ISSN:0975-9646

Role of Database in Multi Agent Resource Allocation Problem

Manish Arora[#], M. Syamala Devi[*]

[#]DOEACC Society, Chandigarh Centre Chandigarh-India

[*]Panjab University Chandigarh- India

***Abstract*—Multi-agent technology, a modern computer science tool, is widely used in developing intelligent systems. Agents in multi-agent system are autonomous and capable of doing tasks independently, however they share information with each other to achieve their respective goals. Information may be a result of task or a request to other agent to perform some task. To develop an integrated system, database forms basis. This paper presents the role of database in resource allocation problem in multi-agent environment. The aim of problem is to allocate funds to most deserving and competing funds seekers. Agents of multi-agent resource allocation system share information with back support of database. Agents are developed in Java Agent Development Framework (JADE). Instance of Java class in agents is used to interact with database. This methodology to integrate database and agents has been found useful in solving resource allocation problem.**

***Keywords*— Agent, Multi-agent System, JADE, FIPA, Resource Allocation, JSP, User Defined Type, Intelligent System, database, DML.**

I. Introduction

The characteristics of Object Relational Database Management System (ORDBMS) have motivated to use as backend support in multi-agent based applications. The capabilities of such databases to store attributes and methods of class as User Defined Type (UDT) and to use them further as column data type in table is another reasonable factor to support agents of resource allocation problem. Multi Agent System (MAS) consists of multiple agents operating in environment in a cooperative manner. Agents in MAS are software entities and can be reactive, proactive or mobile. Agents require information to be shared with each other. Agents have individual goal to achieve, but individual goal ultimately leads to achievement of overall system goal.

Multi Agent System for Resource Allocation and Monitoring (MASRAM) is an agent based system with the support of database to allocate funds to most deserving fund seekers after evaluating their proposal technically and financially. System takes care of both quantifiable and non-quantifiable decision making factors to arrive at a decision. Another goal of the system is to monitor the progress of project and utilization of funds. Decision making factors are assigned weights according to their preferences and then proposals are ranked for allocation. System has three agents: Fund Seeker Agent, Fund Allocator & Monitor Agent and Coordinator Agent.

The purpose of this study was to develop and implement a methodology to integrate agents of MASRAM with object relational database. Java Agent Development Framework (JADE), a FIPA (Foundation for Intelligent Physical Agents) compliant tool is used to develop agents. Oracle 9i is used as backend database for information sharing. Java is used to build business objects that hide the detailed logic and became interface between database and agents. This methodology to integrate agents and database found working effectively. To interact with users of the system: fund seeker, fund allocator and reviewer, JSP is used. Users can put request to this application and receive response from it.

The rest of the paper is organized as follows: Section 2 details background and related work. Section 3 highlights importance of database in system. Section 4 discusses database support for agent. Section 5 describes methodology to implement database support to agents. Finally section 6 summarizes conclusion and scope for future work.

II. Background and Related Work

Resource allocation problem occurs when limited funds are to be allocated to deserving and competing fund seekers. Fund seekers seek funds to execute their projects and/or social schemes. Fund seekers submit their proposal to funding agencies for availing funds. Monitoring is also associated with resource allocation problem to know the progress of project and utilization of funds [1].

Three methods are used to allocate funds: incremental, negotiation and weighting [2]. Quantifiable and non - quantifiable factors are used in decision making process of funds allocation [3]. Non quantifiable factors like image are converted into quantifiable. The proposals are reviewed from technical and financial angles. The proposal with high technical effectiveness and best cost effectiveness is given priority over others [4].

Agent is an autonomous entity which performs a given task using information gathered from environment to act in a suitable manner so as to complete task successfully. Agents have characteristics of reactive, proactive, collaboration, adaptively and mobility. Multi Agent System consists of number of agents that interact with each other. Agents act on the behalf of users and/or other agents. Each agent has different goal to achieve. They cooperate, coordinate and negotiate with each other to achieve goal.

