International Journal of Engineering & Technology, 7 (3.8) (2018) 1-6

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Compendious Study of Interaction Protocols in Multiagent Systems

Dimple Juneja, Chetali Dhiman, Savneet Monga*, Ashutosh Kumar Singh

National Institute of Technology, Kurukshetra
**Email: Savneetmonga18@gmail.com*

Abstract

The premise of the paper is to present a compendious study of interaction protocols pertaining to multiagent systems. Multiagent systems have evolved from the field of Distributed Artificial Intelligence and require numerous agents to cooperate and coordinate to cope with goal search. The primary ingredients to goal search are the language of communication and the interaction protocol. Agents in communication must be able to understand the language of communication and should also follow rules of interaction. The paper focuses on sharing understanding about various agent interaction protocols and it also discusses the promises and challenges each protocol offers to MAS research community.

Keywords: *Agents; Agent Communication Language (ACL); Agent Communication Protocols; Distributed Artificial Intelligence (DAI); Multiagent Systems(MAS).*

1. Introduction

Distributed Artificial Intelligence (DAI) [34], a sub-field of Artificial Intelligence (AI) has its inception since the late 1960's to mid of 1970's and since then it has evolved and diversified rapidly. Its broad scope makes it difficult to describe in few words. DAI was conceptualized as a group of agents that could interact with each other to achieve some set of goals or perform some set of tasks. In fact, it deals with the interaction of intelligent agents [34] where an agent is a software or a hardware entity that senses some input from the environment and produces some output actions that affect it. A thermostat can be considered as the most common example of an agent. Thermostat is a temperature controlling device which is embedded with a sensor for detecting room temperature, in the environment which produces two outputs either Temperature is Low or Temperature is OK and correspondingly takes the action as shown in figure 1.

Although thermostat performs the desired action, still it can't be called as intelligent agent. An agent is intelligent if it can perform flexible and autonomous actions. An intelligent agent should respond timely, be able to take initiative to satisfy objectives and can interact with other agents. This interaction is usually non-terminating until some useful outcome is achieved. Sometimes, even dealing individually with such huge number of agents becomes challenging. Hence we need to deal with them as a group called the society of agents. Such a system with a large number of autonomous agents working together to achieve some objectives is called a MultiAgent System in which the agents communicate with one another using several interaction protocols to form an intelligent agent. In fact, a Multi-agent system is composed of several intelligent agents. Table 1 lists the prominent characteristics that empower an agent as intelligent.

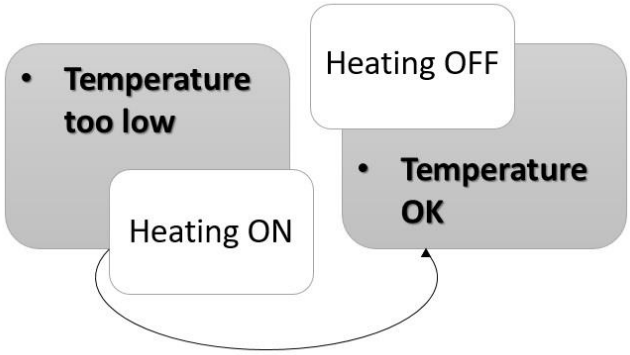


Fig 1: Thermostat as an Agent

Table 1: Characteristics of Intelligent Agent

Characteristics	Description
Learning/ Reasoning	Adapts behavior from experience
Reactivity	Capable of reacting appropriately
Autonomy	Control over actions and internal states
Goal oriented	Well defined goals
Communication	Needs to interact
Cooperation	To provide faster and better solutions
Mobility	May navigate within communication networks

MAS are finding application in variety of fields owing to various factors such as speeding up the system through parallelism, scalability, robustness, coordination and cooperation, just to list a few. The agents in a MAS continue to interact very frequently and for such an interaction various interaction protocols exist. The main intent of this paper is to present a compendious study of agent interaction protocols. The paper is structured into five sections. Section 1 provides a brief introduction about the emergence of MAS. Section 2 acknowledges the work of eminent researchers in the field of MAS and section 3 throws light on the various agent communication protocols and also presents a comparative view of

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the protocols thus studied. Section 4 discusses the scope of improvement and future research directions in the field. Section 5 finally concludes.

