

Software Engineering of Distributed Systems, KTH

A Brief Comparison of Jade and Zeus

[Distributed AI and Intelligent Agents, Homework 1, Question 4]

A Brief Comparison of Jade and Zeus

1. Prelude

Agent platform is a technological architecture providing the environment in which agents can actively exist and operate to achieve their goals. The agent platform may additionally support the development of agents and agent based applications.

In this paper, we consider two agent platforms Jade and Zeus. JADE is a software framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. Whereas Zeus is an Open Source agent system entirely implemented in Java, developed by BT Labs and can be considered a toolkit for constructing collaborative multi-agent applications.

2. Architecture of Platform

Jade software architecture is based on the coexistence of several JVM and communication relies on Java RMI 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. A single special Main container is always active in the platform, which not only accepts registrations from other containers, but also holds two special agents, AMS and DF.

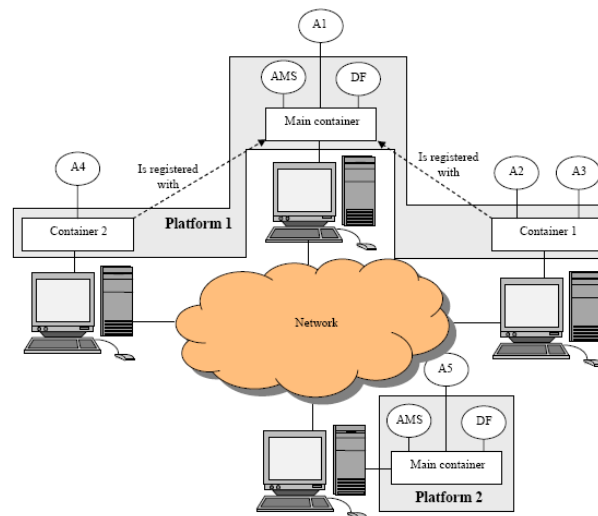


Figure 1. Jade Architecture

The Zeus toolkit consists of a set of components, written in the Java programming language, which can be categorized into three functional groups as depicted in Figure 2: an agent component library, an agent building tool and a suite of utility agents comprising nameserver,

facilitator and visualiser agents. The principle underlying the Zeus toolkit is that application-specific agents can be constructed by configuring the generic Zeus agent, and equipping it with the necessary application functionality. It provides high-level agent development approach that hides the complexities of the Agent Component Library from the agent developer.

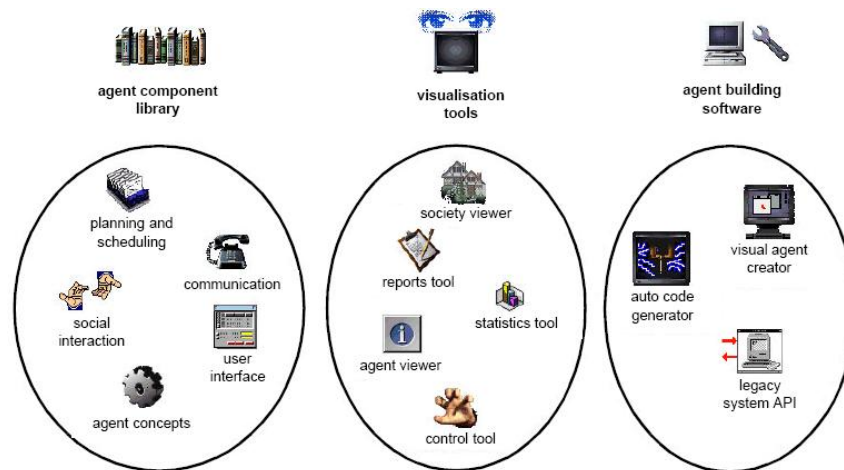


Figure 2. Zeus Architecture

3. Services provided by Platform

The services provided by each platform can be categorized and compared by table 1.

