Architectural Review On Multi Agent Knowledge Management

Dhanashree R. Kamble

ABSTRACT: The importance of knowledge management is growing day by day. In current environment, handling and managing of fast growing knowledge is very difficult which results in consuming more time for knowledge serving and therefore degrading the overall productivity. Therefore decision of choosing Multi-Agent system for implementing enterprise Knowledge Management (KM) results in reducing the time overhead for serving the relevant knowledge to the end-user by automatic communication between the agents. The main aim of this paper is to propose knowledge management system architecture for multi agents that will be helpful in disseminating knowledge to public sector organizations in a better way and thus enhancing the productivity. The knowledge management system architecture described in this paper also provides scalability, reusability and supports system adaptability. The paper first reviews the literature on various KM frameworks and the various tools for implementing multi-agents and also proposes Multi Agent Enterprise KM System (MAEKMS) architecture. Using this architecture, the paper describes how the agents communicate with each other automatically for knowledge dissemination which results in fast processing and therefore enhancing the service qualities of the organization. This paper then implements the Multi Agent Enterprise Knowledge Management System (MAEKMS) architecture using JADE with Eclipse for public sector unit for the modules which shows knowledge reusability, knowledge retrieving and agent automatic inter communication of the overall system.

Keywords: Knowledge Management (KM), Multi-Agent System, MAEKMS, Enterprise, Public Sector Unit (PSU), JADE

1. Introduction

Knowledge Management (KM) [1, 12] is defined as to provide relevant information to the right people at right time Knowledge is classified into two types, tacit and explicit [14]. According to A.D Marwick [14], tacit knowledge contains the knowledge stored inside the human brain which comes through experience, learning etc. This knowledge is very difficult to maintain and it is unstructured. Explicit knowledge mainly deals with the knowledge stored in a mechanical or technological way that has been extracted or codified from human brain in many forms like handbooks or information systems, databases, internal newsletter, documentation [14]. There are a few models describing the relation of tacit and explicit knowledge. One of them is described by Nonaka and Takeuchi (1995). Nonaka and Takeuchi (1995)[15] proposed a conversion model for those two knowledge types to provide the framework for KM development, and this model is illustrated in Fig.1 We are using Multi-Agent methodology using JADE (Java Agent Development) for implementing KM. An agent can be defined as a software and/or hardware component of system who accomplishes tasks on behalf of its user. Agents are reactive, autonomous and co-operative in nature. They have the ability of knowledge based reasoning. Based on the model represented below, the conversion from tacit to tacit knowledge is known as socialization. It creates the knowledge by the interaction of two or more human beings and by sharing their experiences. On the other hand, the conversion from explicit to explicit knowledge is known combination mode. Combination mode creates new explicit knowledge from multiple sources existing explicit knowledge and by some external sources to bring new explicit knowledge. Externalization is the conversion from tacit to explicit, it is the process of tacit knowledge created through capturing it by humans, either by writing it down or capturing it on computer, it is required to translate this knowledge into comprehensible forms that can be understood by others. The conversion from explicit to tacit is called Internalization, which is the process of understanding and absorbing explicit knowledge into tacit knowledge, by the individual.

	To: **Tacit**	**Explicit**	
Tacit	*Socialisation*	*Externalisation*	**From**
Explicit	*Internalization*	*Combination*	

Fig. 1 Nonaka and Takeuchi Conversion Model

1.1. KM Life Cycle:-

Nissen [2] proposed KM lifecycle model consisting of six phases: create, formalize, organize, distribute, apply and evolve. It shows as Figure 2.

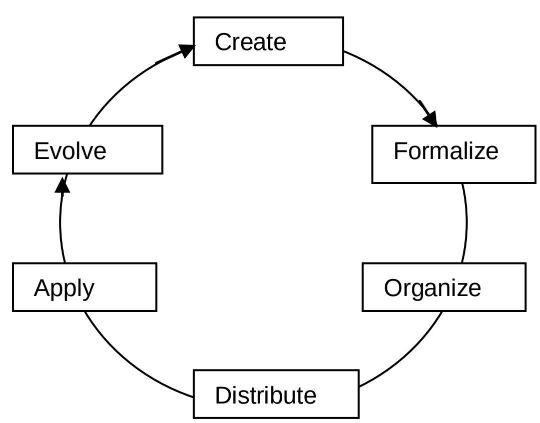


Fig 2. Knowledge Management LifeCycle

Dhanashree R. Kamble
Dept. of Computer Science & Engineerin, School of Engineering & Technology, IFTM University, Moradabad 2dhanashree@gmail.com

The knowledge management lifecycle starts with the create phase where new knowledge is created. Then the newly created knowledge is formalized and therefore stored in knowledge base according to the knowledge organizing mechanism. When knowledge is required, it can be searched and use relevant knowledge by accessing knowledge base. Finally knowledge is applied and further evolved into new knowledge. It also leads to further knowledge creation and completing the lifecycle.

