International Journal of Information Technology and Knowledge Management
July-December 2011, Volume 4, No. 2, pp. 519-525

DISTRIBUTED ALGORITHM FOR MULTI-AGENT ENVIRONMENT

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

Traditional algorithms are suitable when steps of algorithm are to be executed in a sequence and at a single location. These algorithms have limitations, when used in multi-agent environment, where different agents of single application run on different nodes, thus emphasizing the need for distributed algorithms. Agents in multi-agent systems perform task independently and share the result with other agents. This motivates to use distributed algorithm, where tasks of the business process are distributed among agents, so that they can perform in parallel and independently. This paper presents a distributed algorithm for multi agent systems with case study of budget allocation in web-based environment. The algorithm is based on heuristic approach. Three agents of systems perform different tasks of budget allocation procedure like information gathering, filtering, evaluating and budget allocation at different locations and timings. This approach takes advantage of distributing and parallel computing.

Keywords: Distributed Algorithm, JADE, Multi Agent System.

1. Introduction

Distributed algorithms have advantages over traditional algorithm in multi agent environment where many agents operate in environment, gather information from environment and act accordingly. Distributed algorithms are efficient algorithms as compared to traditional ones that run in a particular order. A distributed algorithm is designed and implemented for multi agent environment. The aim is to allocate budget to deserving fund seekers when funds are limited. Agents of system evaluate the proposals technically and financially and then finally allocate budget after ranking proposals.

The structure of the paper is as follows: section 2 briefs the background and related work. Section 3 details the budget allocation problem. Section 4 describes multi agent approach to solve the budget allocation problem. Section 5 defined traditional algorithm to allocate budget, while section 6 describes distributed algorithm for multi agent environment. Section 7 shows the implementation of agents and algorithm using agent development tools.

2. Background and Related Work

An Agent is an autonomous entity, which performs a given task using information gathered from its environment to act in a suitable manner to complete the task successfully. Agent must be able to change its behavior based on changes occurring in its environment. An Agent has characteristics of reactivity, autonomy, collaborative behavior, communication act, mobility, proactive, adaptability and inferential capabilities [1].

A Multi Agent Systems (MAS) consist of number of agents that interact with one another. Agents act on the behalf of users/other agents with different goals and motivation. The Agents in MAS work in a team to achieve common goal by interacting with one another [2]. Agents perform task as a part of their actions. This could be any scientific computation or business logic. The steps to be performed are represented as an algorithm and implemented using agent development tool.

An algorithm is well-defined computational procedure that takes some value as input and produces some value or set of values as output. Algorithm is thus a sequence of steps that transforms input into output [3]. Algorithm can be executed in sequential, parallel or distributed manner. Sequential algorithm or traditional algorithm performs steps in a particular sequence at a single hardware location. With the evolution of technologies in the area of multiprocessor system or multi tasking capabilities in a single processor, the focus of researchers moved from traditional approach to parallel or distributed algorithm. If traditional algorithms are executed in multiprocessor computers or networked computers, these computers would be underutilized. Moreover, majority of the business processes have some tasks that can be performed independently. Executing such algorithms over networked machines, multiprocessor machines or single processor capable of multitasking is cost ineffective in terms of CPU utilization.

Distributed algorithm is designed to run on multiple computational nodes. Nodes could be computer connected in network, a process or a thread. Node carries out a task or part of the algorithm and communicates with others by passing messages. Different nodes perform different tasks

[1]DOEACC Society, Chandigarh Centre, Chandigarh, INDIA
[2]Department of Computer Science and Application, Panjab University, Patiala, INDIA
E-mail: [1]ind_manish@yahoo.com, [2]symala@pu.ac.in

of the same algorithm at the same time. Distributed algorithms have advantage of inherent parallelism and scalability. Node, executing a part of algorithm, has limited information about what the other nodes, executing other parts of same algorithm, have information with them. A fully decentralized algorithm provides a natural path for parallelism [4]. One major challenge in designing distributed algorithm is successful coordinating behavior in the event of processor failure or communication link failure, so these algorithms must be robust enough to handle such situations without affecting the overall execution of process. Distributed algorithms are also designed to exchange information, share resource, increase reliability, and increase performance due to parallelism. These algorithms are suitable for wide range of application in the area of Telecommunication, Distributed Information Processing and Scientific Computation. The algorithm of multi-agent systems differs from that of traditional computation science due to dynamic environment. Visualization agents, their action and behavior can properly define the algorithm [5].

