A Multiagent System Architecture for Retrieving and Showing Information

Vicente Julián, Carlos Carrascosa and José Soler

Dpto. sistemas Informáticos y Computación
Universidad Politécnica de Valencia

Vicente Julián is originally from Valencia (Spain) and received his BS and MS degrees in Computer Engineering from the Polytechnic University of Valencia in 1992 and 1995, respectively. He is a lecturer and a PhD candidate at the Computer Science Department at the Polytechnic University of Valencia, and his current research interests are in multi-agent systems, agent design, information retrieval and real-time systems.

Carlos Carrascosa was born in Valencia (Spain) in 1971. He received his Computing degree in Computer Science from the Polytechnic University of Valencia in 1995. Currently, he is a lecturer and a PhD candidate in the Computer Science Department at the Polytechnic University of Valencia. His research interests include multi-agent systems, learning, information retrieval and real-time systems.

José Soler received his BS and MS in Computer Science in 1993 and 1996 from the Polytechnic University of Valencia. He is now a doctoral candidate in Computer Science. He has been an Assistant Professor in the Computer Science Department at the Polytechnic University of Valencia since 1997. His research interests include multi-agent systems, distributed artificial intelligence and real-time systems.

Julián, V., Carrascosa, C., Soler, J. (2000). **A Multiagent System Architecture for Retrieving and Showing Information.** *Knowledge Organization, 27(1/2).* 10-16. 14 refs.

ABSTRACT: Over the last few years the use of the agent/multi-agent paradigm has grown sharply. This paradigm has been applied to different fields including control processes, mobile robots, and information retrieval. 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 Internet. We present our architecture as a generic and open system architecture. One of its main features is the agents' independence from the network's dynamic. We explain in detail what has already been done in our architecture as well as our future plans.

Keywords: information agents, multi-agent system architecture, Internet, information retrieval

1. Introduction

Over the last decade the way of managing and organizing information has notably changed. Today, the Internet is the most important vehicle for the transportation and storage of information. As the Internet evolves and grows, the act of locating and consulting relevant information available over the net becomes more and more complicated. In most cases, the toolkits for locating and retrieving information that are currently available are oriented towards offering the greatest quantity of references, but the relevance of resources is consistently overlooked. In most retrieval information processes, the user feels overwhelmed because of the high number of documents or references that are found, and the user is not capable of separating the relevant or interesting information (references,

https://doi.org/10.5771/0943-7444-2000-1-2-11 - am 13.01.2026, 15:38:43. https://www.inlibra.com/de/agb - Open Access -

papers, reports, graphs, e-mail, etc.) from the great amount of information offered.

There is, therefore, a clear need for a degree of automatization in the search and retrieval process. This automatization makes locating information easy and permits the analysis of the information by applying the subjective user's point of view. The application of intelligent techniques in complex problems with dynamic environments has increased. Two concepts of great importance which have arisen are the agent and multi-agent concepts. These concepts present an approach for solving complex and dynamic problems in a flexible way.

In this paper, we present an approach using the agent and multi-agent paradigm to manage and organize the information that is needed from the Internet. In this article, we focus exclusively on the system architecture. An agent system architecture focuses on an agent in the large (Huhns & Singh 1998). For the computer science community, the agent and multi-agent paradigms are perceived to be an important new development in software engineering (Wooldridge 1998). In section 2, we describe what we mean by the term "agent" and "multi-agent system". In section 3, we present previous work in this area. Section 4 shows our system architecture and finally, section 5 explains the conclusions and suggests directions for future work.

2. Agent and multi-agent systems

Many definitions of agent have been proposed in the literature, but none has been fully accepted by the scientific community. One relevant definition is that proposed by Wooldridge and Jennings (Wooldridge & Jennings 1995). According to them, an agent is defined by its *flexibility*, which implies that an agent is:

- *Reactive*: an agent must answer to the environment in which it finds itself.
- *Proactive*: an agent has to be able to try to fulfill its own plans or objectives.
- *Social*: an agent has to be able to communicate with other agents by means of some kind of language.

