

*Proceedings of the International Conference on
Information Technologies (InfoTech-2015)
17 - 18 September 2015, Bulgaria*

MULTI-AGENT FRAMEWORK FOR INTELLIGENT NETWORKS

Georgi Tsochev, MSc

*Department Computer Systems at Technical University - Sofia
e-mail: gtsochev@tu-sofia.bg,
BULGARIA*

Assoc. Prof. Roumen Trifonov, PhD

*Department Computer Systems at Technical University - Sofia
e-mail: r_trifonov@tu-sofia.bg,
BULGARIA*

Assoc. Prof. Radoslav Yoshinov, PhD

*Director of Telematics Laboratory at Bulgarian Academy of Sciences
e-mail: yoshinov@cc.bas.bg,
BULGARIA*

Abstract: To create and implement multi-agent systems a platform is needed where the agents operate from. Such environment provides agent models and tools to ease the development of multi-agent systems. This paper introduces framework for Multi-Agent System implementation. Such platform is JADE – Java Agent Development Framework. The Software makes it easy the development of multi-agent applications in compliance with the FIPA specification for generic agent technologies.

Key words: multi-agent systems, security intelligence, framework, java, software agent

1. INTRODUCTION

In recent years, highly topical question is network and information security, whether in software or hardware. In the course of the development of computer networks, and particularly at the time when electronic services such as E-Learning, E-Health, E-government and etc. are entering in our lives, this question is asked repeatedly and subjected to severe attack by researchers and scientists. A variety of means for analyzing the traffic and information passing through the network are used in order to determine whether a network is secure or not. Due to this fact, many companies and institutions are using standard means of protection such as antivirus, firewall, intrusion prevention and detection systems and etc. So in order to defend from the new threats new technologies like "intelligent networks" and "security intelligence" are introduced

During the development of "security intelligence" can be used agent-based systems. Although there is no single definition of agent [1], all definitions agree that the agent is essentially a special software component that have autonomy, which provides interoperable interfaces between any systems and / or behave like a man working for a client in achieving its agenda. Even if an agent systems is based on an agent working within an environment and if necessary interaction with users, they usually consist of multiple agents. These multi-agent systems (MAS) can model complex systems and to introduce the possibility of the agents to share common or conflicting goals. Such agents may interact with each other indirectly (by acting on the medium) or directly (through communication and negotiation). Agents may decide to cooperate for mutual benefit, or may compete to meet their own goals. During the development of such systems are often used frameworks.

A framework is a reuse object-oriented technique, which is used to develop applications framed in a given context, and which allows the design and code to be reused. Each framework is composed by a set of classes and relations among them.

A Framework for agent systems allows developers to build multi-agent systems by using both composition and inheritance of a set of restricted but still general enough agent parts. There is a considerable research effort in the development of agent design techniques. Some related works allow researchers to identify patterns in the agent development. However, agent technology is still a beginning field [2].

The purpose of this article is to present the Java oriented framework that will be used in developing information and network security system.

2. STANDARTISATION

In developing agent based systems there are certain standards that must be followed. In the intelligent agents field, standardization activities include:

- ARPA Knowledge Sharing Effort (KSE) - The KSE is a consortium to "develop conventions facilitating sharing and reuse of knowledge bases and knowledge based systems" [3]. The outputs from the KSE are

specifications for the Knowledge Querying and Manipulation Language (KQML) and the Knowledge Interchange Format (KIF), and Ontolingua, a tool for specifying ontologies. It is not clear that the KSE itself is still active, though many individual research groups continue to use and refine the core tools. No body, however, appears to be taking overall responsibility for continuing development of the original standards.