2. Related Work

The section highlights the work of renowned researchers working in the field of multiagent systems. MAS came into existence way back in 1970's and since then the field is growing exponentially. MAS are being described in variety of ways by various researchers. Huhns et al. [25] described agents as elements of a computational environment known as MAS. These elements help agents to communicate, cooperate and negotiate but further research is needed for the development of basis and techniques for societies of autonomous computational agents. WeiB [26]took the first steps towards learning to coordinate actions in multi-agent systems introducing the two elementary ACE algorithms [26] mentioning that the goal of future research is development of learning algorithms for more complex structured multi-agent systems. The work also described key dimensions for classifying multi-agent systems, as well as the criteria for characterizing single-agent and multi-agent learning. Jennings et al. [27] described the formal specification framework for the multi-agent systems. The components, their interaction, task hierarchy, task allocation, information and control flow within the agents and communication within the agents were also emphasized. Lee et al. [28] demonstrated the analysis of the performance and scalability of multi-agent systems highlighting the fact that the operating boundaries of such systems (the maximum number of tasks and agents that the system can sustain) need to be investigated for deployed applications. Rimassa et al. [29] highlighted the software framework for development of agents under FIPA specifications called Java Agent Development Environment (JADE) [29].Marin et al. [30] described the fault-tolerance feature of the agents and also the framework for fault-tolerant multi-agent systems. Goldman et al. [31] defined the greedy approach to build communication between decentralized cooperative multi-agent systems and showed that it produces near-optimal solutions but a heuristic approach is needed for correct tuning of the parameters. Balaji and Srinivasan [33] focused on design methodology of multi-agent systems and also highlighted the merits and demerits of the existing methods. Singh et al. [32] Investigated various existing protocols and concluded that most of the protocols have one or the other limitation.

An acute expression of the works highlight the fact that multiagent systems have been the focus of research for more than four decades and is still a developing field. MAS are finding applications in almost all fields ranging from personal home applications to unattended surveillance systems. Since, agents in MAS must communicate to achieve the maximum throughput, agent communication techniques play a vital role in development and further success of MAS. Next section throws light on existing agent communication techniques.

3. Agent Communication Techniques

Before describing languages and protocols needed by MAS, it shall be noted that an agent is usually designed with the ability to understand, judge, act and communicate as well. The communication is usually bidirectional between the sender agent and receiver agent as well. Communication enables an agent to coordinate their actions and behaviors with other agents. Figure 2 depicts various components of agent communication.

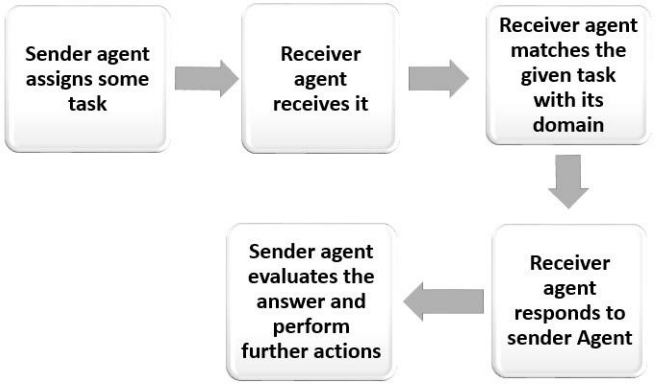


Fig 2: Components of agent communication

Agents with different capabilities must communicate with each other. They can communicate by participating in dialogue. An agent can be active (functions as master), passive (functions as slave) or both (as peer). An active agent must be able to issue queries and assertions and also can potentially control other agents by making them respond to the query. In contrast to active agent, a passive agent should be able to accept query from source and be able to respond to the query. An agent functioning as peer can act as active or passive in the dialogue. It must be able to make and accept queries and assertions. Table 2 delineates the capabilities of agents.

Table 2: Agent Capabilities

	Basic Agent	Passive Agent	Active Agent	Peer Agent
Receives assertions	√	√	√	√
Receives queries		√		√
Sends assertions		√	√	√
Sends queries			√	√

The coordinating agents cooperate using communication protocols specified at three levels. As shown in figure 3, lowest level depicts the interconnection methods whereas middle level specifies the format, syntax, and information to be transferred. Further, top level specifies semantics of the information. The three level protocol can serve as binary as well as n-ary communication protocols.