Traditional algorithms are found unsuitable or cost ineffective when implemented in multi-agent system due to following reasons.

(i) In multi agent system, different agents perform different tasks of the problem independently.

(ii) Agents reside on different machines/nodes.

(iii) Solution, in traditional algorithm, is found in centralized, sequential and deterministic environment while in multi-agent system, solution is obtained as result of distributed interaction of agents.

(iv) In case of multi agent system, if traditional algorithm is used, sequence of events is very important and requires mutual exclusion. In case, an agent, performing one action, requires some kind of resources, it has to broadcast request to all the agents operating in environment for the required resource. If agent receives, clear signal from every agent only then it can use the resource. If any other participating agent declines the resource, requesting agent will have to wait. Failure in connection or communication links during this process leads to delay in executing the process or algorithm.

Many nowadays applications are based on distributed computation; it may be reading email or browsing Internet etc. Some form of distributed computing is involved, ranging from simple client server computing to grid computing. In web application, server process keeps providing information to other client processes, even if, some of them fail or get disconnected. Web image retrieval using multi-agent technology uses distributed algorithm, where agents searches the images with characteristics like color and shape stored on different locations. Images are grouped together and mobile agent searches from these groups. The images are ranked according to similarities in features and are shown to client [6]. Distributed algorithms are being used in various multi agent systems like scheduling, production and risk allocations [7-10]

Based on background, it is observed that in multi agent application, business logic can be depicted as distributed algorithm and implemented accordingly. The parts of algorithms are executed by agents on different nodes. On similar lines, a distributed algorithm has been designed to allocate budget to deserving fund seekers and implemented for multi-agent budget allocation problem in web based environment.

3. Budget Allocation Problem

Budget allocation problem occurs when limited funds are to be allocated to most deserving and competent fund seekers. These funds are allocated to fund seekers to execute their projects. In India, lots of funds are allocated in various areas like education, research & development and social oriented schemes. Funds seekers submit project proposals and these proposals are filtered according to criteria set by fund allocator. The proposals are then evaluated technically and financially. Both quantifiable and non-quantifiable decision-making factors are considered to rank the projects. Weightage is given to each decision-making criterion. After ranking, budget is allocated according to availability. Quantification of non-quantifiable factors is done using heuristic approach and fuzzy system [11]. From the review of current practices and expert views, five high level decision making factors are identified; Solution Delivery & Contribution, Technical, Financial, Capacity & Expertise and Risk Management.

4. Multi-Agent Based Budget Allocation

Based on the problem mentioned above, a multi-agent system for resource (budget) allocation is designed that interacts with users (fund seekers, fund allocator and reviewer) through web-based interface. The system has back support of database to share information between agents even if, agents are off line due to some reason. Three agents have been designed based on requirements of actions, responsibilities and autonomy. The Fig. 1 shows the model of complete web based budget allocation system using multi-agent technology named as Multi Agent System for Resource Allocation and Monitoring (MASRAM).

(1) Coordinator Agent

Coordinator Agent interacts with three types of users of MASRAM i.e. Fund Seeker user, Fund Allocator user and Reviewer guser. Fund Seeker user seeks funds, Fund

Allocator user allocates funds and monitors the utilization while Review user reviews the allocation.

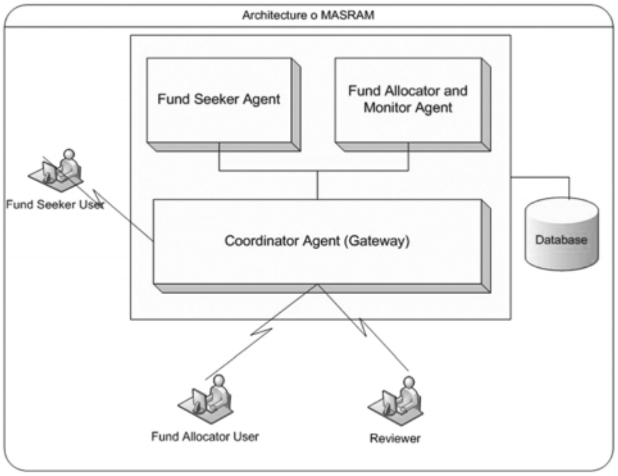

Fig. 1: Architecture of Budget Allocation

(2) Fund Seeker Agent

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

(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.