Thus, in order for a tool to be termed *agent,* the tool should be able to satisfy the above requirements. A small percentage of existing software fulfills this criteria.

In the most recent papers on the subject, there is a tendency to label an agent according to the role it plays. For instance, an agent whose task is oriented to the World Wide Web is known as an *information agent*. Agents are not usually developed in isolation but rather as parts of a multi-agent system (Huhns & Singh 1998). The agents interact with each other purposefully. Most purposeful interactions – whether to inform, query or decide – require that the agents talk to one another, be aware of each other and reason about each other. In other words, the agents must be developed to perform as members of a multi-agent system.

3. Information agents

Over the last few years, an increasing number of different applications based on the agent paradigm have appeared. Many of these attempt to find a solution for the explosion of information. Taking into account other factors, the Internet becomes an excellent proving ground for agent and multi-agent development.

The common tasks for these kinds of systems are:

- Information retrieval on the net
- Information filtering based on specific criteria

There are several recent works in this field. Amalthaea is a good example of a multi-agent architecture. In Amalthaea there are two types of agents. One of them filters information and the other searches for information on the web (Moukas 1996). The agents in this system are created and evolve through a genetic algorithm. Another good example of an information agent is WebMate. This agent searches for information based on a user request and helps the user in the information retrieval process. WebMate recommends and suggests the best information for a specific user (Chen & Sycara 1997). The process suggested in WebMate is based on the user's preferences. The learning algorithm used is called TF-IDF (Salton & McGill 1983). TF-IDF represents the documents through vectors in a n-dimensional space. This method is also used in other systems like Syskill & Webert (Pazzani, Muramatsu & Billsus 1996), Personal Webwatcher (Mladenic 1996) and Amalthaea.

An interesting characteristic of many of these systems is the specific functionality; in other words, the system can do only one thing. Until now, the management of all kinds of documents or information was impossible. This problem forces the agents to specialize their work according to a specific theme. Two examples are "DO I CARE" (Starr, Ackerman & Pazzani 1996), which is designed to visit Web pages previously found and detect interesting changes, and

CiteSeer (Bollacker, Lawrence & Giles 1998), an agent that automates the process of finding and retrieving scientific papers (in postscript format) on the Web. Finally, these architectures are characterized through their personal functionality. They work for a single user. Obviously, if the agent works to perform or facilitate the user's search on the net, the agent needs to know the user's normal behavior and reactions during the search process.

Our generic multi-agent architecture takes into account recent works in this field. We employ the same user learning method as in those works. One of the agents that we present has personal functionality. However, there are some differences between our work and the other approaches. We have tried to create a system that is independent from the inherent dynamic of the net. In our system a change in the net is not critical and therefore does not affect it. Also, the number of services offered by our architecture depends on the specialist agent collection in the system. With this approach, the personal functionality of the system can evolve in the future if new services appear.

4. Proposed architecture.

In our architecture, we distinguish between two kinds of agents. One type of agent works directly with the users attempting to supply them with different services. We have labeled this agent *James* (or *steward* agent). The other type of agent in the architecture is a *specialist* agent, which performs a specific service.

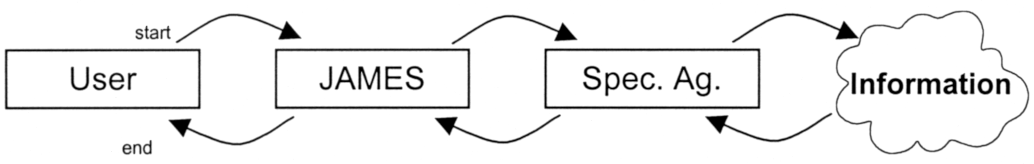

Fig. 1 Service Request Evolution

When a James agent needs to perform a certain kind of service, for example, a user's request, it contacts a specialist agent through a negotiation mechanism. The contacted specialist agent knows how to achieve the solution for the required service. In Figure 1, we see the evolution of a user request. Not all the cases are the same. The agent can be proactive, performing its own processes without any direct user supervision. This proactivity is based on its own knowledge and user preferences.

