ISSN No. 0976-5697

Volume 4, No. 8, May-June 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Architecture Model for Communication between Multi Agent Systems with Ontology

Vishal Jain
Research Scholar, Computer Science and Engineering Department, Lingaya's University, Faridabad
vishaljain83@ymail.com

Dr. Mayank Singh
Associate Professor,
Krishna Engineering College, Ghaziabad
mayanksingh2005@gmail.com

Abstract: Internet is considered as repository of collection of web documents. The documents may be unstructured, structured and semi-structured. It is known that mostly of these documents are unstructured. Although there is larger number of documents residing on web storage house, still the users are unable to find relevant information about given domain. The reason is the uncertainty of documents that creates users in dilemma by providing hundreds of results and keywords in response to given query. This paper illustrates the use of JADE (Java Agent Development Framework) in improving the process of information retrieval. JADE creates well defined applications on given topic by providing relevant and concise information in form of GUI environment. JADE provides easy process of development by ensuring set of services and agents. The current paper also focuses on JADE features including its intended architecture model, its tools that are useful in building development environment. It also covers Interaction Protocols and provides information about communication between agents. There are various versions of JADE that does not match with latest content languages and ontologies that are used in JADE. It also supports ontologies with the help of Content Reference Model.

Keywords: Information Retrieval, Multi Agent Systems (MAS), Ontology, JADE

I. INTRODUCTION

Since Internet is based on traditional client- server approach. It is controlled by various central servers consisting of various topics of information. The information is transferred to systems located at centralized location. Then what is way to transfer or retrieve information to/from complex systems. It leads to concept of Multi Agent Systems (MAS).

Systems where individual self authorized agents derive new facts with the help of other agents are called Multi Agent Systems [1]. In these systems, individual agents create their different models and prototypes instead of following standard ontology. An agent is defined as entity with or without body. Agents use different kinds of knowledge sources and resolve differences among themselves to provide relevant information [2]. We can also define MAS in terms of characteristics of agents as follows:

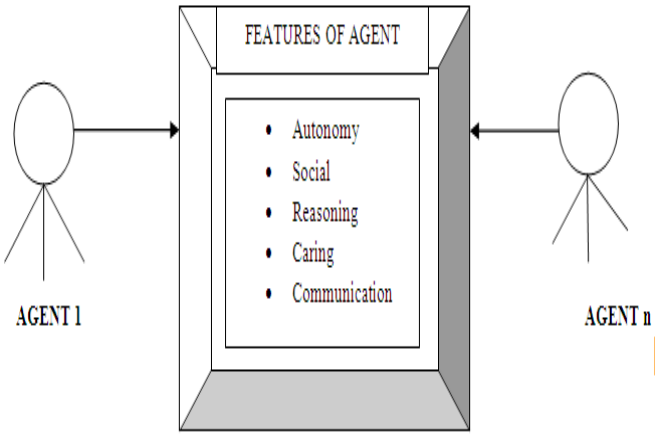

Figure 1: Characteristics of Agents

In order to realize with Multi Agent Systems (MAS) and interact with their agents, a framework is developed that is called JADE (Java Agent Development Environment) [3, 4]. It is considered as middleware that implies agent platform and development framework. Various researchers and developers have contributed to the development of JADE. These developers are working on different platforms for providing innovative solutions in creation of agents in complex systems. In Section 2, we have described a brief survey on Information Retrieval (IR) technology. It describes the process of Information Retrieval (IR). Section 3 gives information about JADE platform. In Section 4, we have defined our ontology by giving example of 'The Bakery Shop' for supporting JADE content language and features. Section 5 concludes about the results that are derived from paper. The last Section 7 displays list of References used in this paper.

II. INFORMATION RETRIEVAL (IR)

It involves identifying and extracting relevant pages containing that specific information according to predefined guidelines [5,6]. There are many IR techniques for extracting keywords like NLP based extraction techniques are used to search for simple keywords. IR mainly focuses on retrieval of natural language text. It addresses retrieval of documents from an organized well defined huge collection of documents on web [7]. When we type any query or keywords in search engine, we get large number of results matching those keywords. The results may or may not be relevant. The process of retrieving information is shown below:

Background knowledge stored in form of ontology can be used at any step. As we have ranked list of documents, they are indexed to form document in represented way. These documents produce ranked results which are given to admin. Admin solves user query which leads to transformation of user query.

