INTRODUCTION OF KNOWLEDGE MANAGEMENT ARCHITECTURE USING MULTI AGENT

Sheikh Amanur Rahman[1], Anamika Bhardwaj[2], Mukul Pathak[3] and Shobhit Singh Rathore[4]

[1, 2, 3]Department of Computer Science & Engineering, Galgotias College of Engineering & Technology, Greater Noida (U.P.), India
[1]sheikhamanur@gmail.com
[2]anamikabhardwaj@gmail.com
[3]mukul.pathak2501@gmail.com
[4]SDG Software India Pvt. Ltd., Noida (U. P.), India
shobhit.rathore@gmail.com

ABSTRACT

In current Public Sector environment, handling and managing of fast growing knowledge is very difficult which results in degrading the overall productivity. Therefore decision for implementing Knowledge Management (KM) using Multi Agents results in reducing the time overhead for serving relevant knowledge to end-user. The aim of this paper is to propose KM architecture for multi agents that will be helpful in disseminating knowledge to public sectors in a better way and thus enhancing the productivity. The paper first reviews the literature on various KM frameworks and tools for implementing multi-agents and then proposes a multi agent enterprise KM (MAEKM) architecture which describes how knowledge dissemination is done which results in fast processing and therefore enhancing the service qualities of the organization. This paper then implements MAEKM architecture using JADE Framework for public sector. The paper describes the need for implementing this architecture and provides several benefits after implementing it.

KEYWORDS

Public Sector Unit (PSU), Knowledge Management (KM), Multi Agent, MAEKM, JADE (Java Agent Development)

1. INTRODUCTION

Knowledge Management (KM) [1, 12] can be described as to give relevant information to the right people at right time and at right place. It mainly concern using, spreading, sharing, representing and storing of knowledge. Knowledge is categorized into three types, explicit, tacit, embedded [14].explicit knowledge involves people accessing information what they need; only the knowledge which is relevant and important is stored. Time to time it is reviewed, updated, or discarded the knowledge. Tacit knowledge is embodied knowledge that is context dependent and personal in nature and hard to define and largely experience based. It includes cultural belief, values, attitudes, mental models. Knowledge locked in process, products and structures is embedded knowledge. It's difficult to understand and change.

Multi agent technology can increase efficiency of knowledge management and provide intelligent services in public sector unit. Multi agent is combination of agents and its environment. Agent have several characteristics Partially Autonomous, Local views, Decentralization .Multi agent system can find solution of their problem them self .main feature of multi agent system can easily modified and reconstructed. JADE (Java Agent Development Framework) is software framework and allows multiple FIPA-compliant agent and using SL

DOI : 10.5121/ijcsit.2011.3618

and XML as communication language. JADE create multiple container for agents. Set of container is platform. Platform must have main container (combination of 2 agents AMS and DF agent). AMS (Agent Management System) can create and kill containers. DF (Directory Facilitator) used to advertise services of agent.

The purposes of this paper are to (a)Literature survey on various knowledge management framework and tools for implementing Multi Agents (b)Introduce MA enterprise knowledge management architecture (MAEKM) involved knowledge distribution in public sector, (c) Partially implemented the Multi Agent enterprise knowledge management (MAEKM) architecture using JADE with Eclipse for public sector unit.

1.1 KM life cycle

Nissen [2] proposed KM lifecycle model comprises of six phases. They are as follows: Create Formalize, Organize, Distribute, Apply and Evolve. The phases of life cycle are shown in Figure l. The knowledge management life cycle starts with the create phase. After the creation of new knowledge, it is formalized and stored in knowledge base, depending upon the knowledge organizing mechanism. When any knowledge is required, it is searched in knowledge base and can be access if the relevant knowledge is there. Finally knowledge is applied and further evolved into new knowledge. It also leads to further knowledge creation and completing the lifecycle.