1.2. Knowledge Management in Industries

Engineering can be considered as a knowledge-oriented industry. Even low weighted projects need thought, knowledge and skills from many sources which may include electronic media, documents, and people. Various Engineering firms are there who have been managing knowledge informally for years but the industry who is facing challenges today mean that most of the organizations now required a more structured and consistent approach to knowledge management[11]. Industries implementing projects are usually organized into stages with deadlines and there are various teams assigned to those projects. The problem is that capturing 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. These types of problems generally bound the information flow and can create obstacle to knowledge learning. This paper outlines an approach to assessing and implementing Knowledge Management using Multi-Agents. It highlights the importance of aligning KM initiatives to the business goals. The rest of the paper is organized as follows. Section 2 provides the literature review and current techniques, comparisons of different KM frameworks and research related to implementing multi-agent KMS. This section also discusses the various tools for implementing multi-agents and then shows the features of JADE when compared with others. Section 3 discusses the proposed work on Multi Agent Enterprise Knowledge Management System (MAEKMS). It also discusses the motive of implementing the proposed architecture. In Section 4, detailed design of a multi agent knowledge management system for a Public Sector Unit (PSU) is described. This section also described how the functionalities like knowledge creation, knowledge extraction, knowledge reusability are working in the MAEKMS architecture. Section 5 discusses the implementation of new Multi-Agent Enterprise Knowledge Management System (MAEKMS) architecture for PSU which we have proposed and its specialty. In section 6, we have shown the result and observations after implementing the proposed architecture. Finally, section 7 summarizes the conclusion and future work anticipated in this direction.

2. Related Work

2.1. Skills of KM in Organization

Individual knowledge plays the important role in an organization. Individual knowledge management should be supported by the organizations by monitoring at knowledge workers' actions and move toward to manage their personal knowledge. Encouragement must be there for efficient KM behaviors in the organization. KM tool should help in enhancing the efficiency at work and should be able to help in managing time. KM tools consist of communities of practice by which people create networks of personal significant knowledge. KM within an organization can be empowered with coordination, collaboration, and cooperation.

2.2. 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:

1. **Juan P. Soto KM framework[16]**:
    - Juan described the framework by dividing into two parts:-Knowledge Agency and User Agency
    - He explained how the agents are structured into the two agencies with the goal of joining all the agents closely in charge of managing knowledge and supporting the different stages of the knowledge model.
    - INGENIAS methodology is used(an agent oriented software engineering methodology for multi-agent systems development) for developing the KM framework

2. **Meso and Smith KMS Architecture[17]:**
    - The architecture consists of functions, technology and knowledge.
    - Their architecture proposed that technology should include e-task management, messaging etc.
    - The functions utilize KM processes in using the knowledge, finding the knowledge, creating and packaging knowledge.
    - The knowledge is used for answering the questions such as the know-how, know-what, know-why etc.

3. **Nonaka and Takeouchi 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)

4. **Hahn and Subramani framework for knowledge support[18]:**
    - They suggested two important considerations for managing knowledge i.e. locus of the knowledge and level of priori structure.
    - The locus of the knowledge determines how a user and an expert are connected by the KMS.
    - The level of a priori structure determines the extent to which KMS usage imposes the burden of a translation to a form that corresponds to implicit logic underlying the priori structure.

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. Different agent-based architectures have been proposed to support activities related to KM [3-7]. Some architecture has been designed for the development of KMS. However, most of them focus on a particular domain and can only be used under specific circumstances. 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, 3APL, and Agent Factory, JADE.

Features of Cougaar [9]:-

- 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.

Features of JACK [10]:-

- JACK is a commercialized framework which is based on BDI agent realization [16].
- 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's 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 based on FIPA Specifications.
- 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

2.2.1 Newly developed Technology:-

Features of Agent Factory of Deepak Dahiya and pooja Jain, 2012-

> Agent Factory [22] is also based upon BDI concepts and it also includes an interpreter embedded within a distributed FIPA-compliant Run-Time Environment that allows the deployment of agents on desktops and servers, personal digital assistants, and mobile phones. Agent Factory provides a set of tools to help programmers in the development tasks; in particular, a tool for the construction of agents from a set of UML sequence diagrams and a suite of tools that facilitate the testing and debugging of applications.

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.