Table 1. Services comparison of Jade and Zeus

	Jade	Zeus
Supported Platform	Java 2	Java 2
Implemented standards	FIPA, work with CORBA (Orbacus)	FIPA
Communication	ACL, support for inter-platform messaging with plug-in MTPs (RMI, IIOP are ready and HTTP, WAP are already scheduled), ACL and XML codec for messages.	KQML and ACL
Mobility	weak mobility	no
Security	JADE Object Manager provides connection authentication, user validation and RPC message encryption. The JADE socket proxy agent acting as a bidirectional gateway between a JADE platform and an ordinary TCP/IP connection,	ASCII-encoded, Safe-Tcl scripts or MIME-compatible e-mail messages for transportation; using public-key and private-key digital signature technology for authentication, cash and secrecy.

In addition, the Jade agent platform 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 also creating and starting the execution of an agent on a remote host, provided that an agent container is already running.

On the other hand, Zeus provides a host of visual editors which an agent developer uses to specify the information required to define a Zeus agent. All the editors essentially facilitate the identification and description of a set of agents, selecting agent functionality and inputting task and domain-related data. Hence, the output of the Zeus tool-kit is a logical description of a set of agents and a set of tasks to be carried out in a domain, together with executable code for each agent and stubs for executable code for each task. Moreover, Zeus has comprehensive facilities to analysis, visualize and debug multi agent systems.

4. Comparison of implementation of a simple scenario (i.e. Service Implementation, Service Registration, and Service Discovery)

In Jade, a “yellow pages” service, provide by a specialized agent called Directory Facilitator, allows agents to publish descriptions of one or more services they provide in order that other agents can easily discover and exploit them.

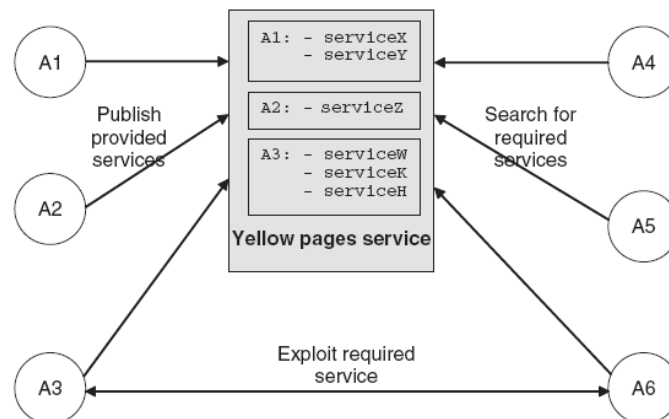


Figure 3. Yellow Pages Service

In order to publish a service an agent must create a proper description (as an instance of the `DFAgentDescription` class) and call the `register()` static method of the `DFService` class. An agent wishing to search for services must provide the DF with a template description. The result of the search is a list of all agent descriptions that match the provided template. The service implementation, in other words, the actual job, or jobs, an agent has to do is carried out within “behaviour”.

In Zeus, agent library components are also used to implement standard “utility” agents, such as an agent name server, which provides a white-pages facility for agent address look-up; a

facilitator agent, which provides a yellow-pages facility through which agents find other agents capable of performing a task; a database proxy agent (DB), whose sole function is to store and retrieve messages from proprietary databases on demand from other agents, and a visualiser agent. The utility agents fulfill a support role in their society and can be used in any application without modification. Zeus adapts the Belief-Desire-Intention architecture, and the concrete service implementation is very abstract at high level, consisting of the following activities:

- 1) Ontology Creation
- 2) Agent Creation, for each task agent this consists of:
 - Agent Definition
 - Task Description
 - Agent Organization
 - Agent Co-ordination
- 3) Utility Agent Configuration
- 4) Task Agent Configuration
- 5) Agent Implementation

The purpose of these stages is to translate the design we have derived from the role models into agent descriptions that can be automatically created by the Zeus Agent Generator tool. And it seems the service registration and discovery is automatically done by the Zeus platform according to our deification of agent organization and coordination (coarse-grained control?), we only find the following figure that shows the interaction between three task agents and the standard three utility agents in a small agent society at start up.

