(3), pages 31-37, September 1998.