PERFORMANCE EVALUATION OF ZEUS,J ADE,A ND SKELETONAGENT FRAMEWORKS

David Camacho, Ricardo Aler, César Castro, José M. Molina

Computer Science Department, Universidad Carlos III Madrid

Avda. Universidad nº 30,28 911 Leganés (Spain)

{dcamacho, molina}@ia.uc3m.es, aler@inf.uc3m.es

Abstract--

Due the growing interest in the field of the intelligent agents and multi-agent systems researchers have developed different toolkits. The aim of those toolkits, or frameworks, is to help the designers and engineers to build complex systems based on the agent concept. This paper presents a brief description of some of those frameworks: ZEUS, Jade and SkeletonAgent. These frameworks use its own agent architecture and other facilities like visual programming toolkits, documentation or reusable software libraries to facilitate the definition and development of multi-agent systems. The main aim of this paper is to compare the different frameworks in a common domain: to search news in several electronic newspapers. Because every one of those multi-agent toolkits use different features to buildt he whole multi-agent system, the behavior of any of those systems is expected to be different. The empirical evaluation measures the request time and the number of retrieved documents for the different systems. Finally, the paper discusses the conclusions for the previous experiments.

Keywords: **Multi-agent Systems, Intelligent Agents, Performance evaluation of multi-agent architectures.**

I. INTRODUCTION

Currently, there is an increasing interest in the research and development of intelligent agents. There exist different toolkits, frameworks, libraries, etc... that allow the designer to build the agento r agents that implement the functionality desired. More particularly, there are several frameworks that allow the designers to define the architecture of those agents, and the interrelations between them. This kind of systems, usually named Multi-Agent Systems (MAS), is a very active research field [6, 9].

It is possible to classify the different toolkits, or frameworks, to build MAS into two categories: Commercial and Research Products.

Comercial tools, like: AgentBuilder, Gossip (Tryllian), Intelligent Agent Factory, Jumping Beans, etc... need to be paid for (although there are free trial versions). However, there are several research tools that can be used to implement intelligent software agents or multi-agent systems like: Jade, Jafmas, MadKit, ZEUS, etc... Those tools have complete free versions of its software to implement agents. It is possible to find more information about these (and other) tools in:

- www.multiagent.com/Software/Tools_for_buildin g_MASs/index.html
- agents.umbc.edu/Companies/index.shtml
- http://www.agentbuilder.com/AgentTools/comme rcial.php

When a designer needs to build his/her own MAS, and it is possible to selecta mong severalo ptions to implement the system, it is necessary to evaluate the performance of those frameworks to select the most appropriate [8]. The main goal of this paper is to evaluate the performance of some popularf rameworks (like Zeus or Jade) in a specific domain.

Thisp aper is divided into 6 sections. Section 2 presents a brief description of the MAS frameworks analyzed. Section 3 describes the MAS implemented to test thep revious frameworks. Section 4 presents the experimental evaluation of the different toolkitsw hen the MAS built are used in a specific domain. Section 5 presents the conclusions of the paper. And finally, section 6 summarizes the future lines of work of the paper.

II. FRAMEWORKS AND TOOLKITS TO BUILD MULTI-AGENT SYSTEMS

Three different frameworks have being considering in this paper the next multi-agenta rchitectures:

- *ZEUS*. Developed by BT Laboratories in the Advanced Applications & Technology Department. (http://www.labs.bt.com/ projects/agents/ zeus/index.htm)
- *Jade*. Developed by Multimedia Technologies and Services department of CSELT (http://sharon.cselt.it/projects/jade)
- *SkeletonAgent*: Developed by the System Complex and Adaptive Laboratory (Scalab). (http:// scalab.uc3m.es/~agente)

Next sections will describe the main features for each framework.

A. *The ZEUS AgentB uilding Toolkit*

Theg oal of ZEUS project [1, 5], is to facilitatet he rapid development of new multi-agent applications by abstracting into at oolkit the common principles and components underlying some existing multi-agent systems. The main idea in ZEUS project is to create a relatively general purpose and customizable, collaborative agent building toolkit that could be used by software engineers with only basic competence in agent technology to create functional multi-agent systems. Thus, the ZEUS design philosophy encapsulates the following principles:

- The toolkit should clearly delineate between the *domain-level* problem solving and *agent-level* functionality.
- The use of the toolkit is based on the *visual programming* paradigm.
- The toolkit support an open design to ensure it is easily extensible.
- Finally, ZEUS try to utilize standardized technology wherever feasible, or be designed with standardization in mind.