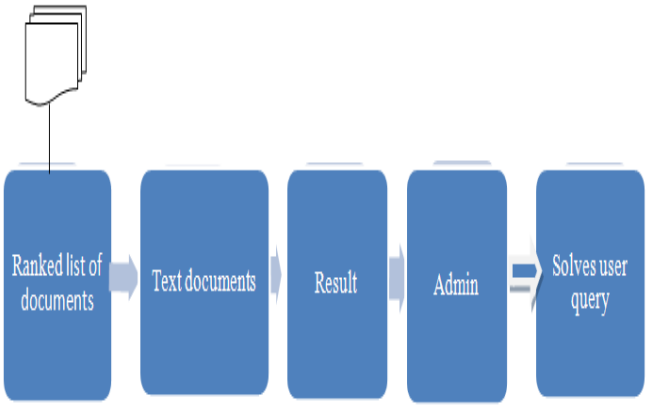

Figure 2: Retrieval of Information [2]

III. MUCH WE KNOW ABOUT JADE?

There are two conditions that must be fulfilled to work on JADE:

a. FIPA – compliant Agent Platform
b. Packages to develop agents.

FIPA stands for Foundation For Intelligent Physical Agents [3]. The first output document of FIPA is called FIPA 97 Specifications. It specifies rules that allow agents to interact between systems by defining transport level protocols. JADE is open framework RAD tool for development of J2EE. Uses of JADE are listed below:

a. It allows people to work in proper way by following rules and regulations.
b. It enables communication and negotiation with other agents.
c. JADE systems are efficiently designed an deployed in fields like Internet Services, Mobile Environment, Peer Management etc.
d. It offers Portability i.e. it can run on different platforms.

A. Feature of JADE [12, 13]:

a. Distributed Agent Platform*:* - The agent platform has several hosts and agents. Agents are type of java threads and lives in Containers. A Container is running instance of JADE running environment containing several hosts. There are two types of containers – Main Container and Other Containers. Main Container hosts AMS and DF.

a) AMS: - It stands for Agent Management System. It gives naming of agents in order to distinguish them other agents. It offers services like White Pages Services, Life Cycle and assigns AID (Agent Identifier) to each agent.

b) DF: - It stands for Directory Facilitator. It gives yellow pages service which enables an agent to find other agents providing services to attain their goal. Agents can interact with DF by exchanging ACL (Agent Communication Language) messages using Publish and Subscribe schema.

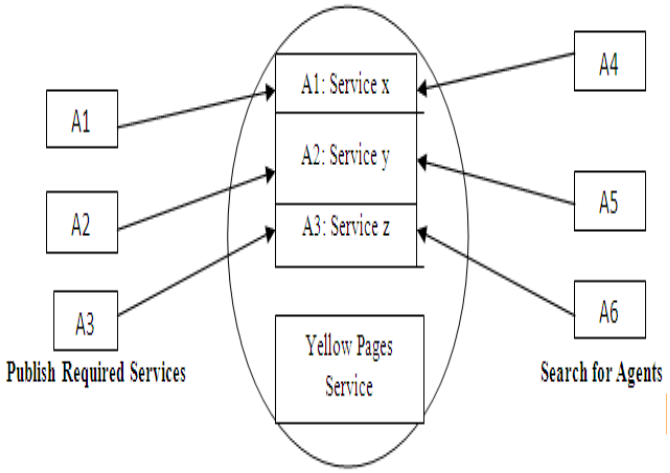

Figure 3: Role of DF

FIPA compliant Agent Platform: - JADE fully complies with FIPA Specifications. Its architecture is given below:

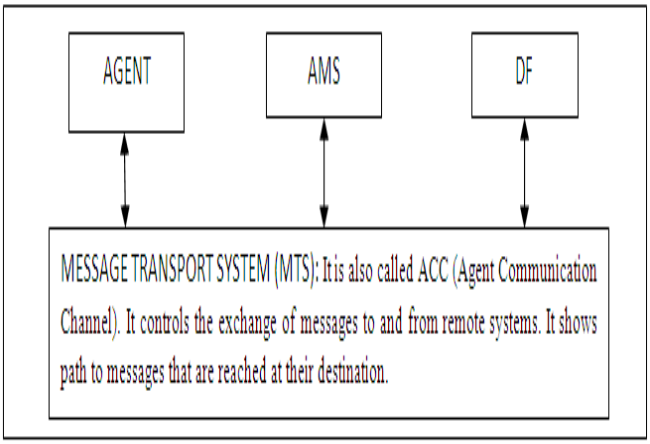

Figure 4: FIPA Agent Platform

(a). Intra-platform Agent Mobility: - It includes transport of state and code of agents.
(b). It executes multiple, concurrent agent activities using Behaviors.
(c). FIPA Interaction Protocols are ready to use. FIPA gives agent GUID (Globally Unique Identifier) at time of startup platform.
(d). It supports application content and ontologies.
(e). In Process Interface: - It allows external applications to launch agents.
(f). Debugging and Graphical Tools: - They help to develop multi agent applications based on JADE.
(g). There is modified version of JADE that can also run on mobile devices called LEAP add-on. It has same API's like JADE so that agent can run on JADE as well as LEAP.

B. Agent Tasks:

Various steps are required in execution of agents. They are listed below:

a. Create an object by extending AGENT class.
b. It is given an identifier that is called AID.

c. It is registered with AMS and remains in ACTIVE state.
d. Then setup () method is executed. It is method where agent activity is started. This method is used to modify data registered with AMS. It also adds tasks to queue of messages.

Various steps are required to stop the execution of agents:

a. doDelete ():- It is method used to stop the execution.
b. takedown (): - It is executed when agent is about to go to Deleted Stage. It is used for cleanup operations.

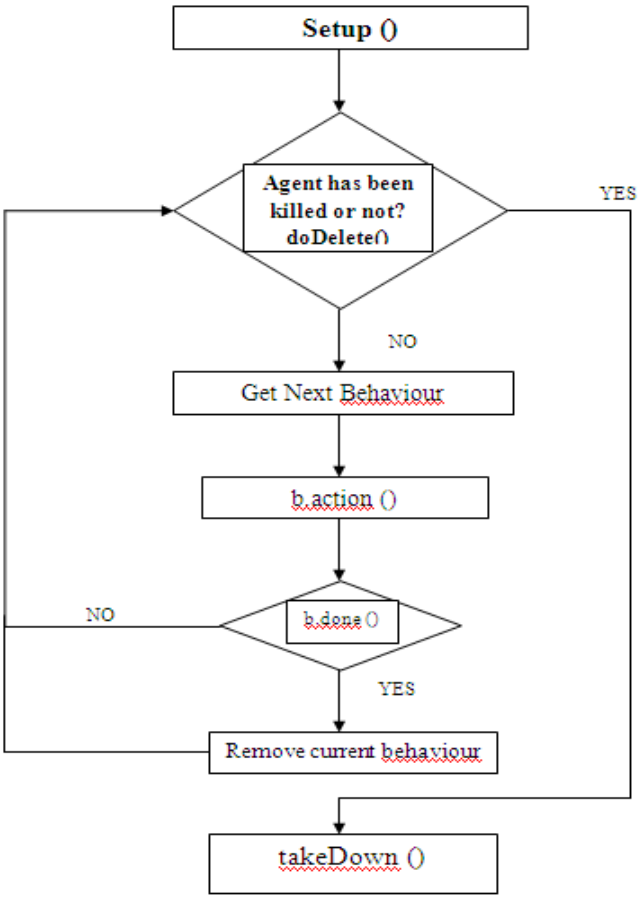

Figure 5: Agent Execution Model

In above model, we have used some methods associated with Behaviors. Behavior is defined as abstract base class for modeling agent tasks. They represent tasks carried out by agents. Methods associated with Behaviors:

(a). Action (): - It tells what behavior actually does.
(b). Done (): - It tells whether behavior is finished or not. If it returns true, then remove current behavior from pool of behaviors and go to next behavior.
(c). Restart (): - It is used to resume execution of blocked behavior.
(d). onStart (): - It is called only once before first execution of action ( ) method.
(e). onEnd ():- It is called once after done () method returns true i.e. when behavior has been removed from pool of agent behaviors.