- **OMG Mobile Agent Facility (MAF)** - MAF is a proposal from a consortium including IBM, General Magic, Crystaliz, GMD-Fokus and The Open Group. It was jointly submitted as a proposal to the Object Management Group, the custodian of the CORBA standard, for an extension to the CORBA standard to include objects that can migrate and retain their computational state [4]. The MAF submission was debated by OMG, but its current status is unclear. The MAF documents are no longer reachable from the home page of General Magic, though there is still a reference on the IBM site pointing to a page at the OMG [6].
- **The Agents Society** was set up "to assist in the widespread development and emergence of intelligent agent technologies and markets" [5]. References on the web site include the "Common Agent Platform/Simple Agent Transfer Protocol" and the MAF (above), but the documents no longer seem to be being updated, and it is again unclear what their status is.
- **FIPA** - FIPA is discussed extensively below.

3. FIPA

The Foundation for Intelligent Physical Agents (FIPA) [7] was found in 1996 to produce software standards for agents and agent-based systems. The first documents of FIPA, named FIPA97 standard, state the normative rules that allow a society of agents to exist, operate and be managed. FIPA 97 describes three technology areas – Agent Communication Language, Agent Management and Agent Integration.

To complete a task or a goal the agents must communicate to each other. In network communications TCP/IP communication protocols provide such standards that facilitate the basic transfer of information. To exchange messages agent communications use Agent Communication Language (ACL). FIPA ACL is a language that specifies a standard for encoding, semantics and pragmatics of messages, but does not set out a specific mechanism for their transportation.

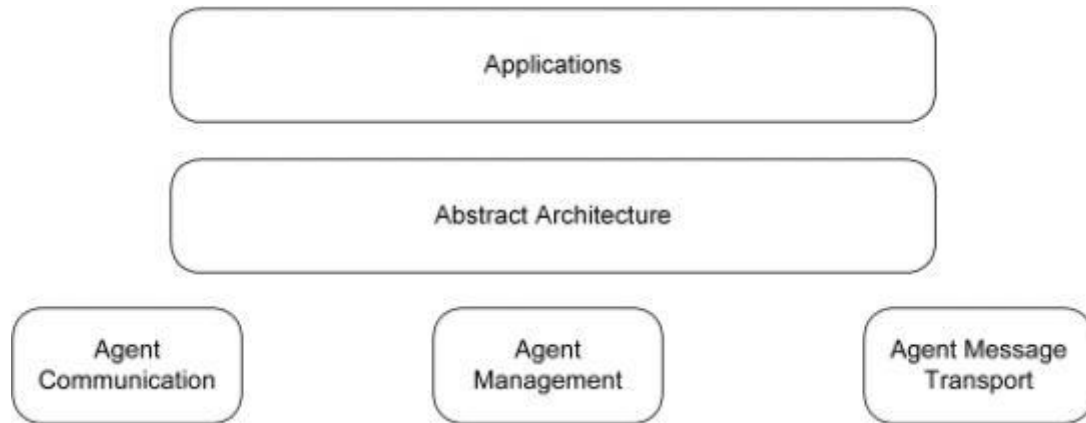


Fig. 1. FIPA 97 categories

Since different agents will run on different platforms on different networks, messages are encoded in textual form, assuming that agents has some means of transmitting like SMTP, etc. Simple example of ACL message is:

```

(request
  :sender (agent-identifier :name alice@mydomain.com)
  :receiver (agent-identifier :name bob@yourdomain.com)
  :ontology travel-assistant
  :language FIPA-SL
  :protocol fipa-request
  :content
    ""((action
      (agent-identifier :name bob@yourdomain.com)
      (book-hotel :arrival 15/10/2006
        :departure 05/07/2002 ... )
    ))""
)
  
```

In addition to communicate the agents need some supervisory control and is the function of the Agent Management System (AMS). This is the agent that exerts control over access to and use of the platform. It is responsible for maintaining a directory of resident agents and for handing their lifecycle. It establishes the logical reference model for the creation, registration, location, communication, migration and operation of agents. The AMS model consists of the following logical components [8]:

- Agent – a computational process that implements the autonomous, communication functionality of an application.
- Directory facilitator – it provides yellow pages services to other agents. It is optional component of the AP.
- Agent management system - is a mandatory component of the AP. It exerts supervisory control over access to and use of the AP.