When ZEUS is used to build a multi-agent system the requirements from the viewpoint of a user of the toolkit should be summarized in:

- *Configure* a number of different agents of varying functionality relationships.
- *Organize* the agents in whatever manner using system-supplied organizationalr elationships.
- *Imbue* each agent with selected system-supplied and/or user-defined communicative and coordination mechanisms.
- *Generate* automatically the executables for the agents.

Different assumptions are made whena ny software agent is build using ZEUS, this assumptions trying to facilitate, and also to describe the typical application domains of these agents. The principal assumptions made regarding the agent behavior are that the agents are:

- Deliberative, goal-directed and rational.
- Always truthful when dealing with other agents.
- Versatile, i.e. can have may goals and can engage in a variety of tasks.
- Temporally continuous.

The agents should bed eliberative in the sense, that they should explicitly reason about their actions in terms of what goals to pursue when to adopt new goals and when to abandon existing goals. In addition, the requirement for goal-directed behavior implies the agents only select actions that they expect in some way to advance the attainmento f their desired goals.

The main characteristics in the building process of a Multi-Agent System, in ZEUS, could be summarized in:

- This toolkit has user-friendly graphical interfaces that allow programming and debugging the agents of the system.
- The framework in ZEUS allows to the designer use different negotiation techniques tot est the implemented agents.
- ZEUS provides to engineers a complete reference about the architecture of the agents that this technology builds and the specifications about the multi-agents that it could be developed using the framework.

B. Jade

JADE (Java Agent Development Framework) is a software framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems througha middle-ware that claims to comply with the FIPA specifications [7]a nd through a set of tools that supports the debugging and deployment phase [2]. JADE agent platform tries to keep high the performance of a distributed agent system implemented with the Java language. In particular, its communication architecture tries to offer flexible and efficient messaging, transparently choosing the best transport available andl everaging state-of-the-art distributed object technology embedded within Java runtime environment. JADE uses an agent model and aJ ava implementation that offer a good runtime efficiency and software reuse. This agent model is more "primitive" than the agent models offered by others ystems, but such models can be implemented on the top of our " primitive" agent model [2]. This framework is built using the combination of two main products: a FIPA-compliant agent platform and a package to develop Java agents.

When this tool is selectedt o build the Multi-Agent System, it has both advantages and disadvantages that could be summarized in:

- Jade does not have a powerful programming environment, this framework only provides to the user a set of interfaces that allow him to debug the implemented agents.
- One of the better characteristics in Jade is that it has an excellent documentation, a good API to reuse the provided libraries to build new agents.
- Using Jade, a set of communication libraries (or packages) is provided to the software engineers, those libraries allow them to isolate the communication problem.

C. SkeletonAgent

This framework is built by a set of Java librariest hat allow the engineer to implement the desired agent. SkeletonAgent tries to wrap the "*agent concept*" into a set of reusable libraries[3, 4]. Those libraries should be used by the software engineers to develop the agent (or agents) with their own skills and the interrelation between them. Once those agents are built it is possible to develop the MAS that will be usedt o solve the problems.

When any Multi-Agent System is built using SkeletonAgent, it is necessary to build a set of predefined agents, these agents, and the architecture of the MAS, should be briefly described in:

- Control agents: the SkeletonAgent architecture is based in the "***team***" concept. All the agents in the system belong to one team, and this team are managing by a specific agent named CoachAgent, all the CoachAgent are managed by the ManagerAgent. The control agents manage different problems, like the insertion or deletion of them in the system.

- Execution agents: these agents are involved in solving the task. It is possible to define different kinds ofa gents, like UserAgents (that deals with the users), WebAgents (specialized in retrieve information from the Web), etc…

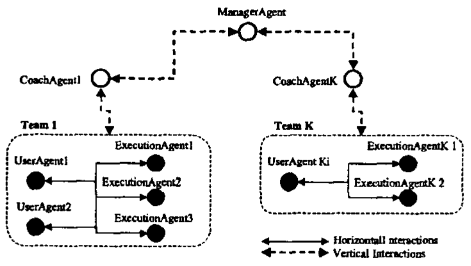

Fig. 1: Multi-agenta rchitecture defined by SkeletonAgent to implement a System.