C. Agent Communication and Interaction Protocols:

The communication model in JADE is based on Asynchronous message passing. The process is as follows:

a) JADE main container maintains RMI registry supported by other agent containers.
b) The process of taking reference of addresses of other containers is called Address Caching.
c) ACL Message is class that exchanges messages between agents.
d) For Sending: - Create new ACL Message object, fill its attributes and call method Agent. send ().
ACLMessage msg = new ACLMessage (ACLMessage.INFORM);
msg.addReceiver (new AID ("Vishal", AID.ISLOCALNAME));
e) For Receiving: - Call receive () method.
ACLMessage msg = receive ();
If (msg! = null) {
"Hello"
}

a. Interaction Protocols:- The standard types of messages defined in ACLMessage class (INFORM, REQUEST, and PROPOSE) allows exchange of messages by agents during conversations is called Interaction Protocols. There jade.proto package that contains methods for initiator and responder in protocols like FIPA-Subscribe (Subscription Initiator/Responder), FIPA- request (Achieve RE Initiator/ Responder) etc.

D. Versions of JADE:

Since JADE follows approach "pay as you go". It means improved versions of JADE provide number of advanced features as compared to older version [4]. The first version of JADE was 1.3. Some of versions of JADE are illustrated in table1:

IV. ONTOLOGY IN JADE

According to FIPA specifications, DF and AMS communicate by using FIPA – SLO content language, FIPA-Agent Management ontology and FIPA – Request Interaction protocol [14, 15, 16].

a. SL-O Content Language is implemented by class *jade.content.lang.Sl.SLCodec*. Any agent can use this language using *getContentManager (). Register Language (new SLCodec ())*;
b. Concept of ontology is implemented by classes in *jade.domain.FIPA Agent Management* package. FIPA Agent Management Ontology defines vocabulary with all constant symbols of ontology. Any agent can use this ontology automatically by *get Content Manager() .Register Ontology (FIPA Management Ontology. Get Instance ())*;
c. FIPA – Request Interaction Protocol is implemented as ready to use in *jade.proto.package*.

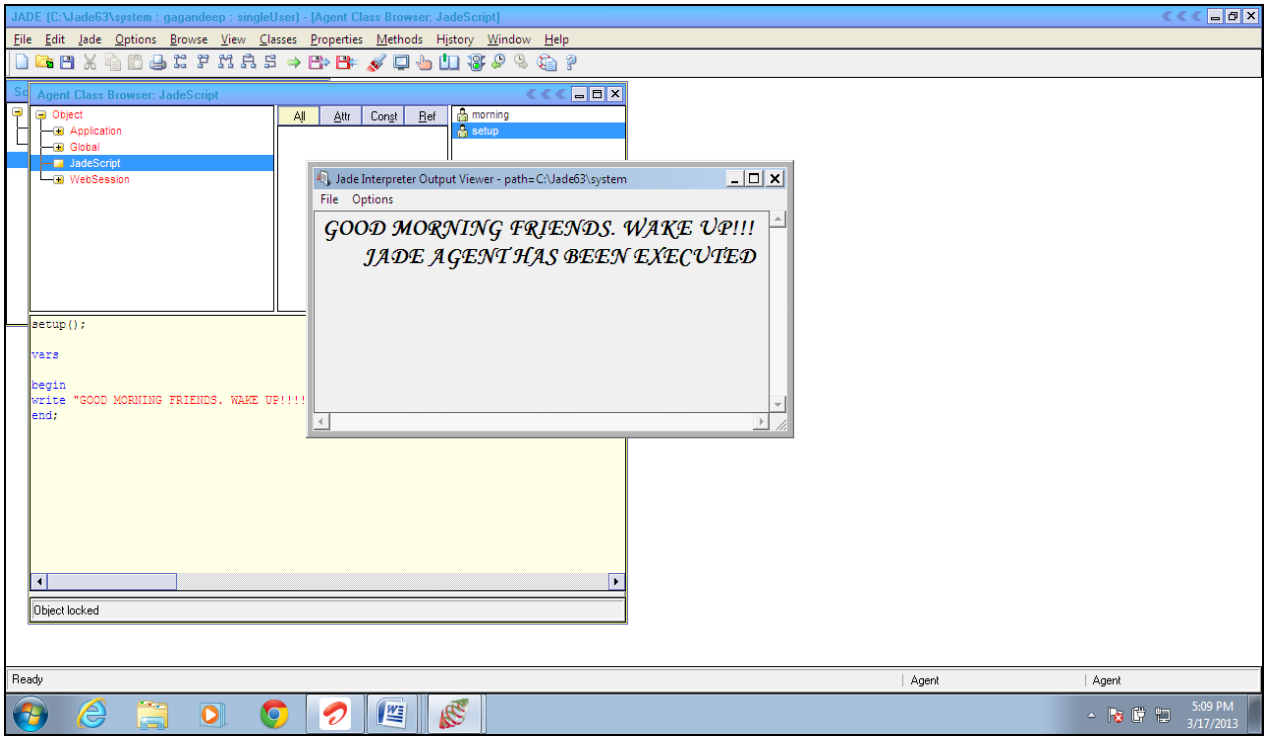

Figure 6: Execution of Agent displaying Good Morning message using JADE 6.3.11

Table 1: Different versions of JADE [8]