Figure 2 shows our general architecture schema. In this schema, an entire range of James agents is shown. Each one works for a specific user. The specialist agents (represented by a search specialist agent) are also shown in Figure 2. All the agents are connected through a local net or Internet. The specialist agents can be placed in different machines. They develop the assigned services by applying expert knowledge in the specific field and connecting to appropriate sources through Internet.

https://doi.org/10.5771/0943-7444-2000-1-2-11 - am 13.01.2026, 15:38:43. https://www.inlibra.com/de/agb - Open Access -

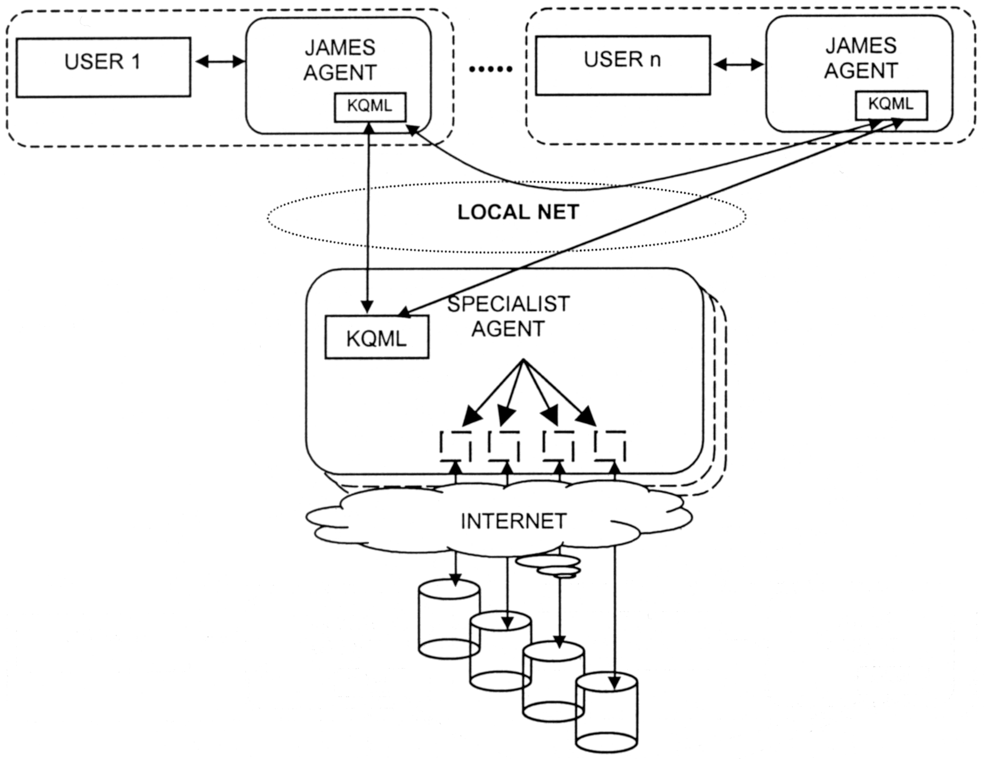

Fig. 2 System Architecture

Despite the fact that we only present two specific kinds of agents, the system is prepared to support other agent types. The system development is completely open. Therefore, inclusion or communication with other agents is possible.

4.1 Personal agent (James)

Inside our architecture, the James agent can be defined as a learning interface agent. According to Maes & Kozierok (1993), these agents are computer programs that employ machine learning techniques in order to provide assistance to a user dealing with a particular computer application. The James agent must be able to provide the required information to the user (it can be any kind of information and can be found anywhere). We divide James into four separate but related modules:

- The learning module: This module contains a representation of the environment. This representation can be changed for diverse reasons. The agent applies its own acquired experience. This experience is acquired by user interactions or by communication actions with other agents. For this reason, we can divide this module into a user submodule and a submodule which is independent from the user:
    - The user submodule makes and maintains a user model representation.
    - The independent user submodule is used to improve the retrieval and filtering process in accordance with independent user criteria. As an example of this action, the agent maintains information about the efficiency and quality of all the specialist agents consulted.
- The filtering module: This module filters the appropriate information. It does this the specifications of the user model representation mentioned above.
- The retrieval module: This module performs the entire information management process. The different services offered by James to the user are located here.
- The communication module: This module permits James to communicate with other agents (specialist agents, other James agents or other external agents).