The architecture provided by SkeletonAgent, see Figure 1,t ries to divide the manage problem in a MAS in two ways:

- First, with a vertical control of the insertion and deletion problems
- Second, with a horizontal relation among the execution agents that belong to a specific team.

This architecture tries to minimize the problems when any agent in the system is unreachable. The main conclusions when this architecture is used could be summarized as:

- The framework does not have graphical programming interfaces, and the debugging of the agents is hard.
- There is a poor documentation about the reusable libraries.
- The concept of agent is well encapsulated and it is possible to reuse a lot of code.

III. A Simple Multi-Agent System to Evaluate Agent Based Frameworks

This section describes the implementation of a simple multi-agent system that allows through theu se of a UserAgent to search for news in a set of electronic newspapers.

A. *MAS Topology*

A simple meta-search engine has been implemented to study the performance system within a particular problem. Int his paper a very simple topology was used (see Fig. 2).

This meta-search engine is built using a set of specialized agents to retrieve information from a set of electronic newspapers. It includes a UserAgenta nd six specialized WebAgents. Two of them offer information supplied by newspaper specialized in financial information (*Expansion_WebAgent* [1], *CincoDias_WebAgent* [2]), two in sports information (*Marca_WebAgent*[3], *Futvol_WebAgent*[4]), and finally two agents specialized in general information (*ElPais_WebAgent*[5], *ElMundo_WebAgent*[6]).
This MAS uses the six previous specialized agents and a UserAgent that make up the meta-search engine. The UserAgent has a graphical user interface (Figure 3) that allows provides:

- The question, or query, that the agents will use to search in the newspapers.
- The number of solutions requested.
- The agents thatw ill be consulted.

The interface used by thisa gent allowst o the user to know:

- The actual state of the agents (active, suspended, searching, finished)
- The messages and contents send among the agents

Finally the entire requests retrieved by the agentsa re analyzed (only the different request are taken into account) and the UserAgent builds an HTML file that will be shown to the user.

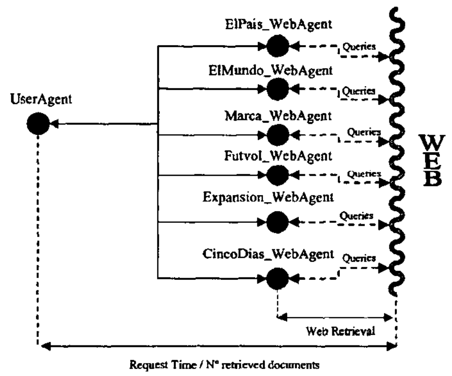

Fig. 2: Multi-agenta rchitecture of the meta-search engine.

[1] www.expansion.es
[2] www.cincodias.es
[3] www.marca.es
[4] www.futvol.com
[5] www.elpais.es
[6] www.elmundo.es

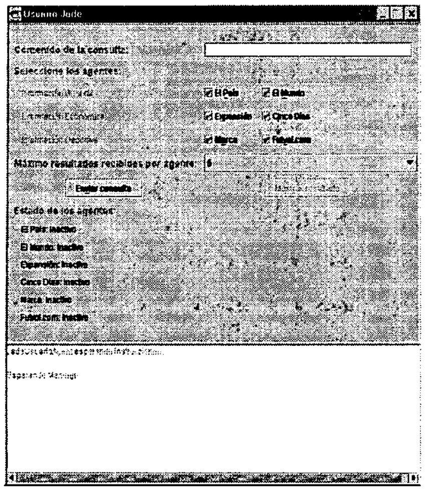

Fig. 3: UserAgentI nterface that allows interacts with the specialized Web Agents

B. Meta-search engine differences

Because the previous meta-search engine has been built using the different multi-agent frameworks, they have different features. All of them need to use the UserAgent and the six Web agents, but the different architectures may use other different agents to work correctly. The main differences should be summarized in:

- ZEUS meta-search engine needs to use the *ANServer agent* that implements the yellow pages for all the connected agents.
- Jade meta-search needs two agents to work, the *Agent Management System* that manage the insertion and deletion of the agents, and a *Directory Facilitator* that is used by all the agents to search for a specific agent with a desired skill.
- SkeletonAgent, as we describe briefly in the previous section, use two control agents: ManagerAgent and CoachAgent tom anage the coordinationa mongt he different agents in the team.