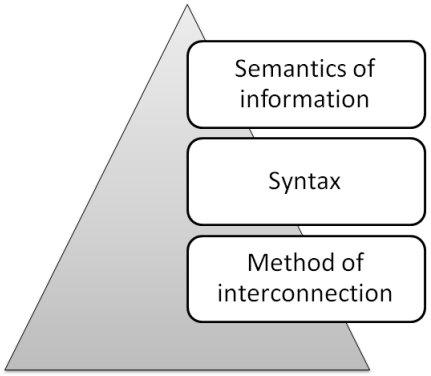


Fig 3: Three levels of communication protocols

Table 3 outlines the data structure of the message being communicated via any communication protocol.

Table 3: Message Format

Sender	Receiver(s)	Language in the protocol	Encoding and decoding functions	Actions to be taken by the receiver (s)

While an agent communicate with other agents it should be ensured that the semantics of the message is clear and the associated action should either be precisely defined or the agents should be designed intelligent enough to perceive the intended action. For example, the statement "I am cold" can be perceived as a simple statement, or as a request for a blanket, or a demand to increase the room temperature. Thus, if sender's intended communication is clearly defined then receiver should interpret the meaning unambiguously. In fact, the existing agent communication approaches such as Choreographies [19], Sequence Diagrams [22, 23] and State Machines [20, 21] do not deal with the meaning of the interactions. Instead, the main focus is on operational details of communication like occurrence and ordering of messages. Table 4 outlines a comparative view of the above listed agent communication approaches.

Further, the language chosen by a protocol for agent communication also play a vital role in the agent communication. Knowledge Query and Manipulation Language (KQML) [24] and Foundation of Intelligent Physical Agents (FIPA) [15] are the most popular agent communication languages (ACL) and are known as a standard format for the message passing. The best known ACL is KQML. KQML separates the semantics of protocols from the semantics of the bounded message. KQML is a performative based language where performatives are designed on the basis of speech-act theory. In fact, performatives are operations that agents perform on each other's knowledge and goal stores. As shown in table 5, KQML performatives can be classified into seven categories.

While an ACL facilitates the communication between two or more agents, Agent Interaction Protocols (AIP) [26] regulates the series of messages between agents. Either the two communicating agents agree to each other or a disagreement is expected to be resolved. In the former case, if the goals of agents are same and all have similar problems then the aim of protocol is to maintain the global coherence between agent without disturbing autonomy of agents and without enforcing explicit global control where as in later case i.e. in the event of disagreement, the main aim of protocol is to maximize the utilization of agents. Figure 4 depicts the important strands a protocol should decide upon.

Table 4: Comparative View of Existing Agent Communication Approaches [23,24]

	Choreographies	Sequence Diagram	State Machine
Definition	It is the arrangement or manipulation of actions (of agents) leading up to an event.	It is a construct of a message sequence chart (MSC).	It is used to describe a sequence of state transitions in an object in response to events
Purpose	It is used to state what each (agent) might expect from the others and what each might offer to the others.	Help find participating agents. Help identify necessary communication between agents.	Can help identify important agent's attributes. Can help refine the behavior description of an agent
Notation	It is a specification of the message flow among the participants. Participants adopt roles, that is, bind to the roles, in the choreography.	The roles of a protocol corresponds to the lifelines of MSC; each edge connecting two lifelines indicates a message from a sender to receiver Time flows downward by convention and the ordering of the messages is apparent from the chart.	It represents the message as transition between the states and state represent the action to be performed whena message is received.

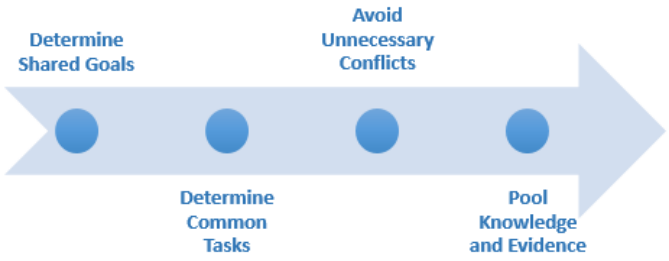


Fig 4: Important Aspects of a Protocol.

Literature indicates that such an agent communication is governed by various interaction protocols depicted in figure 5 and are being described in the upcoming subsection.