Multi-agent technology is used in various decision making applications like production, logistic, transportation and other resource allocations [5, 6]. Agents share information with the help of backend support of database in such complex business applications. Role of database in agent application is to support agents in information sharing. Agents in MAS cannot work without cooperating each other despite the fact that they are autonomous and capable of solving complex problem. Agents use task sharing and information sharing to cooperate with each other. In both the cases, database plays important role [7].

ORDBMSs are widely used for data storage and manipulation as these are strong on handling real world application and in this category, Oracle 9i is popular database [8]. Oracle has collection objects like nested table and Varrays. With nested tables, it is easy to store information of one object in a single table. Another motivation behind ORDBMS to act as backend support for agents is elimination of normalization [9, 10]. Moreover there are conceptual similarities between agent technologies and object relational database e.g. tasks to be performed by agents can be mapped to methods of object in Oracle. Data object holds data. It may be a single data item or complex data structure. Agents use these objects.

Databases are widely used in various agent based applications such as cross enterprise resource planning for small and medium enterprises, Airline Operation Control System and health sectors [11, 12, 13]. Use of databases in these applications has made the communication between agents simple and more effective.

Based on the background and related work, an intelligent agent based system, MASRAM was designed and developed. The agents are implemented in JADE with backend support of Oracle. Implementation of user interface for resource allocation problem using JSP is described in [14].

III. Importance of Database in Resource Allocation Problem

Oracle database forms the basis for effective communication and information sharing between agents of resource allocation problem. With database, agents of the resource allocation problem can share information with each other and request other agents to perform task. In this section, agents of resource allocation problem are introduced followed by role of database in solving problem.

A. Agents of Resource Allocation Problem

Three agents are described below:-

1) Coordinator Agent: Coordinator Agent interacts with three types of users. Coordinator Agent forwards requests received from fund seeker user to Fund Seeker Agent or to Fund Allocator and Monitor Agent depending upon request and type of end user. Results are shown back to users.

2) Fund Seeker Agent: Fund Seeker Agent receives all the requests from Coordinator Agent and act accordingly. This agent interacts with Coordinator Agent and/or with database.

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 processes all the requests received from Coordinator Agent. This agent also share information with help of database.

Diagrammatically, the system has been shown in figure 1. Fund Seeker Agent and Fund Allocator & Monitor Agent require interaction with ORDBMS while Coordinator Agent interacts with three types of users. Table 1 shows the required database interaction.