James contacts specialist agents to compile information. The specialist agents supply specialized work in a specific field (search robots, local DB, news, mails, *etc.*). The most important feature is that the specialist agents perform the necessary processes without

https://doi.org/10.5771/0943-7444-2000-1-2-11 - am 13.01.2026, 15:38:43. https://www.inlibra.com/de/agb - Open Access -

user interaction. The actions are completely transparent for the personal agent and are user independent. The filtering action based on a user model starts when the James agent receives information. Finally, the information is shown or stored for future actions according to the service.

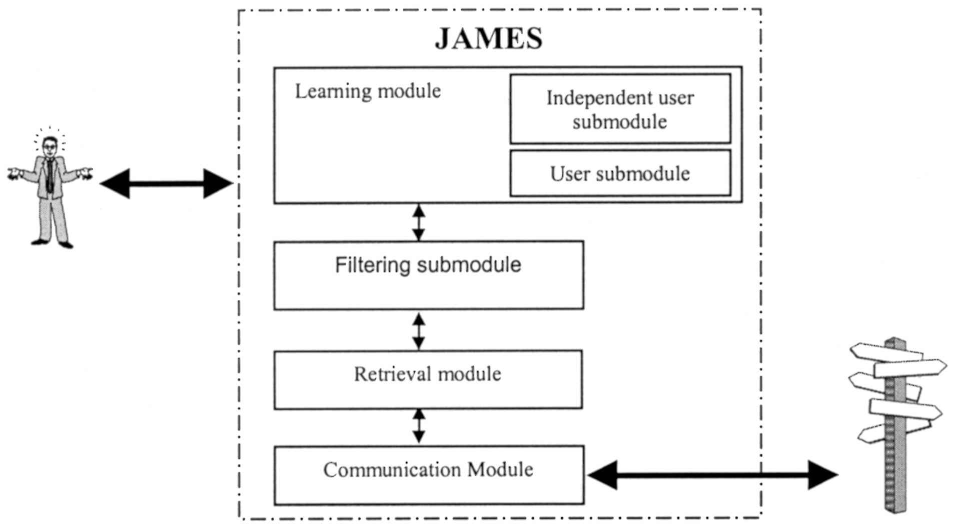

Fig. 3 James agent architecture

One of the most interesting features is the possibility of increasing the services in a dynamic manner. The service modification or increment is done by the specialist agent type. The characteristics of the specialist agents are explained below. When a new specialist agent appears in the system, it sends a subscription service message. If a James agent receives this message, James will decide the service interest of the new specialist agent. Assuming the James agent accepts the new service, the specialist agent supplies the needed code after a negotiation process in order to offer the service to the user.

4.2 Specialist agents (working agents)

The specialist agents are found on the second level of the multi–agent architecture and have no direct contact with the users. Their main purpose is to respond to requests made by the James agents. To do this, there will be at least one specialist agent for each one of the services that are not directly performed by the James agents. One clear example of the services that this kind of agent provides is the search for information. Other specialist agents could carry out the intelligent management of a database or of electronic mail. There is no limit to the number of specialist agents in the system, and they can carry out different services or offer the same service and compete with each other for the James agents' requests.

The main purpose is to provide a communication protocol between the specialist agents and the James agents and to open the system to the introduction of new specialist agents that use different search techniques or perform other services. As a working example of a specialist agent, the search agent we are developing works basically in the following way: each search request made to this agent will be distributed to a subset of all the search robots that exist in the network and that the agent knows. The agent performs the selection of what search robots it must use according to these two criteria:

- The subject of the search
- The efficiency conditions of the environment (greater speed, greater number of answers, etc.)