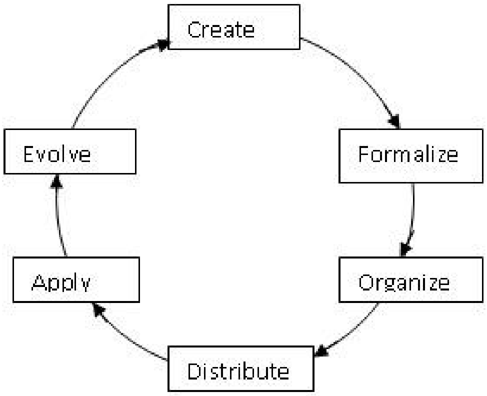


Figure.1 Knowledge Management life cycle

1.2 Knowledge Management in Industries

Engineering can be termed as a knowledge base industry. Even low profile work requires knowledge, thought, ideas and skills from different sources such as documents, electronic media and people. There are several engineering organizations which have been managing and organizing their knowledge manually for years. But nowadays, the industry is facing problem in knowledge management, so most of the organizations required a well structured, formal and consistent approach to knowledge management [11]. Industries projects are usually organized into stages with deadlines and there are several teams assigned to those stages. The problem is that extracting and reusing of knowledge learning from projects is measured to be difficult because teams are of dynamic in nature and may be dismissed before finishing point of the project and therefore gathered to the next project. Such types of problems generally bound the information flow and can create barrier to knowledge learning.

The remaining paper is organized as follows. Section 2 gives the literature review and current techniques and research related to implementing multi-agent KMS. This section also gives the idea about the various tools to implement multi-agents and then shows the features of JADE when compared with others. Section 3 discusses the implementation of new multi-agent enterprise knowledge management (EKM) architecture for Public Sector Unit and its specialty. It also discusses the objective of implementing the proposed architecture. In Section 4, the paper has given the implementation details of the proposed architecture. Section 5 has shown the result and observations after implementing the proposed architecture. Finally, section 6 summarizes the conclusion and future work anticipated in this direction.

2. RELATED WORK

2.1 KM Frameworks and Current Techniques Implementing KMS

A lot of researchers have proposed various KM frameworks where most of the frameworks are prescriptive in nature. A few of the well known frameworks are:

2.1.1 Leonard-Barton,1995 Framework[15]

- The framework solves the problem of creative and shared problems.
- Technological knowledge transferring and absorbing from the outside of organization.
- Integrating new methodologies.

2.1.2 Nonaka and Takeouchi (1995) Framework[15]

- Proposed on the basis of knowledge conversion(tacit-explicit)
- Socialization (tacit knowledge to tacit knowledge conversion)
- Internalization (explicit knowledge to tacit knowledge conversion)
- Combination (explicit knowledge to explicit knowledge conversion)
- Externalization (tacit knowledge to explicit knowledge conversion)

Different techniques have been used to implement KMS. Intelligent agents are one of them [8]. Agents are proactive in nature as they can take the initiative at their own and complete their own goals. The autonomous behavior of the agents meets the required goal of this research because it can minimize the amount of work done by the employees when using a KM system. Another important issue is that agents can learn from their own experience. Consequently, agent systems are required to be more efficient because the agents get experience from their previous knowledge.

A lot of research and commercial organizations are involved in the realization of agent applications and a considerable number of agent construction tools have been realized [4]. Some of the most interesting are Cougaar, JACK, JADE.

2.1.3. Features of Cougaar

- Cougaar framework is based on Java language which is used for distributed applications of agents.
- Multi-tier interaction model is used in Cougaar.
- Components interacts within agents with the help of local publish-subscribe mechanism
- Message passing mechanism is used for communication between the agents
- For loading the components dynamically, Cougaar uses a flexible component model.
- Agent relationship is dynamically negotiated by the method of hierarchical service discovery
- Cougaar also enables the agents to be organized into communities

2.1.4. Features of JACK [10]

- JACK is a commercialized framework which is based on BDI agent realization [15].

- While dealing with agent plans and beliefs, JACK extends the Java by using syntactic constructs.
- JACK does not use language which is based on logic.
- Modularization is done well here through agent planning and agent team concept.

There are some important identifying features of JADE when comparing with other tools:

- JADE is identified with FIPA specification by its own because JADE is totally FIPA specification based.
- JADE provides special functionalities to make simple when developing multi-agent systems. There are very less restrictions on the user code. There is no need to understand any BDI architecture or use it by the user. Users just have to write Java code without required to learn any new special construct.
- JEE, JSE, and JME devices can be deployed by the JADE and therefore provides a homogeneous set of APIs in its runtime environment which does not depend on the underlying network and Java technology.