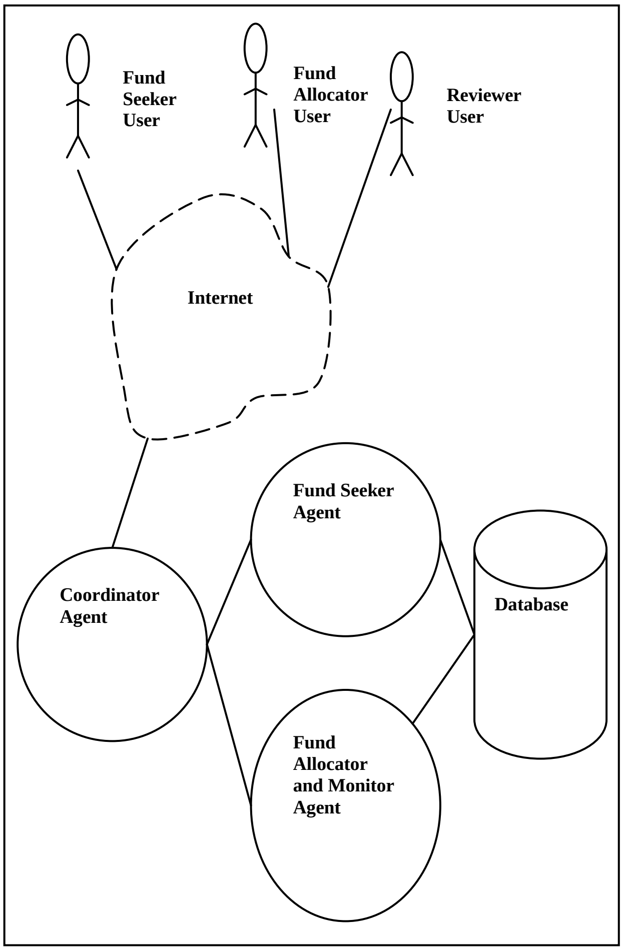

Fig. 1 System Model

B. Role of Database

Agents perform different tasks at different time and moreover their tasks are interleaved while processing request of user of system, so it becomes important to share information while processing request. To share this information, database, oracle 9i is used. Agent also puts information in database and requests another agent to perform some task e.g. Fund Seeker Agent saves proposal on the behalf on fund seeker user in database and Fund Allocator & Monitor Agent picks proposal from database, evaluates it and allocates funds. After allocation, status is kept in database from where Fund Seeker Agent informs fund seeker user about allocation status through Coordinator Agent. Such information will remain available even if agents are not live for some time due to some reason.

TABLE 1

TASKS PERFORMED BY AGENTS

Agent	Task
Fund Seeker	1. Storing and validating user credentials 2. Accepting proposal 3. Finding status 4. Finding funds sources 5. Updating utilization status
Fund Allocator and Monitor	1. Evaluating proposal 2. Assigning appropriate weights 3. Allocating funds 4. Reviewing utilization Status

IV. DATABASE SUPPORT FOR AGENTS

Based on agents and their requirements described above, data objects are created in object relational database, Oracle to support agents. Table 2 describes schema objects created to strengthen Fund Seeker and Fund Allocator & Monitor agents to perform tasks.

TABLE 2

SCHEMA OBJECTS CORRESPONDING TO TASKS

Schema Object	Task
Login	1. Storing and validating user credentials
Allocator	1. Finding funds sources 2. Evaluating proposal 3. Assigning appropriate weights 4. Allocating funds
Proposal	1. Accepting proposal 2. Finding Status 3. Updating utilization status

Agents developed in JADE interact with database through business object developed in Java. This is class file and does following tasks.

- It takes input from agents.
- It validates input from.
- It makes connection with Oracle ORDBMS.
- It performs calculation as per requirement.
- It runs Data Manipulation Language (DML) statement to operate at database level.
- It returns result to agents.

The complete sequence of events has been depicted in figure 2.

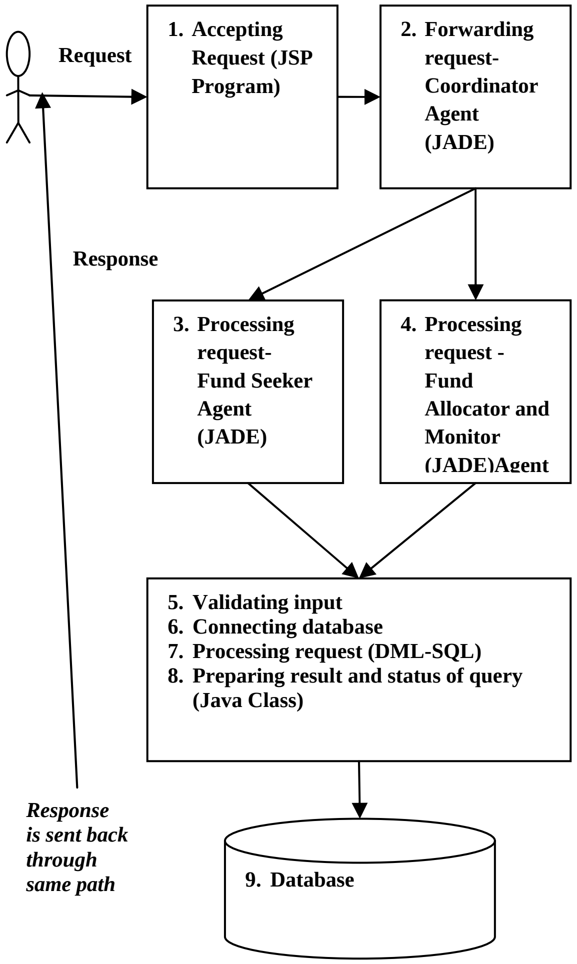
Request
1. Accepting Request (JSP Program)
2. Forwarding request- Coordinator Agent (JADE)
Response
3. Processing request- Fund Seeker Agent (JADE)
4. Processing request - Fund Allocator and Monitor (JADE)Agent
5. Validating input
6. Connecting database
7. Processing request (DML-SQL)
8. Preparing result and status of query (Java Class)
Response is sent back through same path
9. Database

