 UNIVERSITY OF CALGARY	Course Number: SENG 609.22	Course Name: Agent-based Software Engineering
	Session: Fall, 2003	Department: Electrical and Computer Engineering
	Document Name: Infrastructure, APIs, and Software tools for agent-based system (An Overview of JADE)	Document Type: Tutorial Report

Information Collection and Survey
**Infrastructure, APIs, and Software Tools for Agent-based
Systems (An Overview of JADE)**

Course Instructor: Dr. Behrouz Homayoun Far
Student Name: Junling Huang
Date: December, 2003

Infrastructure, APIs, and Software tools for agent-based system (An Overview of JADE)

Abstraction

JADE is a software framework, which simplify the development of multi-agent systems while ensuring standard compliance through a comprehensive set of system services and agents in compliance with the FIPA specifications. It is fully implemented in Java language, and can be considered as an agent middle-ware that implements a software platform and a development framework. This paper highlights its compliance with FIPA, presents its architecture and main features, and outlines its communication and message transport service.

1 Introduction

Due to the efforts of researchers from various disciplines in the last few years, a number of tools and platforms are available that support activities or phases of the process of agent-oriented software development. Most of them are built on top of and integrated with Java. Apart from JADE (Java Agent Development Framework) [1], Some of the most prominent representatives are FIPA-OS [2], LEAP (“Lightweight Extensible Agent Platform”) [3], agentTool [4], JATLite (“Java Agent Template, Lite”) [5], and RETSINA [6].

In this paper, we present JADE (Java Agent DEvelopment Framework) that is a software framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. The next section discusses the compliance with FIPA specifications. Section three describes the architecture of JADE and its main tools and features. Because the communication is the central part in multi-agent system, the communication and transport message service is also outlined. Finally, section four concludes with a brief summary of JADE technical features.

2 Compliance with FIPA specifications

The Foundation for Intelligent Physical Agents (FIPA) [7] is an international non-profit association of companies and organizations sharing the effort to produce specifications of generic agent technologies. Among a variety of specifications that FIPA provided, two main standards have been complied within JADE:

- The FIPA reference model of an agent platform, which defines a set of normative and optional services and rules that allow a society of agents to exist, operate, and be managed. The normative services include: naming service and yellow-page service, message transport and parsing service, and life cycle management services. The JADE Agent Platform includes all those mandatory components that manage the platform, that is the AMS (Agent Management system), the ACC (Agent Communication Channel), and the DF (Directory Facilitator). AMS is the service that provides platform management functionality, such as monitoring agent lifecycles, and supervisory controls over access to and use of the platform; ACC is the service that provides the path for basic contact between agents inside and

outside the platform; DF is the service that provides a yellow page service (naming service) to the agent platform.

- The FIPA standards for communication model (as shown in Figure 1 [8]), which defines the structure of a message that allows to represent and convey information useful to identify sender and receivers, the content of the message and its properties (e.g. the encoding and the representation language), and in particular, the interaction protocols. This model has been fully implemented by JADE and its components have been clearly distinguished and fully integrated: interaction protocols, envelop, ACL, content languages, encoding schemes, ontology, and, finally, transport protocols. The transport mechanism, in particular, is like a chameleon because it adapts to each situation, by transparently choosing the best available protocol. Most of the interaction protocols defined by FIPA are available and can be instantiated after defining the application-dependent behavior of each state of the protocol. SL and agent management ontology have been implemented already, as well as the support for user-defined content languages and ontology that can be implemented, registered with agents, and automatically used by framework [1].