- Message Transport Service is the default communication method between agents
- Agent Platform (AP) provides the physical infrastructure.

Agents may need to obtain a service by other entities in the system, which also can be non-agent based software systems. Agent/Software Integration defines how software resources can be described, shared and dynamically controlled in the community. The FIPA standard describes two agent roles [9]:

- Wrapper agent – encapsulates the non-agent system and lifts it up to the outside software system
- Resource broker – a set of software descriptions to interested agents.

The remaining parts of the FIPA standard deals with other aspects and they are not described here because for the developed information security systems they have not been implemented yet.

4. MULTI-AGENT FRAMEWORKS

Nowadays there are many and various applications to create agent applications. Some of the most commonly used for research activities are JADE, JADEx, AgentBuilder, JAFMAS, Grasshopper-2 , Zeus, JACK.

Grasshopper is an open 100% Java-based mobile intelligent agent platform, which is compliant to both available international agent standards, namely the OMG MASIF and FIPA specifications. Grasshopper includes two optional open source extensions providing the OMG MASIF and FIPA standard interfaces for agent/platform interoperability.

JACK - Jack Intelligent Agent JACK is a framework in Java for multi-agent system development. JACK Intelligent Agents was built by Agent Oriented Software Pty. Ltd. (AOS) It is mature, cross-platform environment for building, running and integrating commercial-grade multi-agent systems. It is built on a sound logical foundation: BDI (Beliefs/Desires/Intentions).

ZEUS - 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. Zeus provides support for generic agent functionality and has sophisticated support for the planning and scheduling of an agent's actions. Agents are composed of file layers – API, definition, organizational, coordination and communication layer.

AgentBuilder is an integrated tool suite for constructing intelligent software which consists two components. The first one is the Toolkit which includes tools for managing the agent-based software development process, analysing the domain of agent operations, developing networks of communicating agent software. The second component is the Run-Time Systems which is an agent engine that provides an environment for execution of agent software. [10]

FIPA-OS – is a component-based toolkit for rapid development of FIPA compliant agents. It is an open agent platform originating from Nortel Networks. The platform supports communication between multiple agents using communication language which conforms to the FIPA standard. It supports openness [11].

5. JADE PLATFORM

JADE (Java Agent Development Environment) is a software framework implemented in Java language. JADE is Open Source project and is distributed by Telecom Italia, the copyright holder, under the terms and conditions of the LGPL (Lesser General Public License Version 2). It can be downloaded from JADE Home Page [12]. It makes it easy the development of agent applications in compliance with the FIPA specifications for intelligent multi-agent systems. While it simplifies the development also ensures the standard compliance through a comprehensive set of system services and agents.

Jade comes with a tools that can simplify the development of an application. It includes runtime environment, library, and graphical tools. The programmers may use a library of classes to develop their agents. For administrating and monitoring the activity of running agents there is a suite of graphic tools, which makes it easier. To work all this there is a runtime environment where JADE agents can live.

5.1 JADE Architecture

Figure 2 represents the JADE architecture elements [13]. Each running instance of JADE running environment is called a Container. It is called that way because it can contain zero or several agents. Every agent is identified by a unique name and they know each other's name. Also they can communicate transparently regardless of their actual location. There is a special container called Main Container in the platform. The normal containers must be told where to find their main container. Besides this difference it also must be the first container to start in the platform. It includes two special agents Agent Management System (AMS) and Directory Facilitator (DF). AMS provides a naming service and represent the authority in the platform. DF provides a yellow pages service by means of which an agent can find other agents providing the service he requires in order to achieve his goal.

The set of active containers is called a Platform. It provides the agents with the basic services such as message delivery.