2.2. Jade Framework

JADE (Java Agent Development Environment)[1] is a software framework which is used for interoperable intelligent multi-agent systems by making the agent applications development easily in compliance with the FIPA specifications. List of features which are provided by the JADE to the agent programmer are as follows:

- JADE is an Agent Platform based on FIPA-compliance. It consists of AMS (Agent Management System), the default DF (Directory Facilitator), and the ACC (Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up.[1]
- Several DFs (Directory Facilitator) can be started at run time in order to build multi-agent environments. Common facilitator advertised their services.
- Java API to send/receive messages to/from other agents; ACL messages are represented as ordinary Java objects.
- Lightweight transport of ACL messages inside the same agent platform, as messages are transferred encoded as Java objects, rather than strings, in order to avoid marshalling and unmarshalling procedures.
- Support for agent mobility within a JADE agent platform.
- GUI is used to manage several agents and agent platforms from the same agent. The activity of each platform can be monitored and logged.

3. MULTI-AGENT SYSTEM

An agent can be defined as a software and/or hardware component of system who accomplishes tasks on behalf of its user [3-6]. Agents are reactive, autonomous and co-operative in nature. They have the ability of knowledge based reasoning. Agent technology is the suitable technology for designing and implementing distributed system for KM.

Multi-agent system (MAS) is now used widely while dealing with the problems occurred in complex application environments, especially in distributed environments. MAS can be defined as a group of agents that define their aim and respective actions, and it combine these functions to accomplish large and complex task such as workflow control, knowledge search etc. Every agent can serve the users or other agent through inter -communication for required problem.

Using Multi-Agent System for implementing KM provides competitive advantage and various benefits like:

(1) It minimizes time overhead for knowledge serving using inter agent communication.
(2) Work can be done parallel by assigning different tasks to different agents.
(3) Robustness is achieved.

(4) Fault-tolerance can also be achieved by one or more agents if controlling and responsibility is sufficiently distributed among different agents.
(5) Scalability can also be achieved by adding more agents to a Multi-Agent System.
(6) Problem solving capabilities can also be enhanced.

4. PROPOSED WORK

4.1 Proposed MAEKM Architecture

The proposed multi agent enterprise knowledge management (MAEKM) architecture is based on serving knowledge with the help of multi-Agent System. The salient feature of this architecture is its simplicity in nature. This new architecture minimizes the time overhead required for knowledge serving. It provides smooth communication between agents which will guarantee to produce the desired result. This new architecture will help the users when knowledge is not directly accessible by the users and therefore agents will help to serve the knowledge. In this architecture, agent communicates to each other and produces the result and serves the knowledge as fast as possible which results in enhancing the quality of services. In brief, our architecture's uniqueness is given below:-

- Simplicity in nature
- Reducing time overhead for knowledge serving
- Smooth communication between agents
- Reusability of the knowledge by agents

The proposed MAEKM architecture is shown in figure 2.

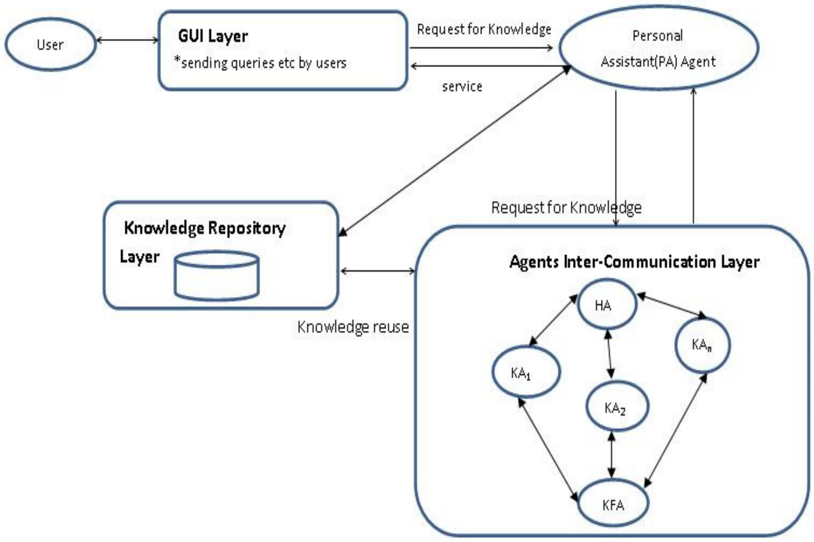

User
GUI Layer
*sending queries etc by users
Request for Knowledge
service
Personal Assistant(PA) Agent
Request for Knowledge
Knowledge Repository Layer
Knowledge reuse
Agents Inter-Communication Layer
HA
KA$_n$
KA$_1$
KA$_2$
KFA

Figure 2: Proposed general MAEKM architecture

The architecture consists of 3 layers:

1. User GUI Layer: This is the top most layer where information is send and receive by the user. At this layer, user interacts with the system to retrieve, create, share, use the knowledge etc. Personal Assistant (PA) Agent serves to user and act as a interface between user and the system.

