International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321 - 8169
Volume: 3 Issue: 2 895 - 898

# Multi-Agents Implementation Frameworks - An Overview

Ms. Meena Agrawal, Dr. Arvind Mittal
Energy Centre, Maulana Azad National Institute of Technology (MANIT), Bhopal, (MP) India.
*Email: oshomeena@gmail.com, Email: am1970nit@gmail.com*

***Abstract:*** Large scale deployment of Micro Grids besides the advanced metering, demand response, reliable communications infrastructure set up has been incorporated into the technological road map of the future smart power grid. To congregate the operation and control needs of distributed energy resources in Micro-Grids the Multi-Agent System (MAS) seem to have splendid features. MAS is an emerging sub-field of Distributed Artificial Intelligence that has the potential to manage the changing face of electric power grid by inculcating intelligent agents into Micro-Grids. To create agents and implement MAS a framework, a platform is obligatory where in the agents reside and operate from. There is a wide range of Multi-agent platforms available on the web like Aglet, Grasshopper, DESIRE, Jadex, ZEUS, JADE etc. Each agent platform has to be evaluated according to the some criteria that have been mentioned in this endeavor. A brief relative appraisal of an assortment of agent platforms has been provided. According to various noteworthy researches the most used platform in micro-grid applications is JADE. This paper presents an architectural and functional overview of the agent building toolkit JADE framework for Multi-Agent System implementation.

***Keywords:*** *Agents Implementation, Jade Framework, Micro Grid, Multi-Agent system.*

*****

## I. INTRODUCTION

In line with the global rhythm the Government of India (GoI) too is determined to expedite the impending Smart Grid Technologies. For smartly addressing the problems related to sustainable energy & environment, the Micro Grid deployment is an important that concurrently extracts the benefits of technical advancements. In 2013, the GoI issued Smart Grid Vision and Roadmap for India which is a 15 year map for transformation of the Indian power system to Smart Grids. With the recent activities around 14 Smart Grid Pilot projects in different states, India has emerged as the hot destination for Smart Grids and hence the Micro Grids [1].

The smart grid is to be materialized as a well-planned plug-and-play integration of several smart Micro-Grids that will be interconnected through dedicated ICT highways for command, data, and power exchange. Micro-grids can efficiently integrate distributed and renewable generation resources into the grid and can island & interconnect seamlessly from the grid during peak hours with properly managing the demand response [2]. New control and management paradigms and technologies that are different from the traditional methodologies are necessary for the operation of Micro Grids which are of distributed renewable resources type primarily.

Agent-oriented software engineering paradigm represents an interesting means of analyzing, designing and building complex software systems, quite suitable to the requirements of making Micro-Grids smart. Multi-agent system (MAS) is a distributed Computational Intelligence (CI) technique that has been applied to solve several power system problems such as market operation, condition monitoring, fault diagnosis, power system restoration and protection. The agent technology has a great potential to solve problems in the control and management of modern power systems that implement smart grid techniques. [3]

A Multi-agent system consists of several agents that interact with one other. An agent may be a physical or virtual entity that can act, perceive its environment in a partial way, communicate with others, and has skills to achieve its goals and tendencies. These interactions are handled by messages that are sent between the agents. Each of the agents can have different goals and behaviors, which together combines to a dynamic system [4]. An agent receives information about a state of its environment, take actions which may alter that state and expresses preferences among the various possible states. The agents interact with one another in order to realize their goals.

## II. AGENT CREATION & IMPLEMENTATION

To create agents and implement MAS a framework, a platform is required where in the agents reside and operate from. By using a platform, the implementation of the system becomes much smaller, because the communication and several other aspects have already been taken care of. This makes it possible to focus solely on the agent implementation itself. There is a wide range of Multi-agent platforms available on the web like Grasshopper, DESIRE, Jadex, TuCSoN, JACK, JAFMAS, ZEUS and JADE [5]. From this wide range of platforms, suitable one can be selected and used. This platform must match some criteria in order to be able to use the agents' possibilities to their full extent and prove to be the right platform for this job. Each of these platforms differ in their features and their flaws and a lot of them are no longer being updated or supported.

## III. AGENT PLATFORM FEATURES

There exist a huge number of approaches, toolkits, and platforms of different quality and maturity, over 100 software products [13]. Therefore a set of criteria requirements for the platforms necessary is to be made. Each agent platform is evaluated according to the following criteria [14]:

- Standard compatibilities: common standards for agent technology are FIPA, OMG, MASIF,
- Communication: support for inter-platform messaging
- Agent mobility: strong (ability of system to migrate code and execution state of executing unit), weak (migration of code only) - clean, efficient method of migrating.
- Security policy: secure intra-platform and homogeneous inter-platform communication.