Steps involved in budget allocation are:

(i) Information Gathering Phase: In this phase, required information is gathered from user as input like complete proposal details, fund available, funds already allocated and criteria to qualify for availing funds.

(ii) Filtering Phase: In this phase, proposals are checked against the criteria set by allocator. Those proposals are considered for allocations who qualify, remaining proposals are rejected.

(iii) Evaluating Phase: Filtering phase evaluates proposals with respect to decision-making criteria. A numerical value is calculated against each criterion. Accumulated assigned values helps in ranking the projects.

(iv) Allocating Phase: Allocating phase allocates budget based on availability and rank of the proposal. The allocation could be zero or 100 percent. Three possibilities are there; one when sufficient funds are available, all the qualified proposals are given funds according to their need. In second case, where funds are more than weighted required and less than total requirement, funds are allocated according to weight. In third possibility, funds are allocated proportionally according to weighted requirement. The steps have been shown in Fig.2.

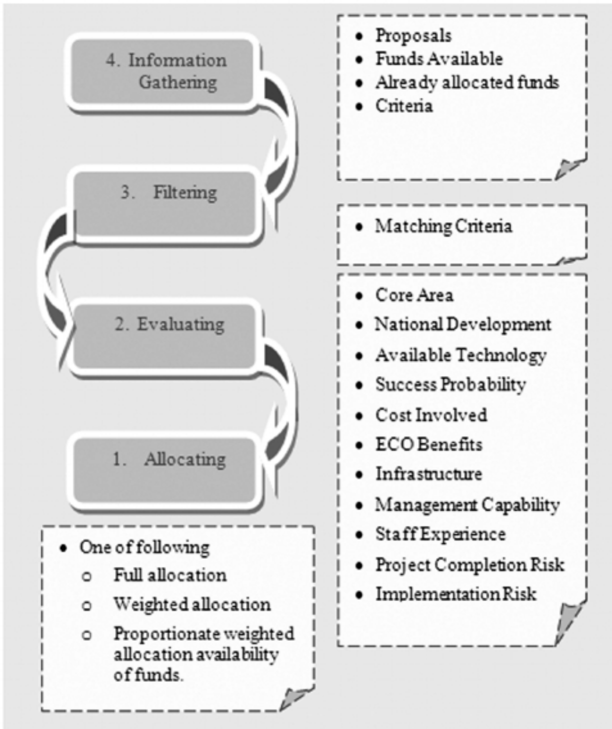

Fig. 2: Budget Allocation Phases

5. Algorithm Steps

This section describes all four phases mentioned above to allocate budget mathematically. Information gathering phase is covered in step I. Steps II and III belong to filtering phase, steps IV and V detail evaluating phase and step VI represents allocating phase as shown in Table 1. The procedures in table are detailed further.

Table 1
Traditional Algorithm

Step I	Do initialize()//Initialization
Step II	Repeat step III to step VI for p = 1 to m
Step III	Do match_criteria() //Matching Criteria
Step IV	Do evaluate() //Evaluating Proposals
Step V	Do assign_weight() //Assigning Weights to Proposals
Step VI	Do allocate() //Allocating Funds

Table 2
Initialization

Procedure initialize()

begin

Let R = {R1, R2, R3…Rm} be set of resources (funds) for a particular category of funds.

Let Ai, 1<=i<=n be n fund seekers requiring funds.

Let Xi,j be the funds requirement by ith fund seeker from jth fund category.

Let r be number of Decision Making Factors (DMF).

Let CMINi, 1<=i<=r be minimum requirement for ith criteria.

Let Ci,j be the criteria value of ith fund seeker jth criteria. 1 <=i<=n, 1<=j<=r.

Let EVALij be evaluation matrix where 1<=i<=n and 1<=j<=r.

Let W = {W1, W2, W3…Wn} be set of weight assigned to ith proposal.

Let ARi,j be fund allotted matrix to ith fund seeker from jth resource.

Let PROJECT_STATUSi stores whether proposal qualified for allocation or not.

