Introduction to JADE (Java Agent Development Environment)

Ravi Ojha, Rahul Jha
Dronacharya College of Engineering

Abstract: **JADE is an enabling technology, a middleware for the development and run-time execution of peer-to-peer applications which are based on the agent paradigm and which can seamless work and interoperate both in wired and wireless environment. JADE is a middleware which facilitates the development of multi-agent systems under the standard FIPA for which purpose it creates multiple containers for agents, each of them can run on one or more systems. Is understood that a set of containers constitutes a platform.**

INDEXED TERMS: *JADE, Agent Management System, Message Transport System, Agent Execution Model.*

I. INTRODUCTION

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.

It 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 is an enabling technology, a middleware for the development and run-time execution of peer-to-peer applications which are based on the agents paradigm and which can seamless work and interoperate both in wired and wireless environment.

- distributed system topology with peer-to-peer networking
- software component architecture with agent paradigm

II. JADE FEATURES

JADE gives you the following features:

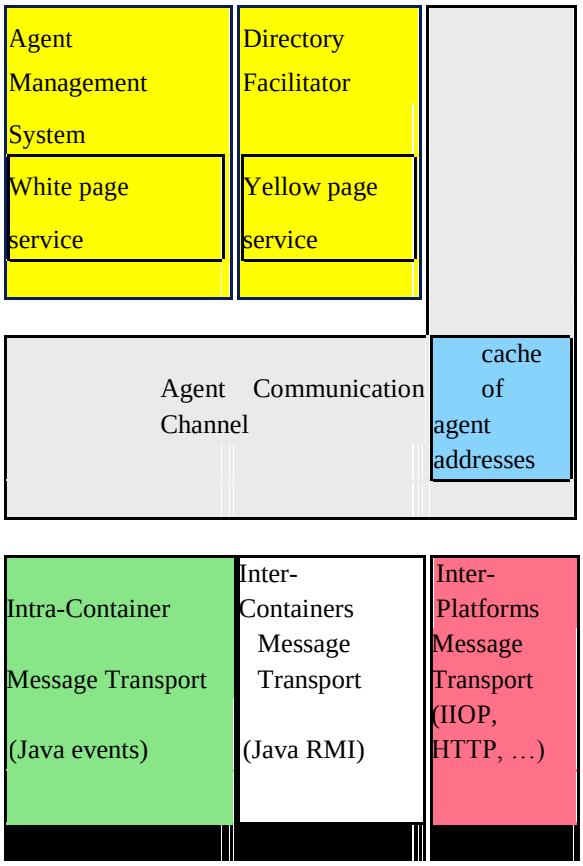
- FIPA-compliant distributed agent platform which can be split onto several hosts.
- Java Application Programmer's Interface.
- Library of FIPA interaction protocols, such as Contract Net, ready to be used.
- Graphical User Interface to manage several agents from the same Remote Management Agent.

III. JADE APPLICATION FIELDS

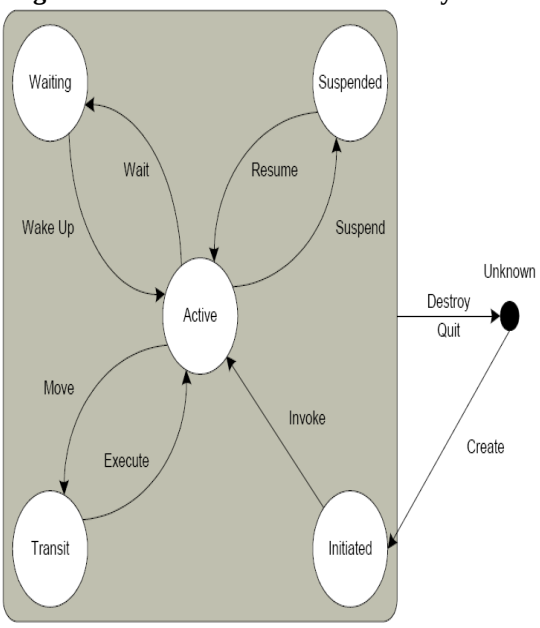
JADE can be used in the following fields, which is very useful for customer use and for the large company based use:

- Mobile Applications
- Internet Applications
- Corporate Applications
- Machine-to-Machine Applications

IV. INTERNAL ARCHITECTURE OF THE JADE RUN-TIME



1. Agent Life Cycle:



2. JADE - Message Transport Service:

- Controls the agent's private queue of ACL messages.
- Distributed Agent Communication Channel.
- The main container is not a bottle-neck, thanks to the distributed caches.
- HTTP MTP provided by EPFL under LGPL.
- String-based, XML-based (EPFL), bit-efficient (Sonera).

3. Some graphical tools to support development:

- RMA (Remote Monitoring Agent)
  - to browse the white-page service
  - to control the agent life-cycle (e.g. remote creation, agent migration, …)
  - to activate/deactivate MTPs on containers
  - to browse white-page services of remote agent platforms
- DF GUI
  - to browse the yellow-page service
  - to make DF federations and browse remote DF's

V. 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 programming language 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).

REFERENCES:

[1] Broadcom Eireann Research Ltd. ASL – Agent Service Layer. 1998. Available from http://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] M.R. Genesereth and S.P. Ketchpel. Software Agents. Comm. of ACM, 37(7):48-53.1994.

[5] Object Management Group. 95-11-03: Common Facilities RFP3 Final Draft. 1995.

[6] 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.

[7] A.D. Lux and D. Steiner. Understanding Cooperation: an Agent's Perspective, in Proc. ICMAS'95. San Francisco, USA. 1995.