

Ontology Based Agent Communication in Resource Allocation and Monitoring

Manish Arora[1] and M. Syamala Devi [2]

**[1] DOEACC Society Chandigarh Centre,
Chandigarh, India**

**[2] Department of Computer Science and Application, Panjab University
Chandigarh, India**

Abstract

The aim of ontology is to share information between sending and receiving agents of Multi Agent System (MAS). It provides standard vocabulary and terms for knowledge sharing and is designed to share information conveniently and understandably. Agent based application requires complex interaction among agents. This complexity is due to agent-agent and agent-user communication. It is required to use ontology in agent based application of resource allocation and monitoring. The purpose of Resource Allocation and Monitoring System is to make the procedures involved in allocating fund resources to competing fund seekers transparent so that deserving candidates get funds. Proactive and goal directed behaviour of agents make the system transparent and intelligent. This paper presents ontology designed and implemented for the purpose of communication among agents of Multi Agent System for Resource Allocation and Monitoring (MASRAM). FIPA (Foundation for Intelligent Physical Agents) compliant software JADE (Java Agent Development) is used to implement ontology.

Keywords: *Multi Agent System, Resource Allocation, ACL, FIPA, JADE, Ontology.*

1. Introduction

Ontology is used to represent knowledge that is shared between different entities. It provides terms and vocabulary used to represent knowledge so that both sender and receiver can understand. Ontology is widely used in many areas like MAS and Biomedical Informatics to share knowledge. The study of MAS focuses on systems in which many intelligent agents interact with each other. The agents are considered to be autonomous software or hardware entities, such as software programs. Their interactions can be either cooperative or selfish. Agent acts on the behalf of users / other agents with different goals and motivations. Agents require ability to cooperate, coordinate and negotiate with each other to complete task successfully [1]. Agents work independently but share information. Ontology helps designers of agent based systems to make information understandable between agents. Ontology is applied in one such agent based application, resource allocation and monitoring.

Resource allocation problem occurs when fixed and limited resources are allocated to competing fund seekers to execute their projects. These resources may be of different types like work force, machine timings, raw material and funds. Fund seekers can submit their project proposals to avail grant to allocating agencies. Projects can be of different nature like R & D projects and social oriented schemes. On receiving the project proposals from fund seekers, fund allocating agencies evaluate proposals technically as well as financially. After the submission of fund request, committee on the behalf of funding agency evaluates proposals. In some cases, fund seekers are asked to present the project proposal. Based on the recommendation of committee, funds are allocated by considering both quantifiable and non-quantifiable factors.

Considering above facts, an integrated decision making system ‘Multi Agent System for Resource Allocation and Monitoring’ (MASRAM) is designed and agents of system require information sharing. This paper describes the ontology based communication between agents of MASRAM. The designed ontology is implemented in JADE.

The paper has been organized into different sections. Second section reviews the related research, third describes the model, fourth details construction of ontology and fifth shows how ontology is implemented in JADE.

2. Review of Related Research

A fundamental characteristic of MAS is communication among different agents operating in the system. Agents exchange information in order to achieve their goals. Message follows Agent Communication Language (ACL) standard which allows encoding/decoding of actual message. The structure of the message is set of terms written in FIPA-ACL like message content, message parameters, encoded message and transports layer information. Contents of messages are written in content language such as FIPA-SL (Semantic Language) and FIPA-KIF (Knowledge Interchange Format). FIPA-ACL is based on Speech Act Theory which means that message represents action or communication act, also known as performative act. Other commonly used communication acts are inform, request, agree and refuse [2].

When agents in MAS communicate with each other, message is sent and main component of message is content slot. According to FIPA specification, value of this slot could be either string or raw sequence of bytes. In real world application, agent needs to send complex information to the receiving agent like list of agencies providing funds. In such scenario, a well defined syntax of the message content is adopted so that both sender and receiver agents can understand and share information. The concept is called Content Language. Two kinds of languages are used SL and Leap. Our research work is based on FIPA-SL content language. This language is used to define concept and symbols used in content of the message and is known as Ontology. Ontology is agreement about shared conceptualization that includes framework for modelling application specific contents for communication among agents [3]. The aim is to clarify meaning of the message for exchange.

Jenyl Mumpower and Thomas A Darling [4] have discussed three procedures that can be used to resolve Resource Allocation problem. In Incremental Appropriation, resource allocation begins with no allocation and then allocates small resources. The process is repeated until resources are exhausted. In the second procedure, multiple negotiators give different concessions. Resources are moved from one point to another and utility function is checked. In the third procedure, different negotiators assign different weights to different programmes.