IV. Experimental Evaluation

The purpose of this section is to compare the previous frameworks performance using the meta-search engine developed.

A. Experimental Setup

Various empirical tests to evaluate the general performance of the MAS have been implemented. The next variables were measured:

- Independent variables:
  - Technology used: ZEUS, Jade, and y Skeleton.
  - Number of Web agents used: from only one Web agent (the Web agent that found the largest number of documents in the minimum amount of time), to the entire Web agents developed (six specialized Web agents).
  - Number of requested documents: from only one document to fifty documents to any active agent in the system (1, 5, 10, 15, 20, 30,4 0 and 50 news).
  - The query that will be searched by the agents. Due to the nature of the electronic newspapers, those queries were made about different subjects.

- The dependent variables:
  - Request time: when any question is sent to the multi-agent meta-search engine, it is measured for each technology the real time that the UserAgent spent to answer the question.
  - Number of articles retrieved.

The same questions were made to each configuration. The following tests were made:

- Configurations of the meta-search engine:
  - Only one Web agent (better Web agent in the retrieving news process)
  - Two Web agents specialized in different electronic newspapers. It wasu sed an agent specialized in general information (*ElPais_WebAgent*), and other agent specialized in financial information (*Expansion_WebAgent*).
  - Three Web agents: (*ElPais_WebAgent, Expansion_WebAgent, Marca_WebAgent*).
  - Finally, all the Web agents developed will be used in the most complex configuration analyzed.

A set of 648 queries to each architecture were made, so 2592 request were made to obtain the empirical results shown in the nexts ection.

B. EmpiricalE valuation

The experiments in this section display the average time to answer questions and the number of articles retrieved for each of the architectures.

Fig. 4 shows the performance for each architecture when the best Web agent (in the searching and retrieving process) is used. It cán be seen how ZEUS and JADE architectures obtain similar results (because the number of retrieveda rticles is equal for all the architectures). Therefore, only two agents are considered in the system, the WebAgent and the UserAgent[7]

[7] Some of those architectures also need other agents to work correctly.

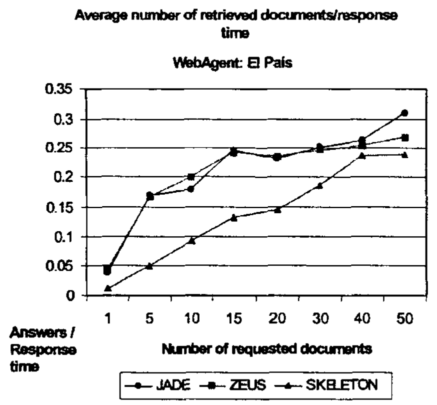

Fig. 4: Average number of retrieved documents/response time for each architecture using a single specialized Web agent.

Figure 5 showst he reply time obtained for the different questions asked by the user. Only two agents were present, one specialized in general information (ElPais_WebAgent) and one agent specialized in financial information (Expansion_WebAgent). This figure shows a better performance for the ZEUS architecture but the JADE architecture has a linear behavior.

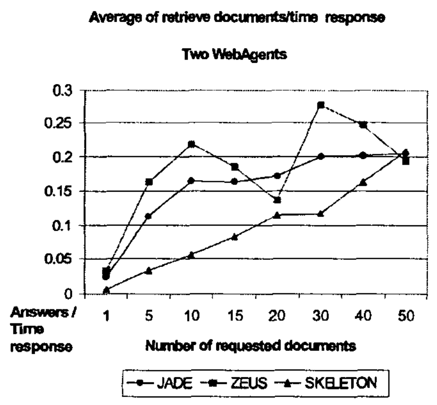

Fig. 5:A verage of retrieve documents/request time, using two specialized Web agents.

The empirical results shown in Figure 6 were obtained using the best specialized agent in the different information sources (ElPais_WebAgent, Expansion_WebAgent, Marca_WebAgent). This figure shows how the behavior for the JADE architecture is still very linear, and how the best performance is for ZEUS architecture. However, when it is the maximum number of possible documents is requested, SkeletonAgent obtains a the best performance.