Deepak Dahiya and pooja Jain 2012 framework

- According to this framework, organization as knowledge can consist of four sets of social enhanced knowledge. Knowledge processes: (1) Creation (also known as Construction) , (2) Storage/retrieval , (3) Transfer, (4) Application
- These processes do not represent a monolithic set of activities, but an interconnected and interviewed set of activities, as explained later in this section.
- Services can be used in multiple application and functionality of the services implemented once
- User needs to Authenticate himself by sing system
- Once authenticated user interface agent gives the appropriate option to the user.
- User can select depending upon his requirement.
- There is one-to-one correspondence between the agent and clien
- Service autonomy
- Reusability, Interoperability , Scalability , cost efficiency

2.3. 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]
- JADE is distributed agent platform**.** The agent platform can be split on several hosts. On each host only one Java Virtual Machine, is executed. Agents are implemented as one Java thread and for effective communication between agents, Java events are used on the same host. An agent can execute parallel tasks, and JADE schedules these tasks in a cooperative way.
- 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.
- Library to manage user-defined Ontologies and content languages.
- GUI is used to manage several agents and agent platforms from the same agent. The activity of each platform can be monitored and logged.

2.4. 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 one of the best technologies when dealing with distributed and collaborative environment in knowledge management. Software agents are being used widely in software applications which range from small systems to large complex systems. 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.

3. Proposed Work on Multi Agent Enterprise Knowledge Management System (MAEKMS)

Our proposed multi agent enterprise knowledge management (MAEKM) system architecture is based on serving knowledge with the help of multi-Agent System. The salient feature of this architecture is its simplicity in nature. Our new architecture minimizes the time overhead required for knowledge serving. It provides smooth communication between agents which will guarantee to produce the desired result.

3.1. Motivation for the work on MAEKMS 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 queries
- Lack of appropriate knowledge
- Degradation of overall quality
- Lack of effective communication when dealing with customer knowledge queries

Therefore we have 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

We 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.

3.2. Proposed Architecture on Multi agent enterprise knowledge management System (MAEKMS)

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:** Our proposed architecture is simple in the sense that management of the knowledge by the agents is done in a very simple and smooth manner and can be understand very easily.

- **Reducing time overhead for knowledge serving:** In the architecture, most of the work of knowledge serving is done automatically by the agents. Therefore knowledge will be served in rapid manner.
- **Automatic computation between agents:** Automatic computation is done in this architecture at Agent's Inter-communication layer.
- **Reusability of the knowledge by agents:** In our proposed architecture, agents reuse their personal knowledge for the related queries if the requested query is very similar to the past queries. It is shown in section 4.3.

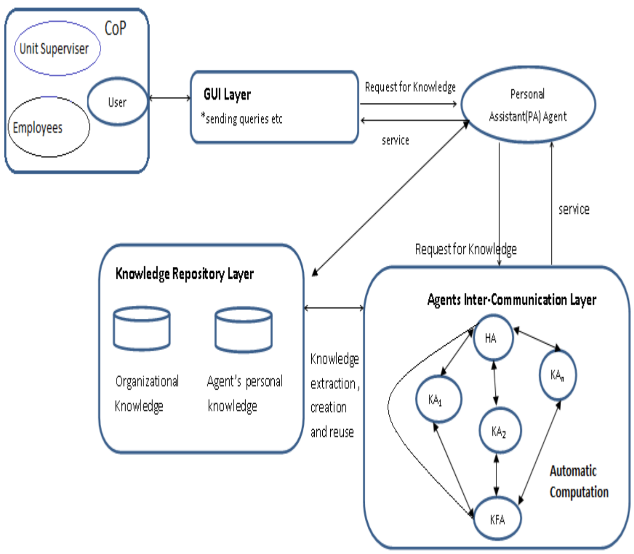


Fig 3: General MAEKMS Architecture

The architecture consists of 4 layers:

CoP Layer: Community of Practice layer consists of users, public employees etc.

GUI Layer: It is the 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 an interface between user and the system.

Agent Inter Communication Layer: It 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. In this layer, automatic computation is done between the agents.

Knowledge Repository Layer: It is the lower most layer. It contains organization's overall knowledge and agent's personal knowledge where all the information is store, retrieve and reuse.