2. Agent Inter Communication Layer: This is the main layer of the architecture. At this layer, Multi-Agents remains active all the time. Agents communicate to each other regarding knowledge processing, retrieval, creation etc and produce the result in lesser time.

3. **Knowledge Repository Layer**: This is the lower most layer. It contains organization's overall knowledge where all the information is stored.

4.1.1. Multi-Agents Descriptions

- **Head Agent (HA):** It handles and manages the all other agents which take part in some KM activity.
- **Personal Assistant (PA) Agent:** This agent serves to the users and therefore handles the user's queries regarding the knowledge processing. PA Agent collaborates with the other agents and act according to the user's queries.
- **Knowledge Agent (KA):** This agent processes the required query and produces the relevant information for the given request. It also manages and updates the Knowledge Repository.
- **Knowledge Filter Agent (KFA):** This agent filters the irrelevant data from the useful data which results in producing the accurate knowledge for the related request query.

4.2. Motive for Implementing MAEKM for Public Sector

Traditionally, in generalized Public Sectors, it is very difficult to handle and manage the organization's knowledge which results in:-

- More time consuming for knowledge serving
- Lack of appropriate knowledge
- Degradation of overall quality
- Lack of effective communication when dealing with customer knowledge queries

Therefore it is decided to implement KM for Public Sector using Multi Agent System in a way such that to get rid of all the problems above and to produce the following result after implementation:

- Smooth communication between agents
- Knowledge reusability by agent's past experience
- Reducing time overhead for knowledge serving
- Serve relevant knowledge at correct time
- Provides best possible solutions

It is decided to take the concept of Multi Agent System instead of Single Agent System because:-

- Single Agent has partial capabilities to solve a problem.
- Single Agent does the computation asynchronously.

4.3. Overall Implementation of PSU Knowledge Management

The implementation of PSU Knowledge management is shown in figure 3.