VERSION	FEATURES
2.2	Composite Behavior is renamed as Complex Behavior
2.4	Some classes are added like Achieve REInitiator/Responder. It gives implementation of all FIPA-Request Interaction Protocols.
2.5	NetInitiator/Responder is implemented that offers API with its functionalities.
2.6	A bug was introduced that reduces scalability.
2.6.1	Bug was fixed. This version is used for stress testing.
3.1	PROPOSE Initiator/Responder method is introduced.
3.5	It supports Topic- Based Communication i.e. it is possible to send messages about a given topic. It is implemented by jade.core.messaging.Topic
6.3.03	It is newer version of JADE that has concept of Database Connectivity, ODBC Driver and Web enabled applications. It is available in various modes: Single, Multi-User and Client (Fat client, standard) Mode.
6.3.11	In addition to 6.3.03, it introduces new features like RoseJADELink, JADE Report Writer. We can create object model using UML and easily convert into JADE class structure using Rational Rose Enterprise Edition. JADE Report Writer enables us to configure and design reports for all schemas in JADE database.
7.0	Features are: f) Impedance mismatch: - JADE reduces developing effort required to move objects from database resulting in increasing productivity. g) It has RPS (Relational Population Service) that allows automatic conversion of objects from JADE database to other databases. h) It has SDS (Synchronized Database Service) that allows secondary database to synchronize with primary database. i) High Performance and Object Caching is available.

There is need to classify all elements that appears within sentences sent by agent as content of ACL Message. *Ontology for a given domain is defined as set of schemas that defines the structure of elements of Content Reference Model [9]*. The elements of Reference Model are as follows:

a. ***Predicates: -*** They are the expressions that tell about the validity of given sentence. It can be true or false. E.g. (works – for (Person: name Vishal) (Company: name Accenture)). It means that person Vishal works for company Accenture.

b. ***Terms: -*** They refer to entities.

c. ***Concepts:-*** They are entities having defined structure. E.g. (Person: name Vishal: age 30)

d. ***Agent Action:-*** They are special concepts that indicate actions performed by same agents. E.g. (Buy Soap: "Cinthol")

- ***e. Primitives:*** **-** They are the expressions that hold value like strings and integers.
- ***f. Aggregates:-*** They are expressions consisting of nested entities.
- ***g. IRE (Identifying Referential Expressions):*** **-** They are expressions that identify entity for which predicate is true.
- ***h. Variables:*** - They are generic elements.

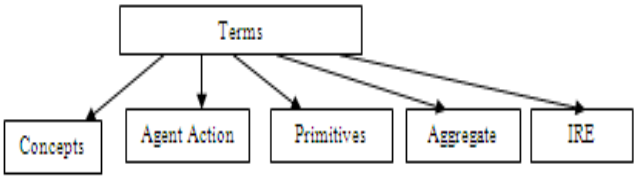

Figure 7: Categories of Terms

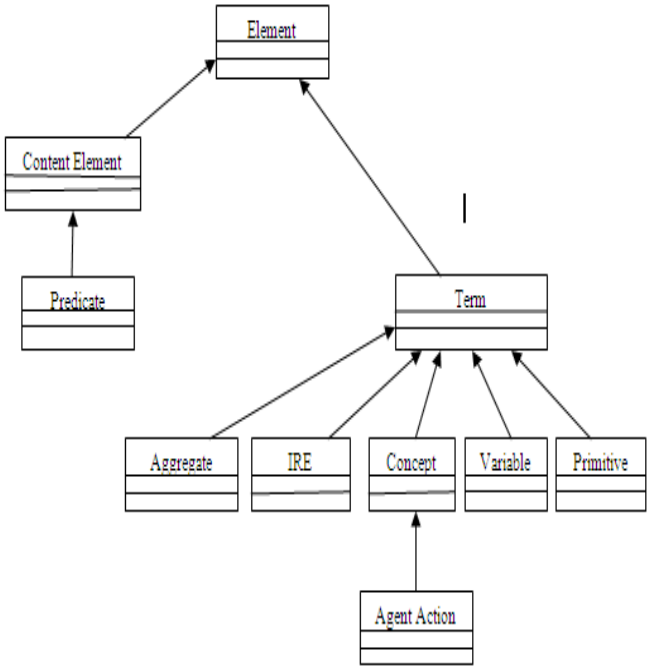

Figure 8: Content Reference Model [4]