3.3. Multi-Agents Descriptions

- **Head Agent (HA):-** It handles and manages 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 processed 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. Detailed Design Of A MAEKMS For A Public Sector Unit (PSU)

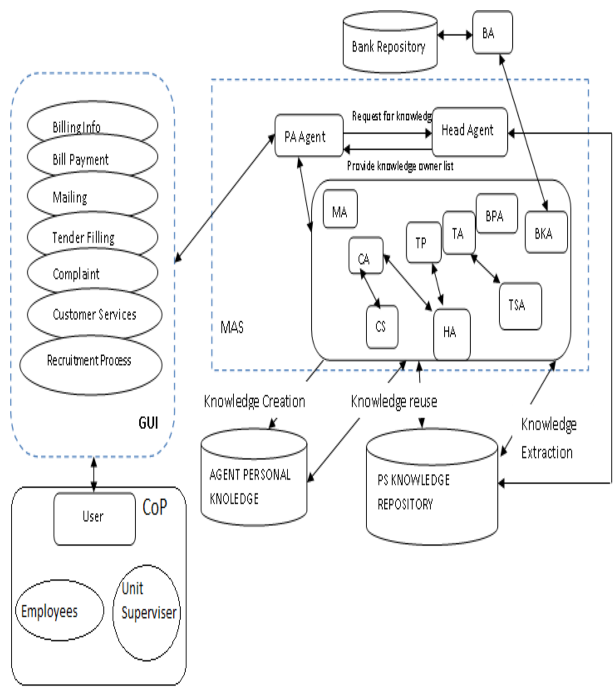


Fig 4: multi Agent KMS for PSU

Based on MAEKMS architecture, we are implementing 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 personal assistant (PA) agent. The conditions set by the customer like type of connection etc. His assistant will serve the request and contacts the head agent (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 personal assistant will then contact the required agent and try to get the best deal for his request.

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 & Placement Cell
- **PA:-** Personal Assistant

4.1. Inter Agent Communication

In figure 5, interaction between Personal Assistant Agent, Head Agent and Customer Service Agent is depicted. This is actually an agent interaction model diagram. These types of diagrams are very useful to see, at first glance, as agents interact with each other.

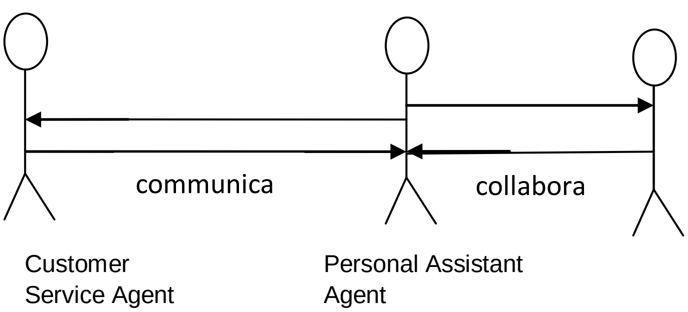


Fig. 5 Interaction between Agents

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 Head Agent for serving query for new connection on behalf of its user.
- Head Agent then replies to PA agent with the list of Agents serving for the related query.
- PA Agent then interacts with Customer Service Agent (CSA) and request for the related query.
- Finally CSA serves the required knowledge to the PA Agent

4.2. Knowledge Creation

Figure 6 shows how creation of the knowledge is done in our implementation.

The figure given below can be described as follows:
Knowledge is created by the Customer Service Agent 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, Customer Service Agent stores the knowledge of most preferred connection plan in its personal database.