End

Table 3
Filtering Procedure

Procedure match_criteria()

begin

For i = 1 to n

For j = 1 to r

If ($CMIN_j < C_{i,j}$ then

Project_status$_i$ = 'Rejected'

$X_{i,j} \leftarrow 0$

Else

Project_status$_i$ = 'Qualified'

End if

End for

End for

End

Table 4
Evaluating Procedure

Procedure evaluate()

Begin

For i = 1 to n

For j = 1 to r

If(PROPJECT_STATUS$_i$ = 'QUALIFIED'

do Evaluate_cri(i,j) // calls procedure to evaluate proposals

End if

End for

End for ; End

Table 5
Weight Assignment Procedure

Procedure assign_weight()

begin

For i = 1 to r

For j = 1 to n

If(PROPJECT_STATUS$_i$ = 'QUALIFIED'

$EVAL_{i,j} = K_i * EVAL_{i,j}/EVAL_{k,i}$;

End if

End for

End for

W_i = Geometric Mean ($EVAL_{i,j}$), 1<=i<=n and 1<=j<=r and PROPJECT_STATUS$_i$

= 'QUALIFIED'

End

Table 6
Final Allocation Procedure

Procedure do allocate ()

being

$$Ri, j \begin{cases} Xi, j \quad \text{if} \quad Rj \geq \sum_{p=1}^{n} Xp, j \\ Wi \ * Xi, j \quad \text{if} \quad \sum_{p=1}^{n} Wp * Xp, j \leq Rj \leq \sum_{p=1}^{n} Xp, j \\ \dfrac{Wi * Xi, j}{\sum_{p=1}^{n} Wi * Xp, j} \times Rj \quad \text{Otherwise} \end{cases}$$

end

6. DISTRIBUTED ALGORITHM TO ALLOCATE BUDGET

Agents of the system perform actions at different time and at different location. The steps mentioned in previous section cover all the actions performed by agents at different times and locations. Hence it becomes important that algorithm may be distributed one. Though agents perform task independently, yet they need to share information with other agents to accomplish overall goal of the system. Table 8 shows the distribution of tasks of algorithm among agents.

Table 7
Evaluate Procedure

Procedure evaluate_cri(i,j)

Begin

If(j = 1) then

Find total number of core keywords in ith proposal and let it be kw

$EVAL_{i,j} \leftarrow kw$

Else if (j = 2) then

Find total number candidates to be trained in i^{th} proposal and let it be kw
$EVAL_{i,j} \leftarrow$ kw
Else if (j = 3) then
Find total number national development and social impact keywords in i^{th} proposal and let it be kw
$EVAL_{i,j} \leftarrow$ kw
Else if (j = 4) then
Find total number technical used keywords in i^{th} proposal and let it be kw
$EVAL_{i,j} \leftarrow$ kw
Else if (j = 5) then
Let tpc be the total projects completed by i^{th} fund seeker
Let tpd be the total projects delayed by i^{th} fund seeker
Let tp be the total projects handled by i^{th} fund seeker
prob_tpc = tpc/tp
prob_tpd = tpd/tp
prob_either = (prob_tpc + prob_tpd)-(prob_tpc * prob_tpd)
$EVAL_{i,j} \leftarrow$ prob_either
Else if (j = 6) then
Let tb be total budget proposed by i^{th} fund seeker
Let tr be total budget required by i^{th} fund seeker
$EVAL_{i,j} \leftarrow$ tr/tb
Else if (j = 7) then
$EVAL_{i,j} \leftarrow$ 1 // time being
Else if (j = 8) then
Let s be sum of infrastructure available (infrastructure ID wise) for all the qualified fund seeker
$EVAL_{i,j} \leftarrow$ infrastructure available for i^{th} fund seeker
Else if (j = 9) then
Find total number management capability keywords in i^{th} proposal and let it be kw
$EVAL_{i,j} \leftarrow$ kw
Else if (j = 10) then
Let s be sum of manpower available (Designation wise) for all the qualified fund seeker
staff(i) $\leftarrow$ manpower available for i^{th} fund seeker
Else if (j = 11) then
$EVAL_{i,j} \leftarrow 1- EVAL_{i,6}$
Else if (j = 12) then
$EVAL_{i,j} \leftarrow$
End if
End

Table 8
Tasks Allocation

Agent	Steps of Algorithm
Fund Seeker	Step I, II and III
Fund Allocator and Monitor	Step IV, V and VI
Coordinator Agent	To interact with user through web based interface