A. Defining Ontology:

Here we have defined ontology by using Bakery Shop Example. It considers a seller agent that manages bakery shop where items like (Cakes, Biscuits) are available for sale. Each item has ID. Each cake has shape, weight and flavor. Each biscuit has its name. The seller agent OWNS a number of items and want to sell them to buyer agents. There are three concepts in this example (Item, Cake and Biscuit), one Predicate (Owns) and one Agent Action (sell). There is class named jade.content.onto.Ontology that contains all schemas of given example.

```
Package bakeryShopOntology;
import jade.content.onto.*;
import jade.content.schema.*;
public class BakeryShopOntology extends Ontology
{
public static final String ITEM = “item”;
public static final String ITEMID = “idnumber”;
public static final String CAKE = “Cake”;
public static final String CAKESHAPE = “cakeshape”;
public static final String CAKEWEIGHT = “cakeweight”;
public static final String CAKEFLAVOR = “cakeflavor”;
public static final String BISCUIT = “Biscuit”;
public static final String BISCUITNAME = “name”;
public static final String OWNS = “Owns”;
public static final String OWNSOWNER = “owner”;
public static final String OWNSITEM = “item”;
public static final String SELL = “Sell”;
public static final String SELLBUYER = “buyer”;
public static final String SELLITEM = “item”;
private static Ontology theInstance = new
BakeryShopOntology();
public static Ontology getInstance()
{return theInstance;}
Private BakeryShopOntology ()
{super(OntologyName, BasicOntology.getInstance())
try{
add (new ConceptSchema(ITEM), Item.class);
add (new ConceptSchema(CAKE), Cake.class);
add (new ConceptSchema(BISCUIT), Biscuit.class);
add(new ConceptSchema(OWNS), Owns.class);
add(new ConceptSchema(SELL), Sell.class);
ConceptSchema cs = (ConceptSchema) getSchema(ITEM);
cs.add(ITEMSERIAL, (Primitive Schema)
getSchema(BasicOntology.INTEGER);
ConceptSchema cs = (ConceptSchema) getSchema(CAKE);
cs.addSuperSchema((Concept Schema) getSchema(ITEM));
cs.add(CAKESHAPE, (Primitive Schema)
getSchema(BasicOntology.STRING);
cs.add(CAKEWEIGHT, (Primitive Schema)
getSchema(BasicOntology.INTEGER);
cs.add(CAKEFLAVOR, (Primitive Schema)
getSchema(BasicOntology.STRING);
ConceptSchema cs = (ConceptSchema)
getSchema(BISCUIT);
cs.add(BISCUITNAME, (Primitive Schema)
getSchema(BasicOntology.STRING);
PredicateSchema ps = (PredicateSchema) getSchema(OWNS);
ps.add(OWNSOWNER, (ConceptSchema)
getSchema(BasicOntology.AID));
ps.add(OWNSITEM, (ConceptSChema) getSchema (ITEM));
AgentActionSchema as = (AgentActionSchema)
getSchema(SELL);
as.add(SELLITEM, (ConceptSchema) getSchema(ITEM));
as.add(SELLBUYER, (ConceptSchema)
getSchema(BasicOntology.AID));
}Catch (Exception e){e.printStackTrace();}}}
```

V. CONCLUSIONS

This paper concludes that JADE can be used efficiently in retrieving information from a given domain. JADE is quite easy to learn and use. It ensures development process by providing set of system services and applications. JADE has various packages that support advanced features of JADE like supporting ontologies and content languages. We have defined

ontology by using the example of Bakery Shop where different schemas defining predicates, concepts have been added. The paper describes the process of communication among agents and their execution. As a future work, we can combine the above ontology and converts it into database using some patterns and semantic constraints

VI. ACKNOWLEDGMENT

I Vishal Jain, would like to give my sincere thanks to Prof. M. N. Hoda, Director, BVICAM, New Delhi for giving me opportunity to do P.hD from Lingaya's University, Faridabad.