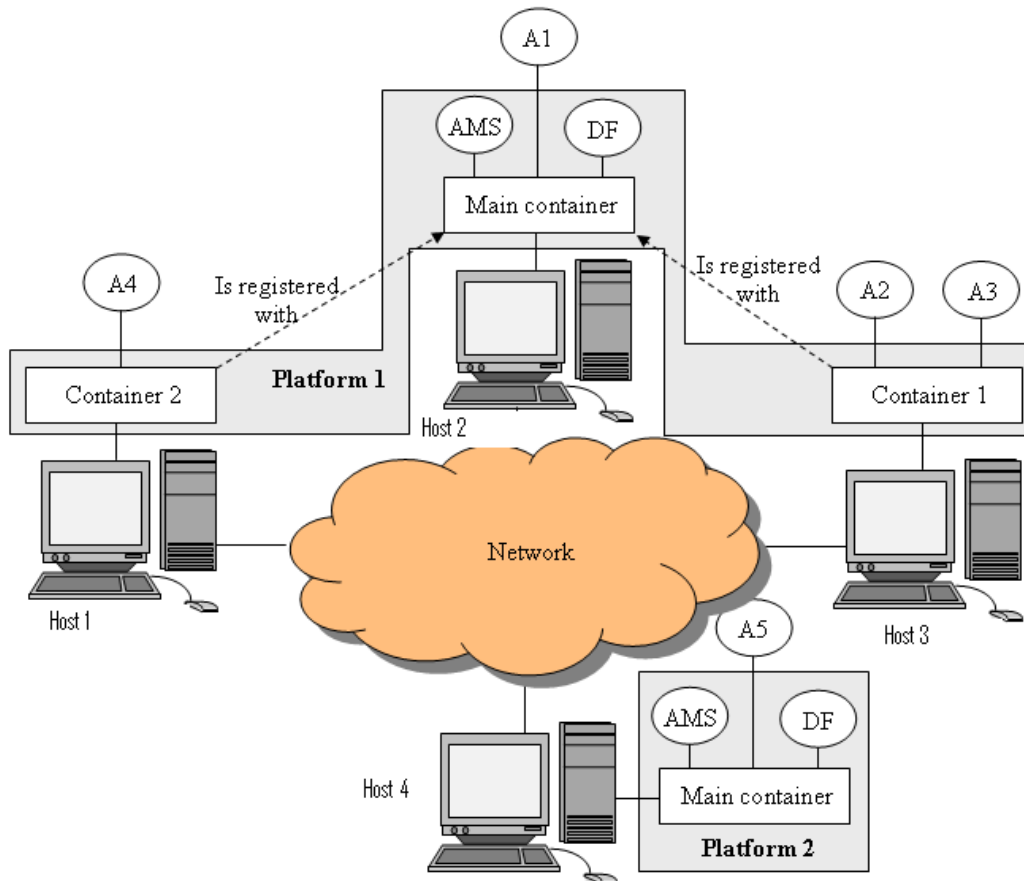


Fig. 2. JADE Architecture

As mentioned agents communicate flawlessly and transparently regardless of whether their location is. Communication is based on an asynchronous message passing. The messages exchanged by JADE agents have a specified format defined by ACL language in FIPA international standard for agent interoperability. This format contains some fields like:

- Sender of the message
- Receiver or a list of receivers
- The actual information of the message – content
- Language of the content
- What is the intention of sending the message, what the sender intends to achieve
- Other control fields

5.2 JADE agent lifecycle

According to Agent Platform Life Cycle defined in FIPA specification there are several states shown in Figure 3.