- Availability Usability and documentations: user and developer level of acceptance
- Development issues: practical applications/development projects.

## IV. AGENT PLATFORM SPECIFICATIONS

The list of few agent platforms evaluated in this paper comes from the FIPA recommendation for publicly available agent platform implementations.

Within the scope of Micro grid applications the following agent platforms are recommended [15-17]:

Grasshopper:
• very good documentation, very good GUI and TUI, high acceptance of users, used in many development projects
• standards MASIF, FIPA, very good security features, plug-in for web interface, logical grouping of functionality of agents
• weak agent mobility with possibility to "simulate" strong mobility, various communication protocols.

JACK Intelligent Agent:
• Supported platforms: Java 2, work with CORBA.
• Implemented standards: FIPA.
•Communication: JACK needs DCI network for communication; similar to TCP/IP it needs one process running as a name-server.
• Mobility: no
• Security policy: internal security provided by JDK.

ZEUS:
• free open source, good documentation, very good user friendly GUI, acceptance of users, used in many development projects
• standard FIPA, very good security features,
• weak agent mobility, various communication protocols,
Communication :ACL and KQML
Supported platforms: Java 2
Security policy : Some security capabilities

JADE:
• free open source, good documentation, very good GUI, acceptance of users, used in many development projects
• standard FIPA, very good security features,
• weak agent mobility, various communication protocols,
• Supported platforms: Java 2 work with CORBA
•Communication: ACL, support for inter-platform messaging with plug-in MTP
• Mobility: weak mobility
• Security policy: JADE Object Manager provides connection authentication, user validation and RPC message encryption.

## V. AGENT PLATFORM COMPARISONS

In agent building the following characteristics are to be considered [18]:

- time required for analysis and design of multi-agent system;
- time required for code generation;
- time required for integration of agent with external code;
- time required for testing and debugging the system;
- number of lines in the code;
- number of new java classes and
- number of reused classes to build news agents in the system

The most comprehensive comparison of agent building toolkits, JADE, JATLite, Skaleton and Zeus, have been taken into consideration due to their Java compatibility and graphical interface. These can be compared in a web-based news retrieval environment based on their software building and runtime characteristics. Runtime characteristics include parameters of response time and number of messages exchanged among agents. Results exhibit similar performance of Zeus and JADE in software building characteristics while JADE outperforms Zeus in runtime characteristics. [19]

The performance comparison of Zeus, JADE and Skeleton in terms of number of documents requested from a web server versus response time while varying the number of web agents. The paper concludes that Zeus performs better the other two when system is less complex. The author provided an evaluation of various tools in terms of availability analysis, environment analysis, development analysis, characteristics analysis and Inter and Intra-process message delivery results. [20]

According to various noteworthy researches the most used platform today for micro grid type applications is JADE which is being discussed in this paper. The JADE platform supports Multi-agent system development with Java language.

## VI. JADE PLATFORM

Java Agent DEvelopment Framework, JADE is a software framework for development of agent-based applications, implemented in JAVA computer programming language. JADE is an open source free software distributed by Telecom Italia Lab, TILAB, the copyright holder, under the terms and conditions of the LGPL (Lesser General Public License Version 2). TILAB is the R&D branch of an Italian telecommunications company and is responsible for promoting technological innovation by scouting new technologies, carrying out and assessing feasibility studies, and developing prototypes and emulators of new services and products. An Open Source Community is in active operation since February 2000. A JADE Board has been created in 2003 that supervises the management of the JADE Project. The latest version of JADE is JADE 4.3.2 released on 28/03/2014. [6]

JADE is a middleware to facilitate development of multi-agent system under the Foundation for Intelligent Physical Agents, FIPA. It is a non-profit association registered in Geneva, Switzerland with a purpose of promotion of emerging agent-based applications, services and equipment. JADE implements basic FIPA specifications that provide the normative framework within which FIPA agents can exist, operate, and communicate. The goal is to simplify the development while ensuring standard compliance through a comprehensive set of system services and agents. JADE system is in compliance with the FIPA specifications and provides a standard implementation of the communication language FIPA-ACL to facilitate the communication between agents. [7]

Where the resources and the control logics are distributed in the environment, use of JADE simplifies the development of applications that require negotiation and coordination among multi agents. Easy to use software libraries to implement peer-to-peer communication and interaction protocols are provided by JADE to developers. By making different layers transparent for the developer to concentrate on the logic of the application, JADE simplifies the management of agent communication and message transport. JADE application programming interface APIs and ready-to-use functionalities allow to strongly reducing the application development time and costs. Special features ease development of agents for very different application sectors including communication networks, auctions, tourism, manufacturing, rescue management, Power system networks, supply chain management, fleet management, e-market, etc. [8]