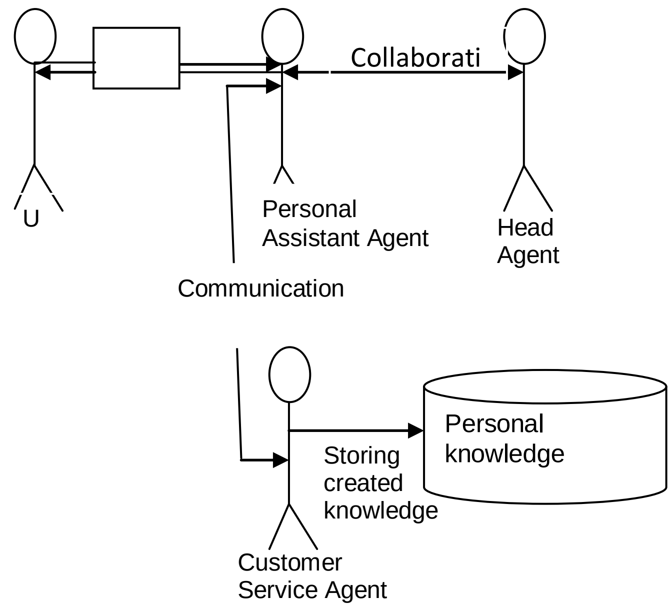


Fig. 6 Knowledge Creation

4.3 Knowledge re-use

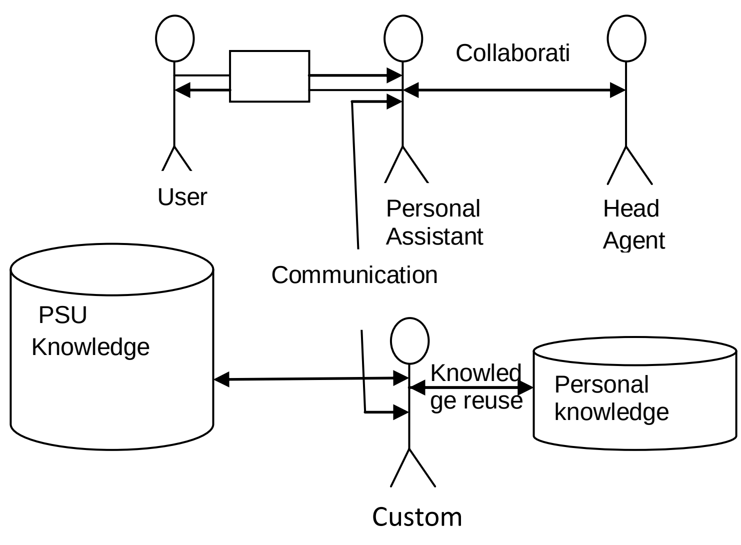


Fig. 7 Knowledge Re-use

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 Customer Service Agent will re-use its knowledge and serve the required knowledge to the PA agent. The steps are as follows:-

a) User requests for some query related to connection plan.
b) PA Agent forward this request by asking the list of agents responsible for the query from the head Agent

c) Head Agent then fetches the list of agents of related query from its personal database and sends back to PA Agent.
d) PA Agent then contact to the customer service agent and request for the result of that query.
e) Customer Service agent (CSA) first checks that whether the requested query is the similar query from the past requested queries or not from its personal knowledgebase.
f) If yes then CSA will reuse that knowledge and bring out the result to the PA Agent
g) Otherwise CSA Agent will fetch the possible result from PSU knowledge base and send back to the PA Agent.

5. Implementation Work On MAEKMS For PSU

5.1 Tools and technologies used
PSU Multi Agent KM architecture is implemented using Java Agent Communication Language with JADE [1] under Eclipse Environment, Oracle 10g database and UML.

5.2 The Yellow Pages Service
A 'yellow pages' service allows agents to publish deescriptions of one or more services they provide in order that other agents can easily discover and exploit them. Any agent can both register (publish) services and search for (discover) services. Registrations, deregistrations, modifications and searches can be performed at any time during an agent's lifetime. Coding for Customer Service Agent publishing their services in yellow pages is given below:-

```
DFAgentDescription dfd = new DFAgentDescription ();
dfd.setName (getAID ());

ServiceDescription sd = new ServiceDescription ();
sd.setType ("Customer-Services");

sd.setName ("PSU KNOWLEDGE MANAGEMENT");
dfd.addServices (sd);

try {

    DFService.register (this, dfd);}

    catch (FIPAException fe)

{

fe.printStackTrace ();

}
```

5.3 Implementation Snapshots

5.3.1 Implementing Knowledge Reusability
Figure 8 shows the communication between PA Agent and Customer Service Agent in which PA Agent sends the requested query of the User to the Customer Sercvice Agent.

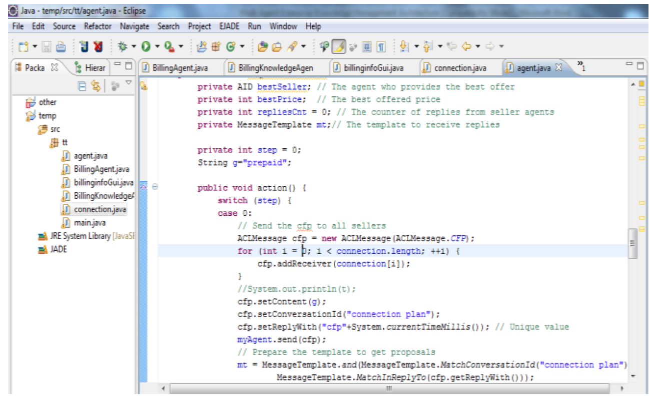


Fig 8 PA Agent send requested query to Customer Service Agent

In Fig 9, Customer Service Agent reuses knowledge from its personal knowledgebase and therefore serves the requested query to the PA Agent. This figure shows the Knowledge Reusability in our architecture.