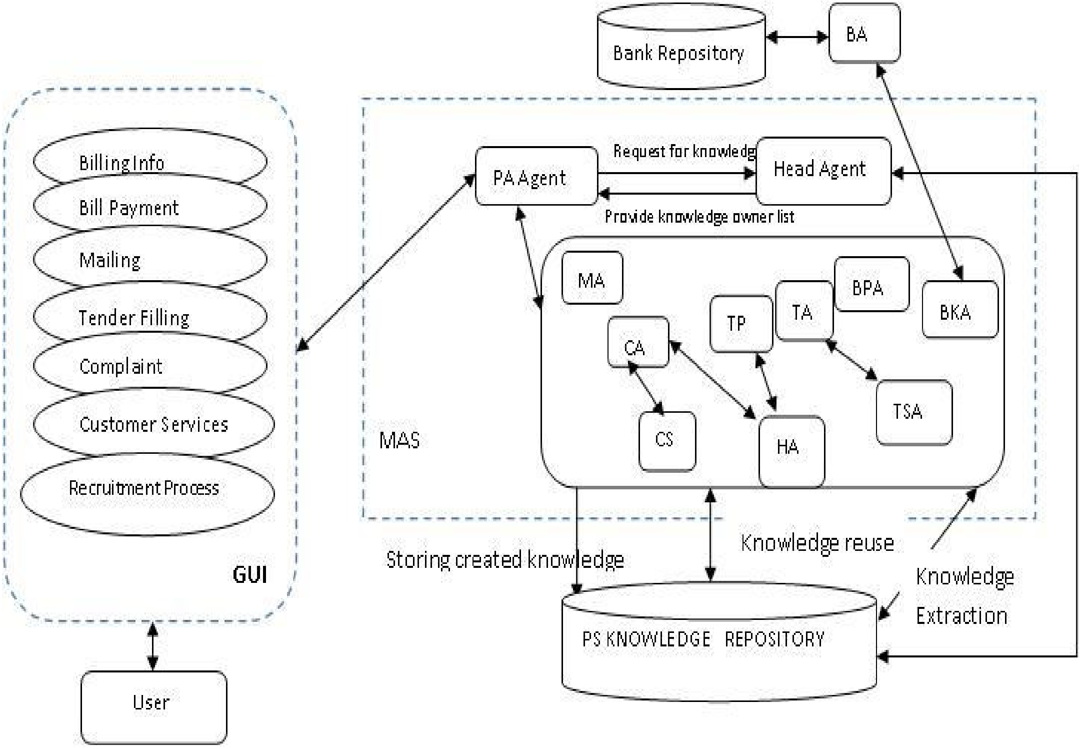


Figure 3: PSU multi agent knowledge management

Based on MAEKM architecture, implementation of PSU Knowledge Management using Multi Agent System to exchange knowledge with each other, in a way that preserves the knowledge, and therefore reaches to the knowledge seeker in a just in time. One of the main tasks of this implementation is to support and encourage collaboration and knowledge sharing. This implementation defines various services like billing information, payment services, complaint services, tender filling services, complaint services, customer support services etc.

The following scenario illustrates the functionality of the knowledge market: A customer wants to apply for a new connection of electricity. Customer requires an analysis regarding the new connection schemes which are best suited according to his need. That is not directly available in the knowledge repository. Via the user interface, he communicates his needs and conditions to his PA agent. The conditions set by the customer like type of connection etc. His assistant will serve the request and contacts the HA in order to find out required agent. HA may use its own internal information about knowledge owners in the system, or possibly, referring to the knowledge repository to find out required agents matching the request and thus provide a list of required agents to the PA agent. Following its own strategy and the preferences specified by the customer, his PA agent will then contact the required agent and try to get the best deal for his request.

4.3.1. Abbreviations Used

➢ **BA:**	Bank Agent
➢ **BKA:**	Bill Knowledge Agent
➢ **BPA:**	Bill Payment Agent
➢ **TA:**	Tender Agent
➢ **HA:**	HRD Agent
➢ **MA:**	Mailing Agent
➢ **CSA:**	Customer Services Agent

- **CA:** Complaint Agent
- **TSA:** Tender Sanction Agent
- **TPA:** Training and Placement Cell
- **PA:** Personal Assistant

4.3.2. Inter- Agent Communication

In figure 4, interaction between Personal Assistant Agent, Head Agent and Customer Service Agent is depicted. This is actually a agent interaction model diagram. These types of diagrams are very useful to see, at first glance, as agents interact with each other.

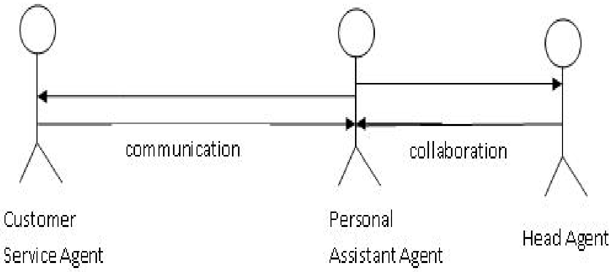


Figure 4: Interaction between agents

The above figure can be described as follows:
A customer wants to buy a new electricity connection. Then according to the figure:-Firstly PA Agent interacts with HA for serving query for new connection on behalf of its user. HA then reply to PA agent with the list of Agents serving for the related query. PA Agent then interacts with CSA and request for the related query. Finally CSA serves the required knowledge to the PA Agent.

4.3.3. Knowledge Creation

Figure 5 shows how creation of the knowledge is done in our implementation.