JADE software Framework simplifies the implementation of multi-agent systems through a set of graphical tools that support the debugging and deployment phases. A JADE-based system can be distributed across machines which not even need to share the same OS and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another, as and when required. JADE is completely implemented in Java language and the minimal system requirement is the version 5 of JAVA. Besides the agent abstraction, JADE provides a simple yet powerful task execution and composition model, peer to peer agent communication based on the asynchronous message passing paradigm, a yellow pages service supporting publish subscribe discovery mechanism and many other advanced features that facilitates the development of a distributed system.

## VII. JADE ARCHITECTURE:

The JADE architecture includes:

- A runtime environment where JADE agents must be active on a given host before one or more agents can be executed on that host.
- A library of classes that programmers can use (directly or by specializing them) to develop their agents.
- A powerful suite of graphical tools that allows administrating and monitoring the activity of running agents. [9]

An application based on JADE is made of a set of components called Agents each one having a unique name. Agents execute tasks and interact by exchanging messages The main JADE architectural elements are Agent, Container, Platform, Main Container, Agent Management System (AMS) and Directory Facilitator (DF). Figure 1 represents these main JADE architectural elements. Agents live on top of a Platform that provides them with basic services such as message delivery. A platform is composed of one or more Containers. Containers can be executed on different hosts thus achieving a distributed platform. If another main container is started, as in host Host 4 in Figure 1, this constitutes a new platform.

Each platform must have a parent container called Main Container that has two special agents called AMS and DF. The DF provides a directory which announces which agents are available on the platform. The AMS controls the platform. Is the only one who can create and destroy other agents, destroy containers and starts-stops the platform.

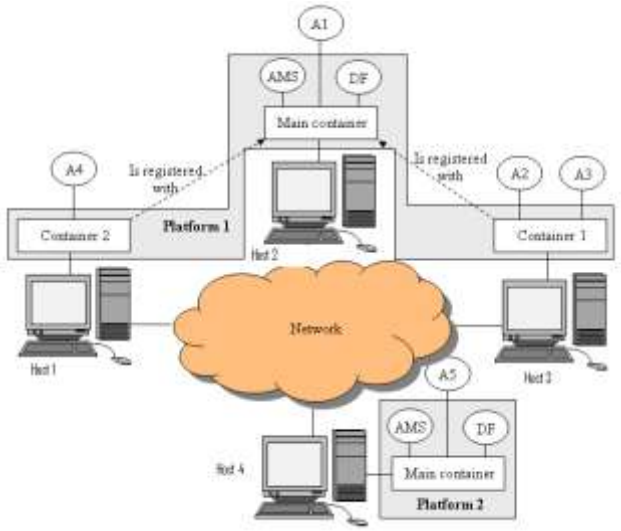


Figure 1. The JADE Architecture [10]

The parent container is the first container to start in the platform and all other containers register to it at bootstrap time. The AMS represents the authority in the platform. The DF provides the Yellow Pages service where agents can publish the services they provide and find other agents providing the services they need.

Further JADE has various powerful debugging tools, mobility of code and content agents, the possibility of parallel execution of the behavior of agents, as well as support for the definition of languages and ontologies. It also includes two Java classes libraries required to develop application agents and the run-time environment that provide the basic services and that must be active on the device before agents can be executed.

## VIII. JADE AGENTS COMMUNICATION

From the functional point of view, JADE provides the basic services necessary to distributed peer-to peer applications in the fixed and mobile environment. It allows each agent to dynamically discover other agents and to communicate with them according to the peer-to-peer paradigm.

Each agent is identified by a unique name and provides a set of services, in application point of view. It can register and modify its services and/or search for agents providing given services, it can control its life cycle and, in particular, communicate with all other peers. Agents communicate by exchanging asynchronous messages, a communication model almost universally accepted [3].

Agents can communicate flawlessly and transparently regardless of whether they live in the same container (e.g. A2 and A3), in different containers (in the same or in different hosts), belonging to the same platform (e.g. A1 and A2) or in different platforms (e.g. A1 and A5).