These two criteria are based on experience. Therefore, if the agent has not yet performed any search, it will launch the first search to all the search robots that it knows. As an example, the way the search specialist agent develops its process can be based on an evolutive algorithm.

4.3 Communication between agents.

The communication between agents is established at two different levels:

https://doi.org/10.5771/0943-7444-2000-1-2-11 - am 13.01.2026, 15:38:43. https://www.inlibra.com/de/agb - Open Access -

- The communication between James agents and specialist agents. The specialist agent performs a service for James by contract.
- The communication between different personal agents (James). The James agents share information about other agents or services.

Due to the complexity of this system architecture, the communication cannot be constructed by a set of predetermined messages. Agents are run on machines which are in different locations and are possibly on different networks. An agent may use different knowledge representations, which must be translated to enable meaningful conversation. It is important to remember that the system will be enhanced with new specialist agents and these agents will perform services that do not exist at this time. For this reason, the goal of the communication language is to achieve coordination between heterogeneous agents.

The communication language selected is KQML (Knowledge Query and Manipulation Language) (Finin *etal* 1994). The core of the KQML language is a set of *performatives* based on the principles of the speech act theory (Searle 1969). These performatives indicate what kind of action the receiver of a message is supposed to perform. This complex set enables KQML to support different types of communication. It also supports synchronous and asynchronous models.

The communication facilitator, which is a new kind of agent, must be present in this environment. The facilitator is endowed with the knowledge of who knows what, so it provides the necessary services between knowledge providers or specialist agents and consumers, or James agents.

5. Conclusions.

In this paper we have presented a system architecture that is based on the agent and multi-agent paradigm. This system architecture permits us to retrieve and manage any kind of information that comes to a user from the network.

This architecture is formed by two kinds of agents, the James agents (learning interface agents) and the specialist agents (information agents).

We are currently developing a prototype formed by a James agent and a specialist agent whose service is related to the search for references or documents from Internet. This prototype uses Java language. We are also planning to create new specialist agents. These new services will be added to the James agents.

References

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.

Chen, Liren; Sycara, Katia. Webmate: A personal Agent for Browsing and Searching

Finin, T; Fritzson, R; Mckay, D; McEntire, R. "KQML as an Agent Communication Language". Proc 3rd Int. Conf. On Information and Knowledge Management, 1994.

Huhns, Michael N.; Singh, Munindar P. Readings in Agents. Chapter 1, pages 1-24. ISBN 1-55860-495-2

Maes, Pattie; Kozierok, Robyn (1993): Learning Interface Agents. In: Proceedings of AAAI'93 Conference Washington, D.C. July 1993. 459-465.

Mladenic, D. Personal WebWatcher: Implementation and Design, Technical Report IJS-DP-7472, Department for Intelligent Systems, J.Stefan Institute, October, 1996.

Moukas, Alexandros. Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem. Proceedings of the Conference on Practical Application of Intelligent Agents & Multi-Agent Technology, London, 1996.

Nwana, H.S. Software Agents: An Overview. Intelligent Systems Research AA&T, BT Laboratories, Ipswich, United Kingdom, 1996.

Pazzani, Michael; Muramatsu, Jack; Billsus, Daniel. Syskill & Webert: Identifying interesting web sites. In AAAI conference, Portland, 1996

Salton, G.; McGill, M.J. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983

Searle, J. "Speech Acts". Cambridge University Press, 1969.

Brian Starr, Mark S. Ackerman, and Michael Pazzani. Do-I-Care: A Collaborative Web Agent, Proceedings of the ACM CHI'96 Conference, April 1996.

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.

M. Wooldridge. Agents and software engineering. In AI*IA Notizie XI(3), pages 31-37, September 1998.

https://doi.org/10.5771/0943-7444-2000-1-2-11 - am 13.01.2026, 15:38:43. https://www.inlibra.com/de/agb - Open Access -