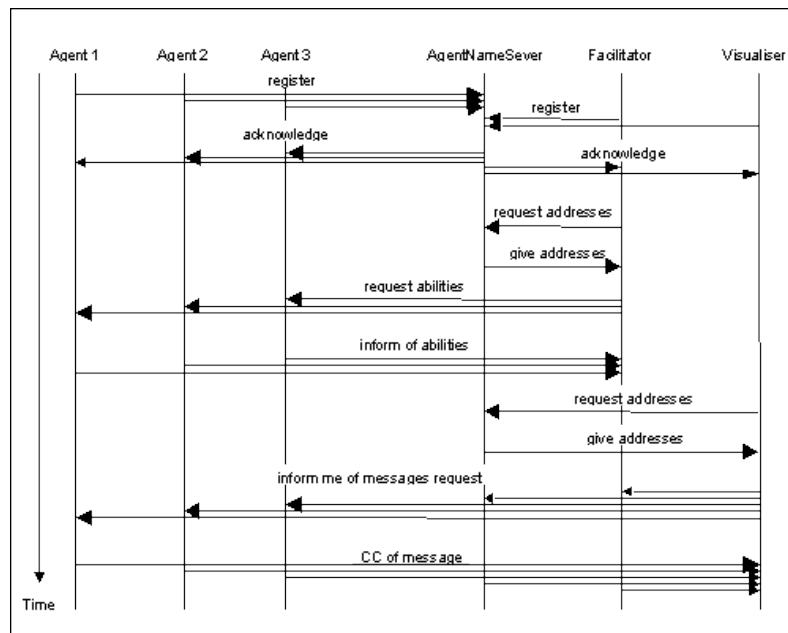


Figure 4

5. List of some notable projects

Jade: number of universities and companies included INRIA, Nice- Sophia-Antipolis, ACACIA research team, ATOS Sophia Antipolis agency within the European CoMMA project, KOD Project IST-12503, CSELT, KPN and Starlab within DICEMAN project, The Business & Technology Research Laboratory, The University of Newcastle (Australia) is evaluating JADE for eBusiness applications, etc. The LEAP is developing a new lightweight runtime environment for JADE deployed in the Internet and in the wireless environment (Motorola, ADAC, Broadcom, BT, TILAB, University of Parma, and Siemens).

Zeus: Intelligent Business System Groups (for agent based work-flow), Electronic Commerce Groups (for agent based virtual marketplaces), BT North America (for agent based network management), etc.

6. Personal opinion/judgment

Compared to Jade, Zeus aims at rapid development powered with code generation, and it abstracts the common principles and components to a much higher level than Jade. Zeus provides a methodology and rather complete set of software components and tools to design, develop and organize agent systems. Moreover it provides a runtime environment, which enables applications to be observed and other assistant tools like reports tool, statistics tool, agents and society viewer etc. The main drawback from our point of view is that Zeus API and source code are poorly documented. Though there are some documentations offered by Zeus, such as case studies, an application guide, technical manual and etc, while these do offer insights into the Zeus MAS, topics are never covered fully enough to enable solving significant programming problems. It is always the case that additional information, elicited from the source code, is always required. The same applies to all other agent platforms to some extent.

Regarding to Jade, it probably has the best (if not worse) documentation, high acceptance of users, intuitive to use and understand from programmer's aspect of view to certain extent. The Jade project itself seems still active, since the recent release is 25.06.2007, and amazingly there is even a book << Developing Multi-Agent Systems with JADE >> published in February 2007.

References

1. << ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems >>
<http://www.agent.ai/doc/upload/200302/nwan98.pdf>
2. << The ZEUS agent building tool-kit >>
<http://www.cse.unsw.edu.au/~wobcke/COMP4416/readings/Nwana-et-al.98.pdf>
3. << JADE – A FIPA-compliant agent framework>> <http://jade.cselt.it/papers/PAAM.pdf>
4. << JADE TUTORIAL - JADE PROGRAMMING FOR BEGINNERS >>
<http://jade.tilab.com/doc/JADEProgramming-Tutorial-for-beginners.pdf>
5. << AGENT PLATFORM EVALUATION AND COMPARISON >>
<http://ups.savba.sk/parcom/publications/agents/TR-2002-06.pdf>
6. << The Zeus Technical Manual >>
<http://labs.bt.com/projects/agents/zeus/techmanual/TOC.html>