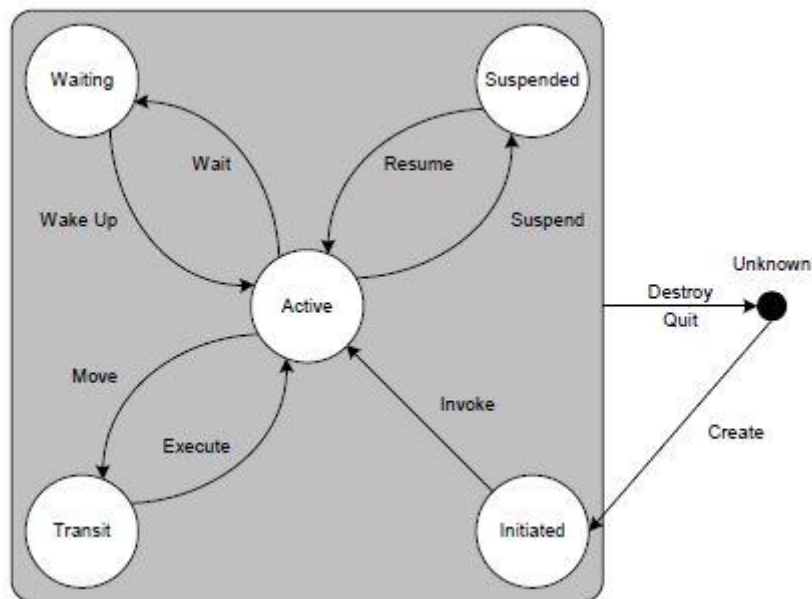


Fig. 3. Agent life-cycle as defined by FIPA

The life cycle of a JADE is as follows:

- INITIATED – the agent is only created, but not registered with the AMS
- ACTIVE – the agent is registered with AMS and has a name. Also it can communicate with other agents
- SUSPENDED – the agent is stopped and his internal thread is suspended
- WAITING – the agent is blocked, waiting for something
- DELETED – the agent is dead and the internal thread has terminated its execution. The agent is not any more in the AMS
- TRANSIT – the agent is moving to a new location [14]

6. CONCLUSIONS

With the review of the various agent frameworks like Zeus, JACK, Grasshopper-2 and others, JADE framework for Multi-Agent System implementation is more simple and easy to work. JADE acts as a middleware that facilitates the development of multi-agent system under the Foundation for Intelligent Physical Agents – FIPA. JADE application programming interface APIs make it easy for the developers. Also with the integrated functionalities the application development time and cost is reduced. JADE tries to support large Multi Agent System as possible. JADE agents are very scalable with the number of simultaneous conversations a single agent can participate in.

REFERENCES:

- [1] O'Hare, G. and Jennings, N. (editors). Foundations of Distributed Artificial Intelligence. Willey-Interscience Publication (1996).
- [2] Avancini, H., & Amandi, A. (2000). A java framework for multi-agent systems. *SADIO Electronic Journal of Informatics and Operations Research*, 3(1), 1-12.
- [3] Neches R, "The Knowledge Sharing Effort", <http://www.wksl.stanford.edu/knowledgesharing/papers/kse-overview.html>, 1994
- [4] "Joint Submission: Mobile Agent Facility Specification", <http://www.genmagic.com/agents/MAF/mafraft7.pdf>, 1997
- [5] Agent Society Home Page, <http://www.agent.org/>, 1997
- [6] OMG TC Work in progress CF RFP 3
http://www.omg.org/library/schedule/CF_RFP3.htm, 1997
- [7] Foundation for Intelligent Physical Agents. Specifications. 1999.
Available at <http://www.fipa.org>.
- [8] FIPA Agent Management Specification. 2000. Available at <http://www.fipa.org>
- [9] <http://www.fipa.org/specs/fipa00079/XC00079B.html>
- [10] <http://www.agentbuilder.com>
- [11] Poslad St, Buckle, Phil. & Hadingham, R. The FIPA-OS agent platform: Open Source for Open Standards
- [12] <http://jade.tilab.com/>
- [13] Caire, G. (2009). *Jade programming for beginners*. Retrieved from <http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>
- [14] Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2010). *Jade programmer's guide*. Retrieved from <http://jade.tilab.com/doc/programmersguide.pdf>