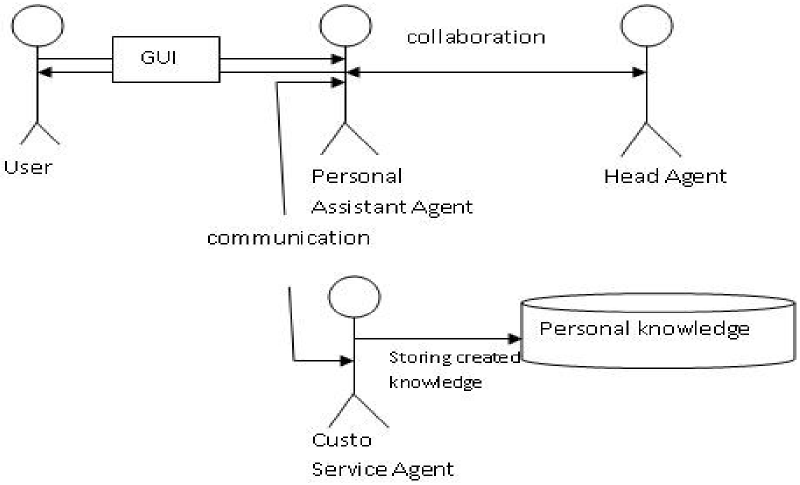


Figure 5: Knowledge creation

The above figure can be described as follows: Knowledge is created by the CSA by monitoring that which connection plan is preferred by most of the customers. Therefore by monitoring the user's activities about to purchase a new connection, CSA stores the knowledge of most preferred connection plan in its personal database.

4.3.4. Knowledge Re-Use

Figure 6 shows how knowledge is reused in the implementation.

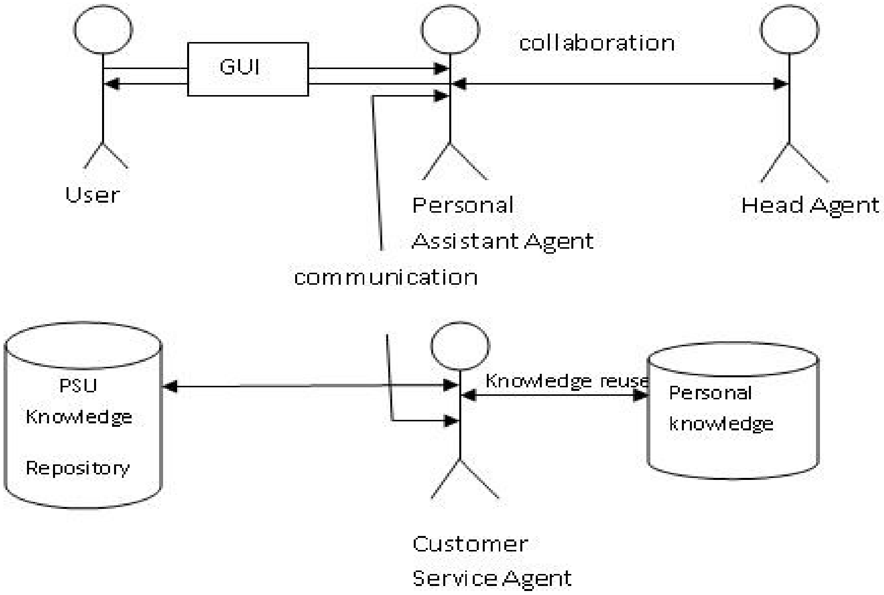


Figure 6: Knowledge re-use

The above figure can be described as follows:-After successful creation of knowledge about connection plan, if another customer request for the similar query about connection plan then CSA will re-use its knowledge and serve the required knowledge to the PA agent.

5. RESULTS & OBSERVATIONS

This paper has partially implemented the proposed MAEKM Architecture in JADE Framework successfully. After implementation it is observed that:-

- This architecture is creating, reusing, managing etc the knowledge in a good manner
- This architecture is providing smooth communication between agents
- Knowledge is serving to the end users in a lesser time
- Relevant Knowledge is serving to the users in correct time.
- Providing best possible solutions according to the related queries.

6. CONCLUSION & FUTURE WORK

This paper gives the proper introduction, working and partially implemented the Multi Agent enterprise knowledge management (MAEKM) architecture using JADE with Eclipse for public sector unit. After successfully implementation it will be able to provide smooth communication between agents, Knowledge reusability will be done by using agent's past experience, reducing time overhead for knowledge serving, able to serve relevant knowledge at correct time, able to provide best possible solutions etc which subsequently enhances the system productivity and quality of services. Implementation of EKM using Multi Agent results in more reliability, fast processing etc.

The future work on this paper will incorporate complete implementation and enhancement of the architecture to fulfill the primary goal of organizational productivity enhancement.

REFERENCES

[1]. Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, “Developing Multi agent System with JADE”, Wiley Series, England, 2007.