Quantification of non-quantitative factors is important to make decision of allocation [5]. The non-quantifiable factors can be measured through fuzzy comprehensive measurement method. Since non-quantifiable factors are measured by human whose knowledge and experiences may not be exact or complete. The probabilistic tools are used to deal with such data. This approach is also used to rank employees' performance using both quantitative and non-quantitative measures.

Monitoring is very important factor to know the utilization of the funds, benefits gained from funding and giving further financial help.

Various agent oriented tools are available to develop intelligent agents that include basic services like communication act. One of them is JADE. It is FIPA compliant tool and same is used to develop agents defined in MASRAM problem. JADE simplifies the implementation of MAS through middle tier. Message in JADE can be passed through string data type, Java object or ontology. The focus of research is to construct and implement ontology comprising complex information in JADE [6].

From review, it has been found that MAS systems are widely used in resource allocation problems, such as transportation, scheduling, production planning and system resources in which ontology plays important role in communication between agents [7, 8].

3. MASRAM Model

Three agents have been designed for MASRAM problem. At the abstract level these agents are:-

- Coordinator Agent
- Fund Seeker Agent
- Fund Allocator and Monitor Agent

3.1 Coordinator Agent

Coordinator Agent interacts with three types of users of MASRAM i.e. Fund Seeker user, Fund Allocator user and Reviewer user. Fund Seeker user seeks funds, Fund Allocator user allocates funds and monitors the utilization while Review user reviews the proposal. Coordinator Agent forwards requests received from Fund Seeker user to Fund Seeker Agent. Coordinator Agent also forwards requests received from Fund Allocator user and Reviewer user to Fund Allocator and Monitor Agent. To summarize, this agent interacts with following agents/users.

- Fund Seeker User
- Fund Allocator User
- Reviewer User
- Fund Seeker Agent

- Fund Allocator and Monitoring Agent

3.2 Fund Seeker Agent

Fund Seeker Agent receives all the requests received from Coordinator Agent and act accordingly. This agent interacts with Coordinator Agent only.

3.3 Fund Allocator and Monitor Agent

Fund Allocator and Monitor Agent in turn evaluates proposal, assigns weights and allocates suitable funds based on allocation procedure. Fund Allocator and Monitor Agent (FAMA) interacts with Coordinator Agent only.

Agents and overview of the interaction among them have been shown is figure 1.

4. Design of Ontology

Ontology for multi agent based application described above is designed and implemented in JADE. After the agents have been defined, next step considered was to design ontology that includes set of concepts and symbols. Table 1 describes the messages passed to Fund Seeker Agent from Coordinator Agent along with type of communication act e.g. *source_type* is message of communication act 'request'. Coordinator Agent uses this message to request Fund Seeker Agent to provide list of funding agencies based on particular type of project. Similarly *fund_utilization* message is used to inform Fund Seeker Agent the status of utilization. Other content messages are also designed based on responsibilities of agent in similar fashion.

Ontology was also designed for Coordinator and Fund Allocator and Monitor agent to communicate with each other. Table 2 describes one such content message, *fund_category* that is passed to Fund Allocator and Monitor Agent by Coordinator Agent. Coordinator Agent informs various fund categories out of which funds can be provided.

Table 3 and 4 show some responses given by Fund Seeker Agent and Fund Allocator and Monitor Agent after receiving requests from Coordinator Agent.

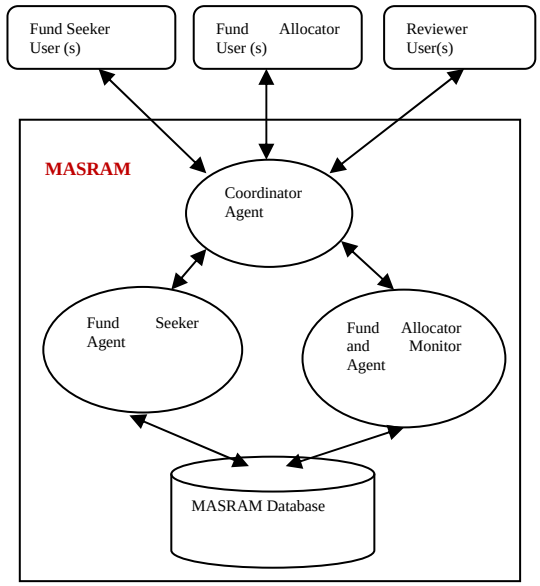

Fig 1. Multi Agent System Resource Allocation and Monitoring Model.