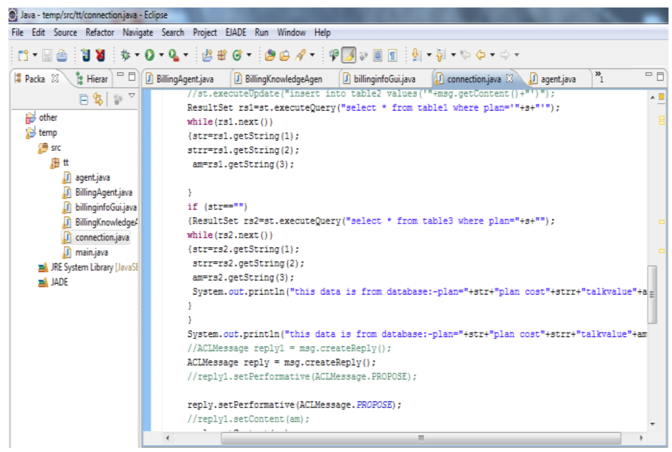
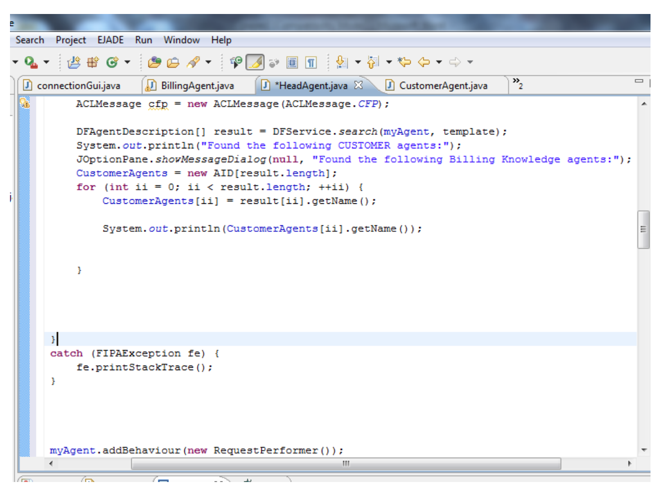


Fig 9 Customer Service Agent reuses its knowledge and serves for the requested query

5.3.2 Head Agent finding Customer Agent:



5.3.3 Fig 10 coding of HeadAgent finding Customer Agent

The code in fig. 10 will be run by Head Agent which finds all the customer agents who have registered their services in yellow pages.