[2]. M. E. Nissen, M. N. Kamel, K. C. Sengupta, Toward integrating knowledge management, processes and systems: a position paper, in: Proceedings of the AAAI Symposium on Bringiwng Knowledge to Business Processes, Stanford, CA, 2008.

[3]. P. Maes, "Agents that reduce work and information overload," Communications of the ACM, vol. 37, pp. 31-40, 1994.

[4]. Reticular Systems, Agent Construction Tools. 1999. http://www.agentbuilder.com. Accessed Oct 3, 2010.

[5]. Reticular Systems, AgentBuilder - An integrated Toolkit for Constructing Intelligence Software Agents, 1999, http://www.agentbuilder.com. Accessed Oct 3, 2010.

[6]. C. Castelfranchi and Y. Lespérance, editors, Intelligent Agents VII. “Agent Theories, Architectures, and Languages” - 7th International Workshop, ATAL-2000, Boston, MA, USA, July 7-9, 2000, Proceedings, Lecture Notes in 1Artificial Intelligence. Springer- Verlag, Berlin, 2001.

[7]. T. Kawamura, N. Yoshioka, T. Hasegawa, A. Ohsuga and S. Honiden. Bee-gent “Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems”. Proceedings of the 6th Asia-Pacific Software Engneering Conference, 1999.

[8]. L. van Elst, V. Dignum, and A. Abecker, "Agent- Mediated Knowledge Management." In proceedings of the Agent-Mediated Knowledge Management, Standford (USA), pp. 1-30, 2003.

[9]. A. Helsinger, M. Thome, T. Wright, Cougaar “a scalable, distributed multi-agent architecture” in Proceedings of the IEEE Systems, Man and Cybernetics Conference, Hague, The Netherlands, 2004, pp. 1910–1917.

[10]. M. Winikoff, JACK_ intelligent agents: an industrial strength platform, in: R.H. Bordini, M. Dastani, J. Dix, A. El Fallah Seghrouchni (Eds.), Multi-Agent Programming: Languages, Platforms and Applications, Springer, 2005, pp. 175–193.

[11]. Constructing Excellence (2004), “Demystifying Knowledge Management a Best Practice Guide for the Construction Industry”; accessed July 17th 2004: www.constructingexcellence.org.uk/resourcecentre/publications/document.jsp?documentID=116179

[12]. Kephart, J. and Chess, D., 2003. The Vision of Autonomic Computing. IEEE Computer, Vol. 36, No. 1, pp. 41-50.

[13]. A.S. Rao, M. Georgeff, “BDI agents: from theory to practice” in Proceedings of the 1st International Conference on Multi-Agent Systems, San Francisco, CA, 1995, pp. 312–319.

[14]. A.D. Marwick, “Knowledge management technology,” IBM Systems Journal, 2001

[15]. Leonard-Barton, D. (1995). The Wellsprings of Knowledge. Cambridge, MA: Harvard Business School Press

[16]. Nonaka, I., and Takeouchi, H. (1995), The knowledge-creating company. NY: Oxford University Press.

Authors

Sheikh Amanur Rahman

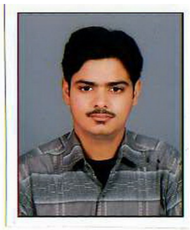
The author is pursuing Post Graduation in engineering from Galgotias College of Engineering & Technology, Greater Noida (U.P.). He has completed engineering from Vishveshwarya Institute of Engineering & Technology; Ghaziabad affiliated to Uttar Pradesh Technical University in 2008.

Anamika Bhardwaj

The author is pursuing Post Graduation in engineering from Galgotias College of Engineering & Technology. She has completed engineering from Apeejay College of Engineering; Sohna (Haryana) affiliated to Maharishi Dayanand University in 2007.

Mukul Pathak

The author is pursuing Post Graduation in engineering from Galgotias College of Engineering & Technology, Greater Noida (U.P.). He has completed engineering from College of Engineering & Technology; Moradabad(U. P.) affiliated to Gautam Buddh Technical University in 2010.



Shobhit Singh Rathore

The author is currently working in a well known IT company. He has completed engineering from Invertis Institute of Engineering & Technology; Bareilly (U. P.) affiliated to Uttar Pradesh Technical University in 2009.