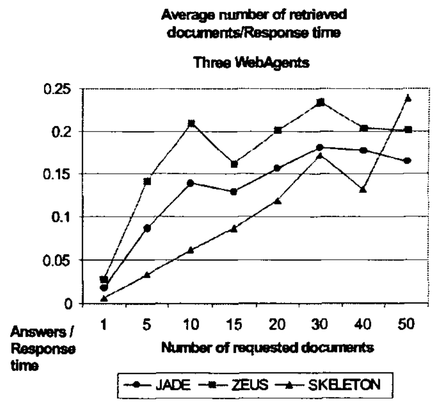

Fig. 6:A verage number of retrieved documents/response time, using three specialized Web agents.

Finally, Figure 7 shows the obtained performance when all the possible WebAgents are used. First, it can be seen observed a linear behavior for all them ulti-agent systems implemented, and the similarity when few documents are requested. However, for this configuration that uses:

- For ZEUS architecture eight agents (six WebAgents, one UserAgent, and the ANServer agent).
- JADE architecture that uses nine agents (six WebAgents, one UserAgent, a Directory Facilitator and the AgentM anager System).
- And finally, SkeletonAgent with nine agents (six WebAgents, one UserAgent, and two control agents)

When the number of documents request are more than twenty, the best performance shown is for this last architecture (SkeletonAgent).

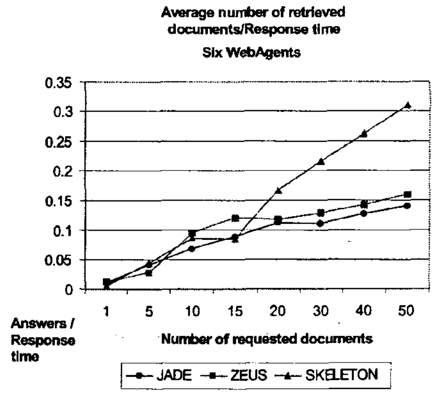

Fig. 7: Average number of retrieved documents/response time. All the Web agents are used.

V. Conclusions

There is an increasing number of toolkits and frameworks that could be used to implement software agent-based systems, and Multi-Agent systems. Any of those frameworks provide its own agent architecture and build-up methodology to deploy the systems. But it would be interesting to know which architecture performs better in different domains, to help software engineers to choose the most appropriate tool. This paper has shown how deploying a simple Multi-Agent system can be used to obtain the empirical evaluation of several frameworks that later will be used to compare them.

On the other hand, the empirical results show how the number of the agents used by the MAS could be used to evaluate its performance. It is interesting to remark:

- The deployed Multi-Agent Search engine is able to measure the different performance of the architectures or configurations used for any architecture.
- The ZEUS architecture has a better behaviour in those agent configurations with only a few agents.
- JADE implements Multi-Agent configurations with a regular behaviour.
- SkeletonAgent has a better performance only in the most complex configuration (using all the possible specialized WebAgents), when a lot of documents are retrieved. So the performance of this architecture rise when the number of agents and the communication among them grows.

Acknowledgments

The research reported here was carried out as part of the research project funded by CICYT TAP-99-0535-C02.

References

[1] Azarmi N.,T hompson S. ZEUS: A Toolkit for BuildingM ulti-Agent Systems. Proceedings of fifth annual Embracing Complexity conference, Paris April 2000.

[2] F. Bellifemine, A. Poggi, G.Rimassa. JADE- A FIPA-Compliant agentf ramework. Proceedings of the 4th International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-99). pp. 97--108, 1999.

[3] Camacho D., Molina J.M., Borrajo D. A Multiagent Approach for Electronic Travel Planning. Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2000), Austin, Texas, USA. August 2000. AAAI Press.

[4] Camacho D.,M olina J.M., BorrajoD ., Aler R. *MAPWEB*: Cooperation between Planning Agents and Web Agents. Information & Security:A n International Journal. Volume 7. 2001

[5]C ollis J., Ndumu D., Nwana H., Lee L.. The Zeus Agent Building Tool-Kit. BT Technology Journal 16(3), July 1998, p60-68.

[6] Ferber J. Multi-agent systems: an introduction to distributed artificial intelligence. Ed. by Addison Wesley, 1999.

[7] Foundation for Intelligent Agents (FIPA). Specifications 1998. Available from http://www.fipa.org/specifications.

[8] Wagner T., Rana O. Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems. International Workshop on Infrastructure for Scalable Multi-Agent Systems. Ed. by Springer-Verlag. Lecture Notes in Artificial Intelligence (LNAI 1887). Barcelona, Spain,2000.

[9] Wooldridge M. Multi-Agent Systems: an Introduction. Ed. by John Wiley, 2001.