5.3.4 Agents Inter-Communication Snapshots

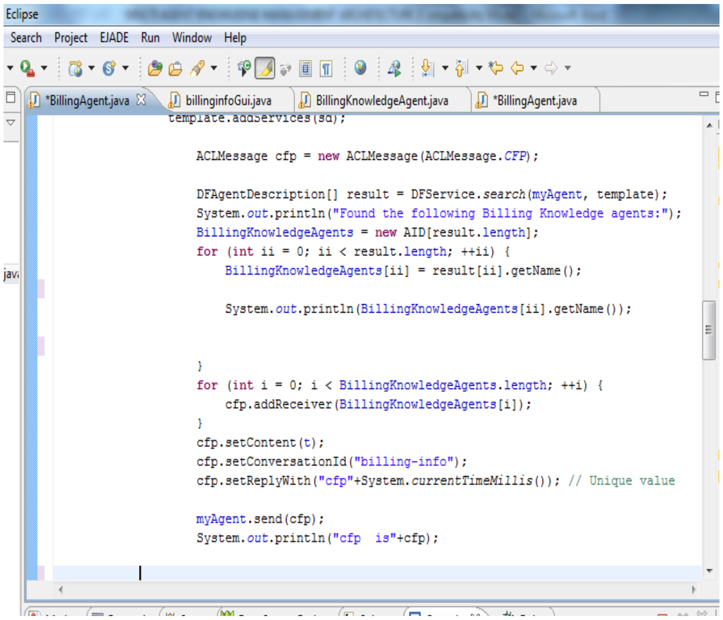


Fig. 11 Sending Message from Billing Agent to BillingKnowledgeAgent

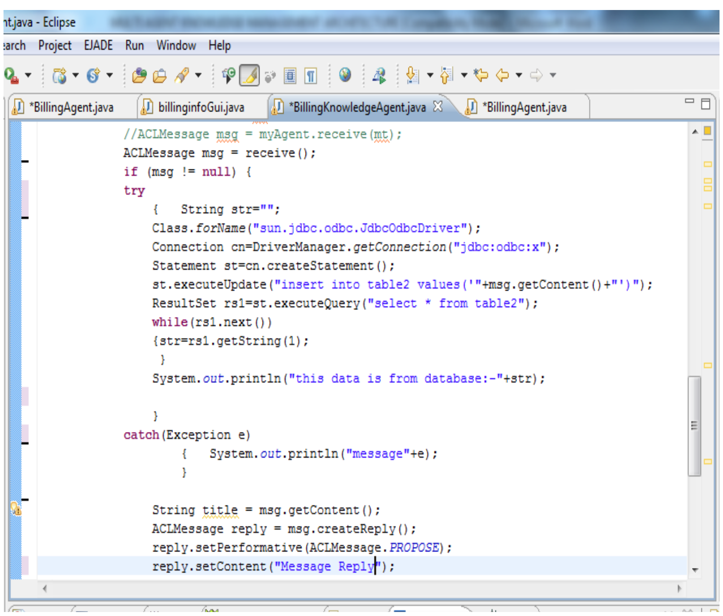


Fig 12. Receiving Message from BillingKnowledgeAgent and reply to Billing Agent

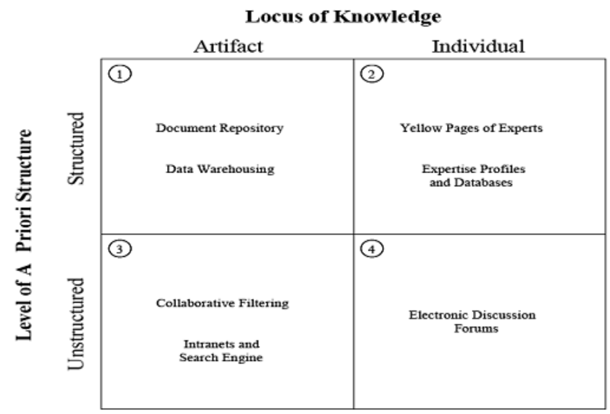
6. Results and Observations

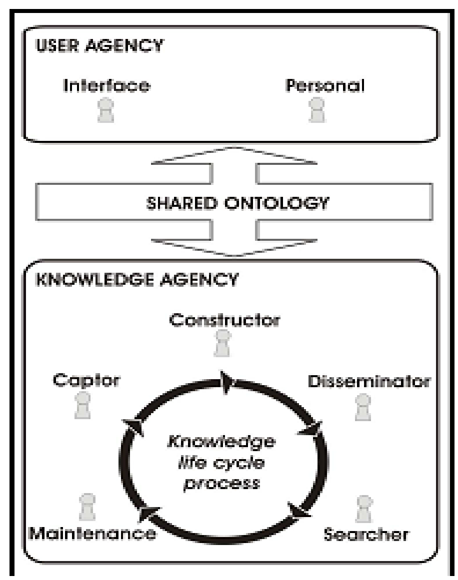
Based on the architecture and the detailed design proposed on MAEKMS and the subsequent implementation of some of the modules, a framework comparison was carried of the following frameworks: Juan P Suto Framework and MAEKMS Framework, Hahn and Subramani Framework and MAEKMS Framework

6.1 Framework Comparison

6.1.1. Juan P Suto Framework and MAEKM Framework:

According to fig 13(a), constructer agent is responsible for storing knowledgebase but where to store is not mentioned in the framework of Suto while in our proposed MAEKM Framework, repository layer is there which is helpful in knowledge reusability, knowledge extraction as well as knowledge creation as shown in fig 13(b).





(a) Juan Framework without repository layer[16]
(b) MAEKM Framework with repository layer

Fig 13. Juan P Suto Framework and Our proposed MAEKM framework

6.1.2. Hahn and Subramani Framework and MAEKM Framework:

Hahn and Subramani Framework(figure 14) suggested two important considerations for managing knowledge i.e. locus of the knowledge and level of priori structure. The level of a priori structure determines the extent to which KMS usage imposes the burden of a translation to a form that corresponds to implicit logic underlying the priori structure. The locus of the knowledge determines how a user and an expert are connected by the KMS. But this framework unable to show the functionality of Knowledge Reusability while in our proposed MAEKM Framework, there is a concept of knowledge Reusability functionality which is described in section 4.3 in detail.

Fig .14 Hahn and Subramani Framework without Knowledge Reusability [18]

To summarize, the above results and observations on comparison of the various frameworks with MAEKMS can be summarized in the form of table shown below:

Table 1 Comparison between the Frameworks

Item/Framework	Juan P.Suto framework	Meos and Smith Architecture	Hahn and Subramani Framework	Proposed MAEKM Architecture
Application	YES	NA	NA	YES
Knowledge Reusability	YES	YES	NA	YES
Repository	NA	NA	YES	YES
KM Process	YES	YES	NA	YES
Multi Agent Collaboration	YES	NA	NA	YES

7. Conclusion and Future Work

This paper has proposed the architecture, detailed design and implementation of MAEKMS for a PSU. Further, the work done has been compared with other existing frameworks in terms of knowledge creation, knowledge reusability and knowledge retrieving of the overall system. Overall, it is observed that:-

- MAEKMS architecture is reusing, retrieving the knowledge in a good manner
- MAEKMS architecture is providing smooth communication between agents
- MAEKMS architecture is providing automatic computation between agents.
- Providing best possible solutions according to the related queries.