Agents can also work in parallel, e.g. Fund Seeker agent can be busy in gathering information from Fund Seeker FS1 while Fund Allocator and Monitor Agent is may be in evaluating and allocating budget to FS2 and FS3 fund seekers. Fund Seeker Agent passes the information to Fund Allocator and Monitor Agent through database support and well-defined ontology [12]. Distributed Algorithm completes the task when all participating agents complete their tasks. Step IV of the algorithm is further divided into multiple tasks enabling us to use multi threading. There are 12 different decision making factors categorized into five high level factors as discussed in previous section. These are independent except 5.1 as is it dependent on 2.2 (Table 13). The computations of these factors are multithreaded taking advantage of both parallel and distributed environment. The distributed algorithm is described in table 9.

Table 9
Distributed Algorithm

// Distributed Algorithm
Begin
Thread FS_thread $\leftarrow$ Requests agent 'Fund Seeker Agent' to provide fund
requirement from fund seeker user
Thread FA_thread $\leftarrow$ Requests agent 'Fund Allocator and Monitor Agent' to
provide takes information from fund seeker agent
Do while fund requirement found and budget found
Begin //assign task to Fund Allocator and Monitor Agent
Do initialize()
For I = 1 to m // allocating categories
Create thread t_i = budget() // calls budget procedure
End for
Thread_wait //Wait for all the active threads to finish
End End

Table 10
Distributed Algorithm (Allocation Procedure)

//budget procedure
begin
do match_criteria()
do evaluate_da(i)
do assign_weight()
do allocate_funds()
end
procedure evaluate_da(i)
begin
for j = 1 to r//no. of decision making factors
create thread t_{ij} = evaluate() // calls evaluate procedure
end for
thread_wait // waits for all the active threads to finish
End

7. IMPLEMENTATION

The above mentioned algorithm is implemented in web based multi agent system for resource allocation and monitoring. All the three agents reside on server side. Two servers are used for agent-based application: one as database server and second as agent server. Agent Server hosts all three agents. Three types of users interact with agents from client machine through Graphical User Interface (GUI). Interface programs to interact with the users are implemented using JSP (Java Server Pages). Agents are implemented using JADE, Java Agent Development Framework, a FIPA (Foundation of Intelligent Physical Agents) compliant framework to develop agents, and oracle is used as backend database. The interaction between agents and database is through Java business classes [13].

Information gathering task has been distributed between two agents; Fund Seeker Agent and Fund Allocator & Monitor Agent. Fund Seeker Agent seeks information from fund seeker user while Fund Allocator & Monitor Agent seeks information from Fund allocator user and reviewer user. Both store the information in database. Fund Allocator & Monitor Agent from time to time senses the requests to allocate funds and does the remaining phases of filtering, evaluating and allocating. To evaluate proposals, separate eleven threads are started one each for eleven independent decision-making factors. This is takes advantage of parallelism.

Table 11 shows the Budget requirement and budget allocation of three projects in two slots. At first time, Fund Allocator & Monitor Agent finds only one proposal and during next slot, it finds two proposals. The proposals are ranked after evaluation and assigning weights. Project ID 81 is allotted full amount of budget required as it was available, but Project ID 48 and 61 are allocated proportionately according to weight and rank since limited funds are available as shown in Table 12 along with criteria and funds available.

Table 11
Budget Requirement and Allocation

Project Id	Category	Allocato r ID	Seeker ID	Budget Required	Weightag e	Allocated Budget	Slot No.	Rank
48	2	2002	1	720000	53.842	646104	2	1
61	1	2002	1	810000	46.158	553896	2	2
81	2	2001	2	300000	100	300000	1	1

Table 12
Minimum Criteria

Allocation ID	Category	Criteria	Min	Available
2001	1- For R & D	Min Experience of Seeker	3	800000
		Already Done similar projects	1	
2002	2- For quality Education	Min Staff Member	5	1200000
		Min Experience of Seeker	10	
		Already Done similar projects	2	

Table 13 shows the decision-making factors and their weightage in allocation of budget. Table 14 shows the result of evaluation of proposals against these decision-making factors. First proposals are evaluated against these factors and then weightage is given according to decision-making weights.