Table 1: Message passing (coordinator-fund seeker)

Sender Agent: Coordinator ***Receiver Agent: Fund Seeker***	
Content Messgae	Act
Source_type	Request
Fund_utilization	Inform

Table 2: Message passing (Coordinator-Fund Allocator)

Sender Agent: Coordinator ***Receiver Agent: Fund Allocator and Monitor***	
Content Message	Act
Fund_category	Inform

Table 3: Message passing (Fund Seeker-Coordinator)

Sender Agent: Fund Seeker ***Receiver Agent: Coordinator***	
Content Message	Act
Available_source	Inform
Proposal_id	Inform

Table 4: Message passing (Fund Allocator and Monitor-Coordinator)

Sender Agent: Fund Allocator and Monitor ***Receiver Agent: Coordinator***	
Content Message	Act
Allotted_fund_categor y	Inform
Review_proposal	Request

One of the content messages described above is detailed in table 5.

Table 5: Ontology

```
Available_source
(Inform
  : sender (agent-identifier :name FSA@manish:1099/JADE)
  : receiver (agent-identifier :name COA@manish:1099/JADE)
  : ontology ra-ontology
  : language fipa-sl
  : content
  (
    Available_source: (Sequence (
    (
      nature_of_project_id : 1
      Nature_of_project_desc: Information and Technology
      allocator_id: 2001
      allocator_desc: Department of Information Technology
      allocator_address: New Delhi
      criteria: (sequence
                  ( 1, 'Number of Technical Staff Members',5)
                  (2, 'Projects  Handled', 5)
     )
     Fund_available : (sequence  ( 500000))
    )  )
)
)
```

The message shows that Fund Seeker agent named FSA wanted to send list of allocators who are funding project related with 'Information and Technology'. It sends criteria set by allocator. Agent also sends availability of funds. Ontology mentioned here is application specific ontology (*ra-ontology*). On the similar lines remaining messages are defined.

5. Implementation in JADE

To implement agents and ontology, JADE tool is used [9]. Following steps describe implementation procedure.

5.1 Developing Ontology

Ontology in JADE is instance of *jade.content.onto.Ontology* class. Ontology is collection of schemas. For MASRAM problem, *conceptSchema* is used in Java code that extends *Ontology* class of JADE. Table 6 shows the code written in Java. Name of the ontology is defined as *ra-ontology*. Five schemas have been defined. LOGIN and PWD schema are used for user verification, PROJ_NATURE is used to pass type of the project e.g. Education and Research, PROJ-SOURCE is used to send list of the fund allocators along with their criteria and availability. Lastly PROJ_LIST is used to send list of project types. First three schemas are of primitive in nature while remaining two are of the aggregate schema in which list is passed as message. *ObjectSchema.UNLIMITED* means any number of rows can be sent.

5.2 Developing Java Classes

While implementing ontology in JADE, each schema is associated with java class that implements *AgentAction* class of JADE. Each Schema can have one Java class for each schema or can have single Java class for multiple schemas. In MASRAM case, one single java class file is used. Each schema has two public declared methods. One method is *setXXXX* and other is *getXXXX* where *XXXX* is name of schema. These two methods are used to set and retrieve the values respectively as briefed in table 7.

Table 6: Defined ontology listing

```
// import statements
public class RAOntology extends Ontology
{
    public static final String ONTOLOGY_NAME = "ra-ontology";
    public static final String LOGIN = "Login_id";
    public static final String PWD = "Login_pwd";
    public static final String PROJ_SOURCE = "Proj_source";
  …
    public static Ontology getInstance ()
    {return theInstance;    }
    private RAOntology()
    {
      super(ONTOLOGY_NAME,BasicOntology.getInstance());
      try           {
        add(new ConceptSchema(LOGIN),Login.class);
        ConceptSchema cs = (ConceptSchema)getSchema(LOGIN);
cs.add(LOGIN,(PrimitiveSchema)getSchema(BasicOntology.STRIN
G));
cs.add(PWD,(PrimitiveSchema)getSchema(BasicOntology.STRING),
ObjectSchema.OPTIONAL);
…
cs.add(PROJ_SOURCE,(AggregateSchema)getSchema(BasicOntolog
y.SEQUENCE),0,ObjectSchema.UNLIMITED);
..
```

5.3 Passing Message Using Ontology

Third step is to call defined ontology in Java agent program to fill and send message. Table 8 shows the important lines of code of setting message. The java program imports ontology defined earlier along with other packages. *RAOnotlogy.getInstance()* in code sets user defined ontology.