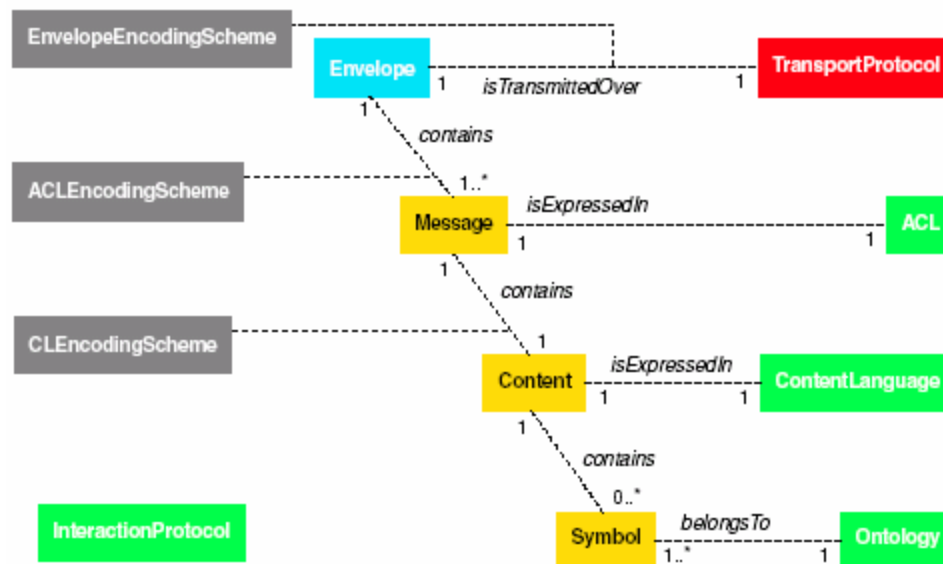


Figure 1 FIPA Standard: components of the communication model [8]

3 JADE

JADE (Java Agent DEvelopment Framework) is a software framework to make easier the development of multi-agent systems in compliance with the FIPA specifications. The goal of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents [9].

3.1 Main Features and Tools

To achieve its goal, JADE offers the following list of features and tools to the application developer:

- FIPA-compliant Agent Platform, which includes the AMS (Agent Management system), the DF (Directory Facilitator), and the ACC (Agent Communication

Channel). All these three agents are automatically activated at the agent platform start-up

- Distributed agent platform, which means the agent platform can be distributed on several hosts. Only one Java application, and therefore only one Java Virtual Machine (JVM), is executed on each host. Each JVM is basically a container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute on the same host.
- Transport mechanism and interface to send/receive messages to/from other agents, and FIPA97-compliant IIOP protocol to connect different agent platforms
- Provides a GUI for the remote management, monitoring and controlling of the status of agents, allowing, for example, to stop and restart agents. The GUI allows user to create and start the execution of an agent on a remote host, provided that an agent container is already running. The GUI also allows controls on remote FIPA-compliant agent platforms.
- Provides a GUI of the DF, by which user can view the descriptions of the registered agents, register and deregister agents, modify the description of registered agent, and also search for agent descriptions. It also allows user to federate the DF with other DF's and create a complex network of domains and sub-domains of yellow pages.
- Provides "The Dummy Agent", which is a simple yet very useful tool for inspecting message exchanges among agents. The dummy agent facilitates validation of an agent interface before integration into the MAS and facilitates interrogative testing in the event that an agent is failing. The graphical interface provides support to edit, compose and send ACL messages to agents, to receive and view messages from agents, and, eventually, to save/ load messages to/from disk
- Provides "The Sniffer Agent", allows to track messages exchanged in a JADE agent platform. When the user decides to sniff an agent, or a group of agents, every message directed to or coming from that agent, or group of, is tracked and displayed in the sniffer window. The user can view, save, and load, every message track for later analysis
- Provides "The Introspector Agent", allows to monitor and control the life-cycle of a running agent and its exchanged messages, both the queue of sent and received messages.
- JADE provides a homogeneous set of APIs that are independent from the underlying network and Java version. More in details, the JADE run-time provides the same APIs both for the J2EE, J2SE and J2ME environment, in theory, application developers could decide the Java run-time environment at deploy-time

- Open-source project that involve the contributions and collaborations of the user community

3.2 JADE Architecture

JADE includes both the libraries (i.e. the Java classes) required to develop application agents and the run-time environment that provides the basic services and that must be active on the device before agents can be executed. Each instance of the JADE run-time is called Container (since it “contains” agents). The set of all containers is called platform and provides a homogeneous layer that hides to agents (and to application developers also) the complexity and the diversity of the underlying tires (hardware, operating systems, types of network, JVM). As depicted in figure 2, JADE is also compatible with the J2ME CLDC/MIDP1.0 environment [8].