Communication is based on an asynchronous message passing paradigm in format defined by the ACL language defined by FIPA. An ACL Message contains a number of fields including the following: the sender, the receiver(s), the communicative act that represents the intention of the

sender of the message, and the content that is the actual information conveyed by the message viz. for an INFORM message the fact the receiver should become aware of, whereas in case of a REQUEST message - the action that the receiver is expected to perform. For instance when an agent sends an INFORM message it wishes the receiver(s) to become aware about a fact (e.g. INFORM "today it's raining"). When an agent sends a REQUEST message it wishes the receiver(s) to perform an action. [10]

The cycle of life of a JADE agent follows the cycle proposed by FIPA. Thus the agents go through different states defined as:

- Initiated: The agent has been created but has not registered yet the AMS.
- Active: The agent has been registered and has a name. In this state it can communicate with other agents.
- Suspended: The agent is stopped because its thread is suspended.
- Waiting: The agent is blocked waiting for an event.
- Deleted: The agent has finished and his thread ended his execute and there is not any more in the AMS.
- Transit: The agent is moving to a new location.[12]

## CONCLUSIONS

An overview of various Agent development toolkits has been done. A functional and model overview of the agent building toolkit JADE framework for Multi-Agent System implementation has been presented in details. 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 and ready-to-use functionalities allow to strongly reducing the application development time and costs.

Various agent frameworks features have been discussed. Comparing many agent building toolkits: Zeus, JADE, JACK and Grasshopper are considered as the best performers of the lot. Few review results exhibit similar performance of Zeus and JADE in software building characteristics while JADE outperforms Zeus in runtime characteristics. Agent platforms are mainly chosen on the basis of which criteria that has been highlighted.

This appraisal for all above mentioned works concludes that for selection of agent building toolkit is to be based largely upon the Standard compatibilities, Communication, Agent mobility and the Development issues besides on the user's choice and the requirements of the particular research area but is to be based largely upon the Standard compatibilities, Communication, Agent mobility and the Development issues.

## REFERENCES

[1] SMART GRID Bulletin | Volume 1, Issue 3 | March 2014; www.indiasmartgrid.org/en/Pages/isgfprofdatabase.aspx
[2] H. Farhangi, The path to smart grid, IEEE Power and Energy Magazine, Jan/Feb 2010.
[3] Thillainathan Logenthiran, Multi-Agent System for Control and Management of Distributed Power Systems; PhD Thesis, Department of Electrical And Computer Engineering National University Of Singapore; March, 2012.
[4] Marcel Koster; Faculty of Mathematics and Natural sciences; Rijksuniversiteit Groningen; Reliable Multi-agent System for a large scale distributed energy trading network, Master Degree Thesis;2010/2011
[5] R. H. Bordini, L. Braubach, et al. A Survey of Programming Languages and Platforms for Multi-Agent Systems, In Informatica, Vol. 30, Pages 33-44, 2006.
[6] JADE, "Jade - Java Agent DEvelopment Framework." [Online] Available:http://jade.cselt.it/.
[7] IEEE FIPA, Foundation for Intelligent Physical Agents. [Online] : http://www.fipa.org/. [Accessed: 14-Jun-2014]
[8] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa ; JADE;A White Paper http://exp.telecomitalialab.com
[9] JADE TUTORIAL; JADE FOR BEGINNERS; 30 June 2009. JADE 3.7 Authors: Giovanni Caire (TILAB, formerly CSELT)
[10] JADE Architechture /JADEtutorials/Tutorial20%JADE%20Architecture.
[11] Jade programming for beginners, 2009. [Online] Available: http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial.
[12] wikipedia.org/wiki/Java_Agent_Development_Framework
[13] Publicly Available Implementations of FIPA Specifications http://www.fipa.org/resources/live systems.html
[14] Agent Toolkit List http://www.csm.uwe.ac.uk/~rsmith/ECOMAS/agent_toolkit_list
[15] Grasshopper – the agent platform http://www.grasshopper.de/
[16] JACK® Intelligent Agents- a framework overview Pedro Valente AOP course@TEK Faculty 12/12/2011
[17] Zeus agent development toolkit [Online]. Available: http://labs.bt.com/projects/agents/zeus.
[18] Shakshuki, and Y. Jun, Multi-agent Development Toolkits: An Evaluation, Springer Berlin/Heidelberg, 2004.
[19] Camacho, D. and Aler, R., ' Software and Performance measures for evaluating multi-agent frameworks', in Applied Artificial Intelligence, Volume 19, Number 6, Madrid, Spain : Taylor and Francis Ltd, pp. 645-657, 2005.
[20] A.R. Silva, A. Romao, D. Deugo and M. Silva, "Towards a reference model for surveying mobile agent systems", Autonomous Agents and Multi-Agent Systems, vol. 4, no. 3, pp. 187231, September 2001.