Table 7: Listing of Java class

```
// import statement
public class Login implements AgentAction
{
        String login_id;
        String login_pwd;
        List proj_source;
        …
        public void setProj_list(List l)
        {       proj_list=l;  }
        public List getProj_list()
        {       return proj_list;       }
…
```

Table 8: Listing of setting message

```
…
Login lg = new Login();
  Types ty = new Types();
ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
getContentManager().registerLanguage(codec);
getContentManager().registerOntology(ontology);
…
        List l = new ArrayList();
        msg.setLanguage(codec.getName());
        msg.addReceiver(new AID(board.getReceiver(),
AID.ISLOCALNAME) );
      msg.setLanguage(codec.getName());
        msg.setOntology(ontology.getName());
        msg.setOntology(ontology.getName());
        action.setActor(new AID(board.getReceiver()));
       action.setAction(lg);
        this.getContentManager().fillContent(msg,action);
        send(msg);
…..
```

6. Conclusion

This paper detailed the way agents interact with each other through ontology. During construction of ontology, FIPA specifications were followed and implemented in FIPA compliant agent development framework, JADE. Ontology was developed in java classes by importing JADE packages. A three layer approach is used. In first layer, java class file defines terms (objects) to be used. In second layer, Ontology and schemas are defined and in third layer, message is filled. This method of ontology found suitable to exchange complex information like multiple records and made them understandable.

7. Scope for Future Work

Future plan includes implementing fund allocation algorithm in JADE so that agent can allocate funds among competing fund seekers. The complete developed MASRAM will be tested to validate the work done.

References

[1] Jarg Denzinger, “Multi Agent Systems”, Department of Computer Science, University of Calgary, Canada, available at http://pages.cpsc.ucalgary.ca/%7Edenzinge/courses/567-winter2006/slides/04-masdef-handout.pdf, accessed on April 11, 2007.

[2] History of FIPA, available at http://www.fipa.org/subgroups/ROFS-SG-docs/History-of-FIPA.htm accessed on Sep 11, 2010

[3] Prashant M, “Integrating Ontologies into Multi-Agent Systems Engineering (MaSE) for University Teaching Environment”, Journal of Emerging Technologies in Web Intelligence, Vol 2, No 1, 2010, pp. 42-47.

[4] J. L. Mumpower, and T. A. Darling, “Modeling Resource Allocation Negotiations”, in IEEE Twenty Fourth Annual Hawaii International Conference, 1991, Vol. 3, pp. 641-649.

[5] J. Cheng, H. B., and Ziping Li, “Quantification of Non Quantitative Indicators of Performance Measurement”, in 4th International Conference on Wireless Communications, Networking and Mobile Computing WiCOM '08, 2008, pp. 1-5.

[6] JADE Administration’s Guide, available at http://jade.tilab.com/doc/administratorsguide.pdf, accessed on June 30, 2007.

[7] V. Gorodetski, Oleg Karsaev, and Victor Konushy, “Mulit Agent System for Resource Allocation and Scheduling”, in 3rd International Workshop of Central and East European conference on Multi Agent System, Prague, Czech Republic, 2003, Vol. 2691/2003, pp. 1067.

[8] Anthony Chavez, Alexandros Moukas and Pattie Maes, “Challenger: A Multi Agent System for Distributed Resource Allocation”, in First International Conference on Autonomous Agents (Agent97): Marina Del Ray, California: ACM Press, 1997, pp. 323-331.

[9] Fabio Bellifemine et el, “JADE – A FIPA Compliant agent framework”, available at www.citeseerx.ist.psu.edu /viewdoc/download?doi=10.1.1.86.6371 &rep=rep1&type=pdf , accessed on July 20, 2010.

Manish Arora holds MCA (GNDU, Amritsar), MBA in Operation Management (IGNOU) and ‘C’ Level (M. Tech) from DOEACC Society, New Delhi. He has nearly two decades experience in teaching, software development and consultancy. Presently, he is working as Principal Systems Analyst in DOEACC Society, Chandigarh Centre and managing different government projects. He has published 3 papers on multi agent technologies in international journals and conferences.

Dr. M. Syamala Devi is a professor in the Department of Computer Science & Applications, Panjab University, Chandigarh (India). She received her Ph.D degree in Computer Science and Systems Engineering from Andhra University, Visakhapatnam and M. E. in Computer Science & Engineering, from NIT, Allahabad. She had also served ISRO, Sriharikota, and NITTTR, Chandigarh. Her areas of expertise include algorithms Design and analysis, Image Processing, Distributed AI. She has about fifty national and international research papers to her credit.