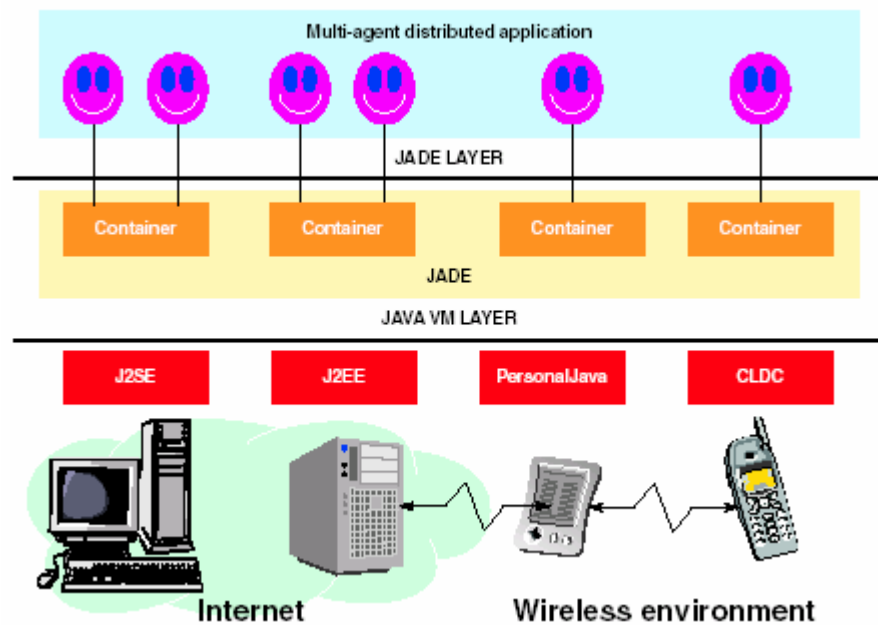


Figure 2 The JADE Architecture [8]

3.3 Communication and Message Transport Service

Communication plays a central role in multi-agent system, and it is also very complicated. JADE provides easy-to-use communications approach between distributed, heterogeneous entities that do not know anything about each other, and the structure of a message complies with the ACL language defined by FIPA.

In JADE, Agents send/receive Java objects that represent ACL messages, within the scope of interaction protocols [10]. In this way, the agent implementers only need to deal with the same class of Java object. JADE hides the complexity of all message coding (encoding/parsing), including the envelope level (String-based, XML-based), Agent Communication Language level (String-based, XML-based, bit-efficient), Content Language level (FIPA SL + API to register user-defined content languages), and Ontology level (FIPA-Agent Management, JADE Agent Management, API to register user-defined ontology). In addition, JADE also provides a library of common interaction protocols that users just need to implement the handle methods.

For message transport service in JADE, it designed as a chameleon that the transport mechanism is selected according to the situation and transparently choosing the best available protocol to achieve the lowest cost for message passing. Apart from that, JADE also provides distributed Agent Communication Channel that the main container is not a bottle neck because of the distributed caches provided. Java RMI, event-notification, HTTP, and IIOP are currently used, but more protocols can be easily added via the MTP and IMTP JADE interfaces.

4 Conclusion

JADE has been fully implemented in Java language, and it tries to provide its users with standard agent technologies compliance with FIPA, while simplifying the development of distributed applications composed of autonomous entities. It is a software framework that tries to hide all complexity of the distributed architecture, and by using it, application developers can only need to focus on the business logic of the application rather than on middleware issues. In addition, JADE provides a homogeneous set of APIs that are independent from the underlying network and Java version. This feature allows application developers to reuse the same application code both for a PC, a PDA or a Java-phone. Finally, JADE is an open-source project that involve the contributions and collaborations of the user community. This allows both users and developers to contribute with suggestions and new code, which guarantees openness and usefulness of the APIs.

5. References

- [1] JADE. <http://sharon.cselt.it/projects/jade/>, 1999.
- [2] FIPA-OS. <http://fipa-os.sourceforge.net/>, 2003
- [3] LEAP. <http://leap.crm-paris.com/>, 2003
- [4] agenTool. <http://www.cis.ksu.edu/~sdeloach/ai/agentool.htm>, 2003
- [5] JATLite. <http://java.stanford.edu/java agent/html/>, 2003.
- [6] RETSINA. <http://www-2.cs.cmu.edu/~softagents/>, 2003.
- [7] Foundation for Intelligent Physical Agents. <http://www.fipa.org>
- [8] F.Bellifemine, G.Caire, A. Poggi, G. Rimassa “JADE – A White Paper”
- [9] F. Bellifemine, G.Caire, A. Poggi, G. Rimassa “JADE – A FIPA-compliant agent frame work” *CSELT internal technical report*. April 1999, pagg.97-108
- [10] F. Bellifemine “JADE and beyond” *presentation at AgentCities Information Day 1*, Lausanne, Feb. 2002