This paper has partially reviewed implemented the Multi Agent Enterprise Knowledge Management System (MAEKMS) 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. Implementing MAEKMS architecture using Multi Agent result in more reliability, fast processing etc. The future work on this paper will incorporate complete implementation of this 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 Bringing 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]. S.A. DeLoach and M. Wood, "Developing Multiagent Systems With Agent Tool". In 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 Artificial Intelligence. Springer- Verlag, Berlin, 2001.

[7]. T. Kawamura, N. Yoshioka, T. Hasegawa, A. Ohsuga and S. Honiden. "Bee-Agent: Bonding And Encapsulation Enhancement Agent Framework For Development Of Distributed Systems". Proceedings of the 6th Asia-Pacific Software Engineering Conference, 1999.

[8]. L. van Elst, V. Dignum, and A. Abecker, "Agent-Mediated Knowledge Management." In proceedings of the Agent-Mediated Knowledge Management, Stanford (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"; 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 InternationalConference on Multi-Agent Systems, San Francisco, CA, 1995, pp. 312–319.

[14]. A. D. Marwick, "Knowledge Management Technology," IBM Systems Journal, 2001.

[15]. Nonaka, I., and Takeouchi, H. (1995), "The Knowledge-Creating Company". NY: Oxford University Press.

[16]. Juan P. Soto, Aurora Vizcaíno, Javier Portillo, Mario Piattini (2006), "Modelling A Knowledge Management System Architecture With INGENIAS Methodology". IEEE proceedings of the 15th international conference on computing.

[17]. Meso, P., & Smith, R. (2000). "A Resource-Based View Of Organizational Knowledge Management Systems, Journal Of Knowledge Management", 4(3), pp.224-234.

[18]. Hahn, J., & Subramani, M. (2000). "A Framework Of Knowledge Management System: Issues And Challenges For Theory And Practice", Proceedings of the twenty first international conference on Information systems, ACM, Brisbane.

[19]. Gandon, F.: "Multi-Agent System to Support Exploiting an XML-based Corporate Memory", Proceedings PAKM'00 Basel, Switzerland

[20]. Van Elst, L., Abecker, A.: "Domain Ontology Agents in Distributed Organizational Memories". IJCAI'2001 Workshop on Knowledge Management and Organizational Memories, August (2001), Seattle, Washington, USA.

[21]. Peña-Mora, F., Hussein K., Vadhavkar, S., Benjamin, K.: "CAIRO: A Concurrent Engineering Meeting Environment For Virtual Design Teams". Artificial Intelligence In Engineering.

[22]. Khaisar Muneebulla Khan, Umesha Naik: "Open Source Software for Personal Information Managers and Personal Knowledge Management". Proceedings of IEEE.

[23]. Grundspenkis, J. (2007), "Agent based approach for organization and personal knowledge modelling: knowledge management perspective", Journal of Intelligent Manufacturing 18(4): 451-457,

[24]. Wright, Kirby (2005), "Personal knowledge management: supporting individual knowledge worker performance", Knowledge Management Research and Practice 3: 156-165.

[25]. M.C. Pettenat and M.E. Cigognini. Methods and tools for developing personal knowledge management skills in the connectivist era.

[26]. http://elilearning.wordpress.com/2007/09/24/eden-2007-methods-and-tools-for-developing-personal-knowledge-management-skills-in-the-connectivist-era

[27]. Xian-yun Wang , Fu-yuanXu: "Study on Knowledge Management System Based on MAS", International Conference on Networking and Digital Society 2010.

[28]. I. Nonaka.: " The Knowledge-Creating Company". Harvard Business Review.

[29]. G. Van Heijst, R. Van der Spek. And E Kruizinga. -"Organizing Corporate Memories".Dans B.Gaines, M. Musen eds, Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96), Banff, Canada

[30]. An Architecture of a Multi Agent Enterprise Knowledge Management System Based on Service

[31]. Oriented ArchitectureIJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012 ISSN (Online): 1694-0814

[32]. Multi Agent Knowledge Management Architecture

[33]. Sheikh Amanur Rahman, Dipti Yadav, Prerna Agarwal, Pankaj Singh Bisht