

JAVA AGENT DEVELOPMENT FRAMEWORK

DIMITRY SALAI, ARKADII OSKIN
Polotsk State University, Belarus

JADE is a Java framework that allows implementation of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. It represents an agent middleware providing an Agent Platform and framework that simplifies development. This article presents the JADE software describing its intended uses and internal architecture. The main architectural issues and design decisions are discussed.

Introduction. The growth in networked information resources requires information systems that can be distributed on a network and interoperate with other systems. Such systems cannot be easily realized with traditional software technologies because of the limits of these technologies in coping with distribution and interoperability. The agent-based technology seems to be a promising answer to facilitate the realization of such systems because they were invented to cope with distribution and interoperability [5].

Agent-based technologies are still in their initial stages and only few truly agent-based systems have been realized. Agent-based 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 environments for the development of agent systems are available.

Several researchers are working towards the standardization of agent technologies (for example OMG [7] and FIPA [6]) and in the realization of development environments to build agent systems. Such development environments provide some predefined agent models and tools to make easy the development process. Moreover, some of them try to allow interoperability with other agent systems through the use of a well-known agent communication language. However, the use of a common communication language is not enough to easily support interoperability between different agent systems. The standardization work of FIPA is in the direction to allow an easy interoperability between agent systems, because FIPA, beyond the agent communication language, specifies also the key agents necessary for the management of an agent system, the ontology necessary for the interaction between systems, and it defines also the transport level of the protocols.

In this article, JADE (Java Agent Development Framework) is presented as a software framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. The goal is to describe JADE main features and, in particular, the architecture of the agent platform.

FIPA specification. The Foundation for Intelligent Physical Agents (FIPA) [6] is an international non-profit association of companies and organizations sharing the effort to produce specifications of generic agent technologies. FIPA is planned not just as a technology for one application but as generic technologies for different application areas, and not just as independent technologies but as a set of basic technologies that can be integrated by developers to make complex systems with a high degree of interoperability.

FIPA is based on two main assumptions. The first is that the time to reach consensus and to complete the standard should not be long, and, mainly, it should not act as a brake on progress rather than an enabler, before industries make commitments. The second is that only the external behavior of system components should be specified, leaving implementation details and internal architectures to agent developers. In fact, the internal architecture of JADE is proprietary even if it complies with the interfaces specified by FIPA.

The first output documents of FIPA, called FIPA97 specifications, specify the normative rules that allow a society of agents to inter-operate, that is effectively exist, operate and be managed. First of all they describe the reference model of an agent platform, as shown in Figure 1. Basically, it identifies the roles of some key agents necessary for the management of the platform, and specifies the agent management content language and ontology. Three key mandatory roles were identified into an agent platform. The Agent Management System (AMS) is the agent that exerts supervisory control over access to and use of the platform; it is responsible for authentication of resident agents and control of registrations. The Agent Communication Channel (ACC) is the agent that provides the path for basic contact between agents inside and outside the platform; it is the default communication method which offers a reliable, orderly and accurate message routine service; it must also support IIOP for interoperability between different agent platforms. The Directory Facilitator (DF) is the agent that provides a yellow page service to the agent platform. Notice that no restriction is given to the actual technology used for the platform implementation: e-mail based platform, CORBA based, Java multi-thread applications and other could all be FIPA compliant implementations.

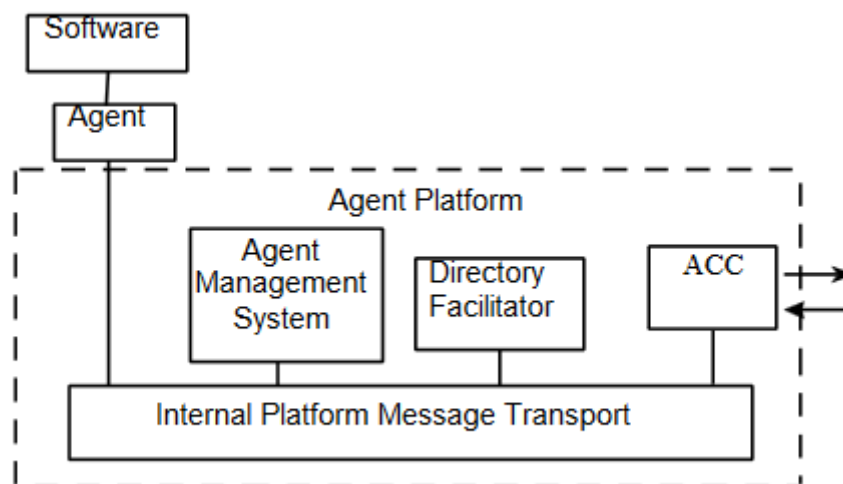


Fig. 1. FIPA reference model of an Agent Platform

JADE

JADE (Java Agent Development Framework) is a software framework to make easier the development of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. The goal of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents. To achieve such a goal, JADE offers the following list of features to the agent programmer:

- 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. The agent platform can be split on several hosts (provided that there is no firewall between them). Only one Java application, and therefore only one Java Virtual Machine, is executed on each host. Agents are implemented as one Java thread and Java events are used for effective and light-weight communication between agents on the same host. Parallel tasks can be still executed by one agent, and JADE schedules these tasks in a more efficient (and even simpler for the skilled programmer) way than the Java Virtual Machine does for threads;
- a number of FIPA-compliant DFs (Directory Facilitator) can be started at run time in order to implement multi-domain applications, where the notion of domain is a logical one as described in FIPA97;
- programming interface to simplify registration of agent services with one, or more, domains (i.e. DF);
- transport mechanism and interface to send/receive messages to/from other agents;
- FIPA97-compliant IOP protocol to connect different agent platforms;
- light-weight transport of ACL messages inside the same agent platform, as messages are transferred encoded as Java objects, rather than strings, in order to avoid marshalling and unmarshalling procedures. When sender or receiver do not belong to the same platform, the message is automatically converted to/from the FIPA compliant string format. In this way, this conversion is hidden to the agent implementers that only need to deal with the same class of Java object;
- library of FIPA interaction protocols ready to be used;
- automatic registration of agents with the AMS;
- FIPA-compliant naming service: at start-up agents obtain their GUID (Globally Unique Identifier) from the platform;
- graphical user interface to manage several agents and agent platforms from the same agent. The activity of each platform can be monitored and logged.

Architecture of the Agent Platform

The JADE Agent Platform complies with FIPA97 specifications and includes all those mandatory agents that manage the platform that is the ACC, the AMS, and the DF. All agent communication is performed through message passing, where FIPA ACL is the language to represent messages.

The software architecture is based on the coexistence of several Java Virtual Machines (VM) and communication relies on Java RMI (Remote Method Invocation) between different VMs and event signaling within a single VM. Each VM is a basic container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute on the same host. In principle, the architecture allows

also several VMs to be executed on the same host; however, this is discouraged because of the increase in overhead and the lack of whatever benefit.

Each agent container is a multithreaded execution environment composed of one thread for every agent plus system threads spawned by RMI runtime system for message dispatching. A special container plays the front-end role, running management agents and representing the whole platform to the outside world. A complete Agent Platform (AP) is then composed of several agent containers. Distribution of containers across a computer network is allowed, provided that RMI communication between their hosts is preserved. A special light-weight container is being implemented for the execution of agents within a Web browser.

There are different packages present in JADE for example

- jade.core.event: implements the distributed event notification service;
- jade.core.management: implements the distributed agent life-cycle management service;
- jade.core.messaging: implements the message distribution service;
- jade.core.mobility : implements the mobility and cloning service;
- jade.content and its sub-packages contain the collection of classes that support creating and manipulating expressions according to a given content and ontology;
- jade.domain contains the implementation of the AMS and DF agents.

Conclusions. JADE is written and represented in Javalanguage and is made by various Javapackages, 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).

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

REFERENCES

1. Создание распределенных информационно-управляющих систем на базе агентно-ориентированного подхода [Электронный ресурс] // ст. XII Всерос. совещание по проблемам управления, Москва, ИПУ РАН, 16-19 июня 2014 г. – Режим доступа: <http://vspu2014.ipu.ru/proceedings/prcdngs/8984.pdf>. – Дата доступа: 02.11.2014.
2. Об одном универсальном методе построения моделей для сложных много-агентных систем [Электронный ресурс] / Компьютерные исследования моделирование. – Режим доступа: http://crm.ics.org.ru/uploads/crmissues/crm_2013_4/13401.pdf. – Дата доступа: 02.11.2014.
3. Особенности разработки мульти-агентных систем на JADE Semantic Agent Framework [Электронный ресурс]. – Режим доступа: http://www.ssc.smr.ru/media/ipuss_conf/15/6_04.pdf. – Дата доступа: 02.11.2014.
4. Леонтьев, А.А. Эволюция информационных технологий [Электронный ресурс] / А.А. Леонтьев // Интернет и библиотека: триумф Интернета или несбывшиеся надежды. – Режим доступа: http://fulltext.pl.spb.ru/unor/metod/NTB_2012_10_35.pdf. – Дата доступа: 02.11.2014.
5. Genesereth, M.R. Software Agents / M.R. Genesereth and S.P. Ketchpel // Comm. of ACM, 37(7):48-53.1994.
6. Foundation for Intelligent Physical Agents. Specifications. 1997. – Mode of access: <http://www.fipa.org>.
7. Object Management Group. 95-11-03: Common Facilities RFP3 Final Draft. 1995. – Mode of access: <http://www.omg.org/docs/1995/95-11-03.ps>.
8. Agent-based control framework in JADE. – Mode of access: http://www.scs-europe.net/dlib/2014/ecms14papers/abs_ECMS2014_0132.pdf. – Date of access: 02.11.2014.
9. Analysis of RDBMS and semantic WEB search in university system (International Journal of Engineering Sciences & Emerging Technologies, Oct.2014.). – Mode of access: <http://www.ijeset.com/media/0003/8N18-IJESET0702913-v7-is2-604-621.pdf>– Date of access: 02.11.2014.
10. Mining in ontology with Multi Agent system in semantic web (The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.5, October 2014) – Mode of access: <http://airccse.org/journal/jma/6514ijma04.pdf>. – Date of access: 02.11.2014.