VII. REFERENCES

[1]. Vishal Jain, Gagandeep Singh, Dr. Mayank Singh, "Implementation of Multi Agent Systems with ontology in Data Mining", "*International Journal Of Research In Computer Application & Management (IJRCM), Vol. No. 3, Issue No.1 ISSN 2231-1009*", January 2013, pp 111-117

[2]. Gagandeep Singh, Vishal Jain, "Information Retrieval (IR) through Semantic Web (SW):An Overview", "*In proceedings of CONFLUENCE 2012- The Next Generation Information Technology Summit at Amity School of Engineering and Technology*", September 2012, pp 23-27.

[3]. Fabio Bellifemine, Agostino Poggi, "JADE- A FIPA – compliant agent framework", In proceedings ICMAS'95, San Francisco, USA 1995.

[4]. JADE Tutorial- Application Defined Content Languages and Ontologies, 2002.

[5]. H. Akkermans, "Intelligent E-Business –From Technology to Value."IEEE Intelligent Systems, 16 (4), pp 8-10, 2001.

[6]. Marat Kanteev , Igor Minakov , George Rzevski , Petr Skobelev , Simon Volman, "Multi-Agent Meta-Search Engine Based on Domain Ontology"

[7]. Vishal Jain, "Information Retrieval through Multi- Agent Systems with Data Mining in Cloud Computing", Int.J.Comp.Tech.Appl, ISSN: 2229-6093 Vol.3 (1), pp 62-66

[8]. JADE, Java Agent Development Environment, 2006, http://jade.tilab.com

[9]. Kobkul, Gareth Clayton and Phayung Meesad, "A Hybrid System based on Multi Agent System in the Data Preprocessing Stage", International Journal of Computer Science and Information Security (IJCSIS), Vol.7, Issue 2, pp 199-202, 2010.

[10]. Vishal Jain and Mahesh Kumar Madan, "Multi Agent Driven Data Mining For Knowledge Discovery in Cloud Computing", International Journal Of Computer Science and Information Technology Research Excellence, ISSN: 2250-2734, Vol.2. Issue 1, Jan-Feb 2011

[11]. Urvi Shah, James Mayfield, "Information Retrieval on the Semantic Web", "*ACM CIKM International Conference on Information Management*", Nov 2002

[12]. Learning JADE, http://www.jade.co.nz/jade/tutorials.htm

[13]. Rajesh and Shiv Kr. Tayal, "An Intelligent Multi Agent Framework for E-Commerce Using Case Based Reasoning and Argumentation for Negotiation", VSRD-IJBMR, Vol. 2 (6), 2012, 232-243

[14]. Ryan Ribeiro de Azevedo, Eric Rommel Galvão Dantas, Fred Freitas, Cleyton Rodrigues, Marcelo J. Siqueira C. de Almeida, Wendell Campos Veras and Rubean Santos, "An Autonomic Ontology-Based Multiagent System for Intrusion Detection in Computing Environments", International Journal for Infonomics (IJI), Volume 3, Issue 1, March 2010

[15]. Bellifemine, F., Caire, G., and Greenwood, D. (2007). Developing Multi-agent Systems with JADE. Wiley Series in Agent Technologies. Wiley Inter-Science.

[16]. Andrew B. Williams, "The Role of Multiagent Learning in Ontology-based Knowledge Management", American Association for Artificial Intelligence (www.aaai.org)

Short Bio Data for the Authors

Vishal Jain has completed his M.Tech (CSE) from USIT, Guru Gobind Singh Indraprastha University, Delhi and doing P.hD from Computer Science and Engineering Department, Lingaya's University, Faridabad. Presently he is working as Assistant Professor in Bharati Vidyapeeth's Institute of Computer Applications and Management, (BVICAM), New Delhi. His research area includes Web Technology, Semantic Web and Information Retrieval. He is also associated with CSI, ISTE.

Dr. Mayank Singh have done his M. E in software engineering from Thapar University and PhD from Uttarakhand Technical University. His Research areas are Software Engineering, Software Testing, Wireless Sensor Networks and Data Mining. Presently He is working as Associate Professor in Krishna Engineering College, Ghaziabad. He is associated with CSI, IE (I), IEEE Computer Society India and ACM.