Fig. 2 Sequence of Events

V. IMPLEMENTATION

The complete browser based system has been implemented by using software: JADE, JSP and Oracle 9i database. In the three layered approach to develop and implement system, front end application has been developed in JSP, agents have been implemented in JADE and schema objects in Oracle. Implementation of front end and agents is beyond scope of this paper. The partial structure of table with user defined data types is shown in Table 3. Instance of java class *'objectclasses.class'* is created in agents (Both Fund Seeker and Fund Allocator & Monitor). The methods of object in agent class make connection with database and perform DML statement. Table 4 shows the partial code of Fund Seeker Agent calling method of object to validate data. Table 5 shows partial code of *'objectclasses.java'* to validate data and Table 6 shows partial code of storing information in Oracle 9i database.

TABLE 3

DATA OBJECTS STRUCTURE

```
create or replace  type type_nest_milestone as OBJECT (
      activity_id number(3),
     activity_desc varchar2(100),
     exp_start_year number(4),
.....
create type type_milestone as table of type_nest_milestone;
create or replace  type type_nest_project_proposal as OBJECT
(
    objective_type ref type_objective_mas,
    expected_outcome_nt type_outcome,
...
);
create    type    type_project_proposal    as    table    of
type_nest_project_proposal;
create table project_proposals
(
                seeker_id          number(8)          references
seeker_mas(seeker_id),
                project_id number(7),
                ....
                expected_outcome_nt type_outcome,
                ...
        tech_used_nt type_tech_used
)    nested    table    expected_outcome_nt    store    as
expected_outcome_tab
...
```

TABLE 4

FUND SEEKER AGENT CALLING OBJECT METHOD

```
getContentManager().registerLanguage(codec);
getContentManager().registerOntology(ontology);
ContentElement                    content                    =
getContentManager().extractContent(msg);
ContentElementList cel = new ContentElementList();
ContentManager cm = getContentManager();
Action action = (Action) cm.extractContent(msg);
cel.add(action);
ACLMessage reply = msg.createReply();
reply.setPerformative(ACLMessage.INFORM);
reply.setOntology(ontology.getName());
List retval = new ArrayList();
objectclasses obj = new objectclasses();
ret = obj.FillPropsal_errors(pn);
Login login = new Login();
login.setProposal_errors(ret);
……
Result result = new Result((Action)content, login);
getContentManager().fillContent(reply,result);

      send(reply);
```

TABLE 45

VALIDATING INPUT

```
public List FillPropsal_errors(proposal prop)
{               List outer_most_list = new ArrayList();
                checkData cd = new checkData();
                try
                {
                                List inner_l = new ArrayList();
                                String er;
                                er =cd.checkString(prop.getTitle(),1,30,false);
                                if (er.length() > 0)
                                                inner_l.add("Title#"+er);
                                 er =cd.checkString(p1.getName(),1,20,false);
                                ……
                                if          (inner_l.size()          >0          )
outer_most_list.add(inner_l);
```

TABLE 6

INTERFACING

```
public String savePageMain(proposal prop)
{
 String sql="";
 try
 {
  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
  conn = DriverManager.getConnection(conn,ora_user,ora_pwd);

  Select = conn.createStatement();
  sql = "select project_seq.nextval pid from dual";
  result = Select.executeQuery(sql);
  if (result.next())
  {
   pid = result.getInt("pid");
   Select = conn.createStatement();
   sql = "insert into project_proposals (";
   sql += "
SEEKER_ID,STATUS,CONTACT_PERSON,CONTACT_DESIGN,
PROJECT_OBJECTIVE ";
   sql                    +=                    "                    ,
PROJET_SUMMARY,PROJECT_START_DATE,PROJECT_END_
DATE,PROJECT_ID)";
   sql += " values (";
   sql += "'" + prop.getChiefname() + "',";
   ......
   sql += ","+ pid +")";
   Select.executeUpdate(sql);
   return pid;
  }
 ….}
 catch(Exception e)
  ...
 }
```