Table 13
Weightage of Decision Making Factors

Decision Making Id	Decision Making Factor	Decision Making Weight
1.1	Core Area	.038
1.2	HR Development	.076
1.3	National Development and Social Impact	.106
2.1	Available Technology	043
2.2	Success Probability	.103
3.1	Cost Involved	.122
3.2	ECO Benefit	.063
4.1	Infrastructure	.027
4.2	Management Capability	.133
4.3	Staff Experience	.098
5.1	Project Completion Risk	.13
5.2	Implementation Risk	.06

Table 14
Calculated Weights

Project Id - 48		Project Id - 61		Project Id - 81	
Decision Making ID	Evaluated Value	Decision Making ID	Evaluated Value	Decision Making ID	Making Value
1.1	0.5	1.1	0.5	1.1	1
1.2	0.5	1.2	0.5	1.2	1
1.3	0.5	1.3	0.5	1.3	1
2.1	0.5	2.1	0.5	2.1	1
2.2	0.455	2.2	0.545	2.2	1
3.1	0.471	3.1	0.529	3.1	1
3.2	0.5	3.2	0.5	3.2	1
4.1	0.605	4.1	0.395	4.1	1
4.2	0.5	4.2	0.5	4.2	0
4.3	0.723	4.3	0.277	4.3	1
5.1	0.6	5.1	0.4	5.1	0
5.2	0.635	5.2	0.365	5.2	0

8. CONCLUSION

In multi-agent systems, agents perform different tasks at different locations and time to achieve their individual goals and overall goal of the system. Traditional algorithms to allocate budget were found unsuitable due to their characteristics of orderly execution and that too at a single location in multi agent environment. The distributed algorithm to allocate budget in multi-agent environment removes the limitation of traditional algorithms. Parts of the budget allocation problem were distributed among two agents. During allocation procedure, further multithreading is used. This distributed algorithm to allocate budget found fit for multi agent environment. This procedure takes advantage of parallel and distributing computation.

REFERENCES

[1] Association of Advancement of Artificial Intelligence, Available at http://www.aaai.org/AITopics/html/multi.html, accessed on April 10, 2007.

[2] Jarg Denzinger, "Multi Agent Systems", Department of Computer Science, University of Calgary, Canada, Available at http://pages.cpsc.ucalgary.ca/~denzinge/courses/567-winter2006/slides/04-masdef-handout.pdf, accessed on April 11, 2007.

[3] Thomas H. Cormen et al., "Introduction to Algorithm", Second edition, MIT Press, Cambridge, Massachusetts London, England, 2001.

[4] Raz Nissim, et al., "A General, Fully Distributed Multi-Agent Planning Algorithm",9th International Conference on Autonomous Agents and Multi Agent Systems, AAMAS, Toronto, pp 1323-1330, 2010.

[5] Jose M. Vidal et al., "The Past and Future of Multi Agent System", AAMAS Workshop on Teaching Multi-Agent Systems, 2004.

[6] Alaa M. Riad, et al., "An Intelligent Distributed Algorithm for Efficient Web Images Retrieval", International Journal of Communication, pp 63-76, 2009.

[7] Ron Xie, et al, "Scheduling Multi-Task Multi Agent Systems", Proceeding on 5th International Conference on Autonomous Agents, ACM New York, 2001

[8] Masahiro Ono, et al, "Market Based Risk Allocation for Multi Agent System", Proceeding on 9th International Conference on Autonomous Agents and Multi Agent Systems, 1, AAMAS '10, 2010.

[9] Toru Ishida, "Parallel, Distributed and Multi-Agent Production Systems -A Research Foundation for Distributed Artificial Intelligence", Proceedings of the First International Conference on Multi Agent Systems, pp 416-422, 1995.

[10] Sebastien Paquet et al., "Multi-agent Systems Viewed as Distributed Scheduling Systems: Methodology and Experiments", Advances in Artificial Intelligence, Lecture Notes in Computer Science (Springerlink), vol 3501, 2005.

[11] Jin Cheng, Haiping Bai and Ziping Li, "Quantification of Non Quantitative Indicators of Performance Measurement", Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference, page(s): 1-5, 2008.

[12] Manish Arora, M. Syamala Devi, "Ontology Based Agent Communication in Resource Allocation and Monitoring", IJCSI International Journal of Computer Science Issues, 7, Issue 6, November 2010, ISSN (Online): 1694-0814

[13] Manish Arora, M. Syamala Devi, "Role of Database in Resource Allocation Problem", International Journal of Computer Science and Information Technology, 2(2) ,pp 668-672, 2011.