DOI: https://doi.org/10.53555/nncse.v1i3.515 Publication URL: http://nnpub.org/index.php/CSE/article/view/515

JAVA AGENT DEVELOPMENT FRAMEWORK

Harsh Gautam[1]*, Mohit Sharma[2], Deepak Yadav[3]
*[1,2,3]*Computer Science, Maharshi Dayanand University*
*[1]*supercoolharsh007@gmail.com, [2]krazymohit90@gmail.com*
[3]depakaafriya@gmail.com

***Correspnding Author: -**
Emal ID - supercoolharsh007@gmail.com

Abstract: -

Java Agent Development Framework (JADE) is a software in JAVA, which is used for the development of agents, implemented in JAVA. It is a distributed agents platform, which has a container for each host where we run the agents. Additionally the platform has various debugging tools, mobility of code and content agents, the possibility of parallel execution of the behavior of agents, as well as support for the definition of languages and ontologies. In this paper, we will be having a detailed study on the functioning of JADE and we will put forward our conclusions on how its functioning can be enhanced.

Keywords: - *JAVA, agents, JADE, software, debugging*

I.

Distributed under Creative Commons CC BY-NC 4.0 OPEN ACCESS

II. INTRODUCTION

Agent-based technologies are still in their intial stages and rare truly agent-based systems have been realized. Agentbased technologies cannot realize their full potential, and will not become global, until standards to support agent interoperability are available and used by agent developers and adequate and appropriate environments for the development of agent systems are available.

In this paper, we present JADE (Java Agent Development Framework) that is a software framework to develop agent applications to be a good and a useful method for developing agents based technologies . JADE main features and, the architecture of the agent platform are discussed further in the paper.

III.Jade

JADE (Java Agent Development Framework) is a Java framework for the development of distributed multi-agent applications. It represents an agent middleware providing a set of available and easy-to-use services and several graphical tools for debugging and testing. One of the main objectives of the platform is to support interoperability by strictly following to the FIPA specifications concerning the platform architecture as well as the communication infrastructure. Moreover, JADE is very flexible, usable and adapted to be used on devices with limited resources such as PDAs and mobile phones.

JADE has been globally and widely used over the last years by many academic and industrial organizations varying from tutorials for teaching support in agent-related University courses to Industrial prototyping. As an example, White stein has used JADE to construct an agent-based system for decision-making support in organ transplant centers.

The JADE platform is open source software, distributed by TILAB (Telecom Italia Laboratories) under the terms of the LGPL license and can be obtained at http:// jade.tilab.com. Since May 2003, the International JADE Board has been responsible and the main reason for supervising the management of the project. Currently, the JADE Board consists of five members: TILAB, Motorola, Whitestein Technologies AG, Prefatory, and France Telecom.

IV. Internal architecture of the jade run-time

A JADE platform comprises of agent containers that can be distributed over the network. Agents live in containers which are the Java process that provides the JADE run-time and all the services required for hosting and executing agents. There is a special container, called the main container, which represents the bootstrap point of a platform: it is the first container to be launched and all other containers must join to a main container by registering with it. The programmer identifies containers by simply using a logical name; by default the main container is named 'Main Container' while the others are named 'Container-1', 'Container-2', etc. Command-line options are available to override default names.

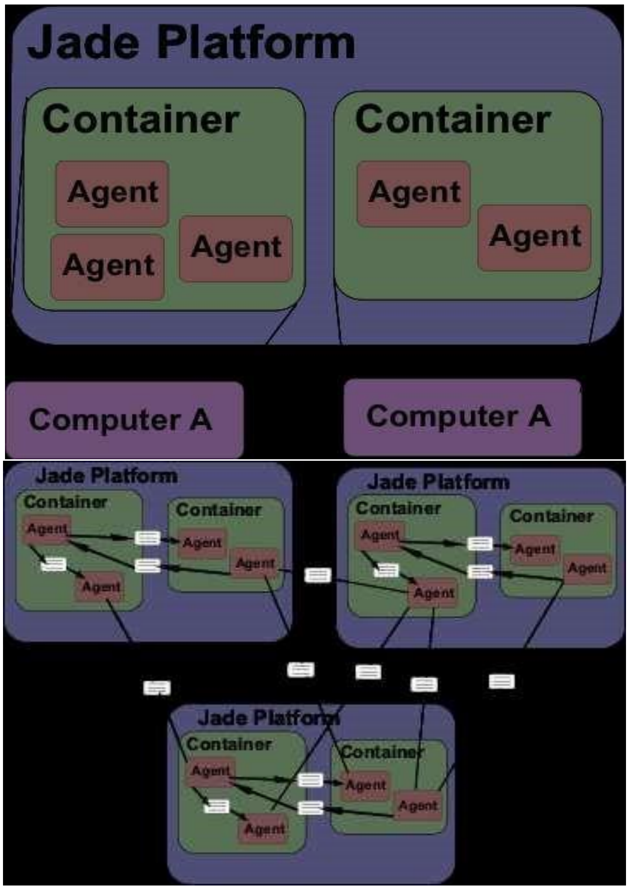

Fig. 1 JADE Architecture

Container is a running instance of the JADE running environment containing several agents. A single Main Container must always be active in a platform and all other containers register with it as soon as they start. You do not have to know how the JADE runtime environment works, but just need to start it before executing your agents.

The main container holds two special agents:

- Agent Management system (AMS) that provides the naming service; name, ensure uniqueness, create/destroy agents.
- Directory Facilitator (DF) that provides a Yellow Pages service by means of which an agent can find other agents providing the services he requires in order to achieve his goals.

The responsibilities of the main container are:-

- Managing the container table (CT), which is the registry of the object references and transport addresses of all container nodes composing the platform
- Managing the agent descriptor table (GADT), which is the registry of all agents present in the platform, including their current status and location • Hosting the AMS and the DF

IV. Conclusion

JADE is written and represented in Java language and is made by various Java packages, giving application programmers both ready-made pieces of functionality and abstract interfaces for custom, application dependent tasks. Java was the programminglanguage of preference because of its many attractive features, particularly geared towards object-oriented programming in distributed heterogeneous environments; some of these features are Object Serialization, Reflection API and Remote Method Invocation (RMI).

JADE provides its users with standard agent technologies while keeping runtime overheads low. The development of JADE is still continuing and further improvements, enhancements, and implementations have already been planned, included support for agent mobility as specified by FIPA98.

References

[1].Broadcom Eireann Research Ltd. ASL – Agent Service Layer. 1998. Available from p://www.broadcom.ie/asl.
[2].B. Eckel. Thinking in Java. , Upper Sandle River, NJ. Prentice Hall, 1998.
[3].T. Finin and Y. Labrou. KQML as an agent communication language. . J.M. In: Bradshaw (ed.), Software Agents, pp. 291-316. Cambridge, MA, 1997.
[4].Foundation for Intelligent Physical Agents. Specifications. 1997. Available from http://www.fipa.org
[5].M.R. Genesereth and S.P. Ketchpel. Software Agents. Comm. of ACM, 37(7):48-53.1994.
[6].A.D. Lux and D. Steiner. Understanding Cooperation: an Agent's Perspective, in Proc. ICMAS'95. San Francisco, USA. 1995.
[7].H.S. Nwana, D.T. Ndumu and L.C. Lee. ZEUS: An advanced Tool-Kit for Engineering Distributed Mulyi-Agent Systems. In: Proc. of PAAM98, pp. 377-391, London, U.K. 1998.
[8].Object Management Group. 95-11-03: Common Facilities RFP3 Final Draft. 1995.