VI. Conclusion and Scope for Future Work

The work presented here explains the procedure to develop and implement database support to web based multi-agent resource allocation system to allocate funds and monitor the utilization. The backend support of database to agents helped them to share information with each other. Agents in JADE are managed efficiently by mapping their tasks to methods of Java class that validates inputs and process request at database level. The approach is suitable to integrate three layers of browser based system: frontend, agents and database. Future scope includes implementation of complex tasks of agent through methods of Oracle objects.

References

[1] Manish Arora and Syamala Devi, "Multi agent system for Resource Allocation and Monitoring" *African Journal of Mathematics and Computer Science Research*, Vol. 1(2), pp. 020-027, 2008.

[2] Jery L. Mumpower and Thomas A Darling, "Modeling Resource Allocation Negotiations", in *Proceeding of IEEE Twenty Fourth Annual Hawaii International Conference*, Vol. 3, pp. 641-649, 1991.

[3] Jin Cheng, Haiping Bai and Ziping Li, "Quantification of Non Quantitative Indicators of Performance Measurement, Wireless Communications, Networking and Mobile Computing", in *4th International Conference WiCOM '08*, pp. 1-5, 2008.

[4] Paul Kauffmann, P.E., Resit Unal, Abel Fernandez , P.E. and Charles Keating, "A model for Allocating resources to Research Programs by evaluating Technical Importance and Research Productivity", *Engineering Management Journal : ProQuest Science Journals*, Vol. 12, No. 1, 2000.

[5] Y. CHevaleyre, P.E. Dumme, U. Endiriss, J.Lang, et al, "Issues in Multi Agent Resources Allocation", Informatica Vol. 30, pp. 3-31, 2006.

[6] Peng Y. et al., "A Multi Agent System for Enterprise Integration", International Journal of Agile Manufacturing, Vol. 1, No. 2, pp 201-212, 1998.

[7] Mikael Berndtsson, Sharma Chakarvarathy and Brian Lings. "Extending Database for Coordination among Agents", International Journal of Cooperative Information Systems, Vol. 6, issue 3-4, pp 315-229, 1997.

[8] Wang Huibing, "Extending Object Relational Database to Support Spatio-Temporal Data", The International Archives of the Photogrammetry , Remote Sensing and Spatial Information Science Issues, Vol. XXXVII, Part 2, Being, 2008.

[9] E.Marcos, B. Vela and J. M. Cavero, "Extending UML for Object Relational Database Design", UML 2001, LNCS Springer-Verlag Berlin Heidelberg, pp. 225-239, 2001.

[10] Ming Wang, "Using Object Relational Database Technology to Support Database Development", Issues in Information System, Vol. XI, No. 1, 2010.

[11] N. Zhou, K. Xing and S. Nagalingam, "An Agent based Cross-enterprise Resource Planning for Small and Medium Enterprises", IAENG International Journal of Computer Sciences, Vol. 37, Issue 3, 2010.

[12] Al-Sammarraie Hosam, Merza Abbas, Shakeel Ahmad and Adil Mustafa, "Intelligent Agent System Architecture for Presenting Health Grid Contents From Complex Database", in IEEE International Conference on Intelligent Systems, Modelling and Simulation, pp. 38-42, 2010.

[13] Antonio J. M. Castro, Eugenio Oliveirac, "A Multi-agent System for Airline Operation Control", in 7th International Conference on PAAMS'09, AISC 55, pp. 159-168, 2009.

[14] Manish Arora and Syamala Devi, "Implementation of Web Based Multi Agent Resource Allocation System", International Journal of Information Sciences and Application,, Vol. 2,No. 2, 2010.