Table 5: Classification of KQML Performatives with their Meanings

Type of Performative	Performative	Meaning
Basic Query	evaluate ask-one	S wants R to simplify the sentence. S wants one of R(s) answer to a question.
Multi-response query	stream-all	Multiple response version of ask all
Response	reply sorry	Communicates expected reply. S cannot provide a more informative reply.
Generic Informational	tell untell	The sentences in S(s)VKB The sentence is not in S(s)VKB
Generator	standby ready	S wants R to be ready to respond to a performative. S is ready to respond to R(s) previously mentioned performative.
Capability-definition	advertise monitor	S is particularly suited to processing a performtive S wants to update to R(s) response to a stream all.
Networking	forward	S wants R to route a performative

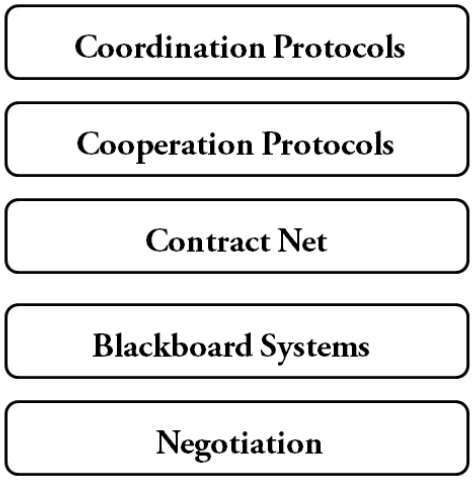


Fig 5: Agent Interaction Protocols

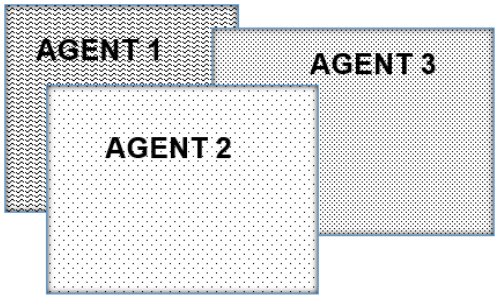
3.1 The Coordination Protocols

In MAS no single agent has sufficient resources, information, or has capability to achieve goals of the system. Due to limited resources in the environment agents have to coordinate their actions with each other for the sake of their interest as well as to achieve goals of the system. There is need for coordination as agents need to supply information which is meant to be sent on time to other agents. Also, actions of the agents must be synchronized. Coordination is also helpful in promoting non redundancy in problem solving. The key agent structures of goal search philosophy are commitment and conventions. Commitments can be viewed as an assurance of handling stated route of actions, whereas conventions gives a method of managing commitments when circumstances get change. A prediction of future activities of other agents is required so that agents could take these predictions into consideration while handling basic dependencies, global constraints, or conflicts in resource utilization. Commitments provide this predictability. With the change in the environment agents must check if the current commitments are yet valid or not. Conventions help in binding the conditions under which commitments need to be reevaluated. It also specifies the correlated actions that should then be started: either preserve remedies or abandon the commitments. An agent will only follow its commitments if the circumstances do not change. In this way agent will know before making any new commitments that it need to reserve some resources in order to facilitate its current commitments. Conventions help in managing the commitments however it does not specify in what way the agent should react with other agents, if any changes are made in the commitments. However, if the goals are dependent then agents must be informed of all subsequent changes.

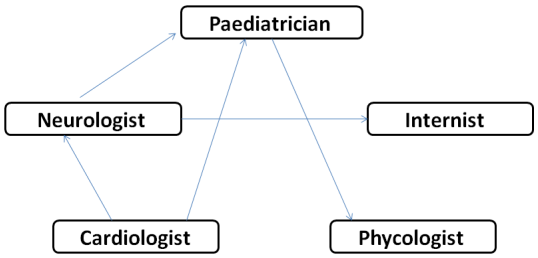
3.2 The Cooperation Protocols

Main approach followed by various protocols for cooperation is to initially decompose the task and then distribute it. Such partitions reduce the complexity of the task. As subtasks requires less intelligent agent and fewer resources. There may be conflict among agents and interaction between the subtasks. In fact, decomposition of task can be agent dependent or it can be carried out by the system designer. Decomposition of task can be either executed contiguously, on the basis of information source or functional by expertise (see figure 6). The decomposed tasks can be distributed further as per the following norms:

Tasks are assigned to agents that have similar potential as that of task.
An agent should be made with a wide view to assign tasks to other agents.
In order to attain coherence overlapping responsibilities should be assigned to agents.
To reduce the synchronization and communication cost highly interdependent tasks should be assigned to agents in contiguous proximity.
Tasks should be reassigned if it's crucial in order to accomplish urgent tasks.



(a): Spatial Decomposition on the basis of Information source or decision point.



(b): Functional Decomposition on the basis of Expertise

Fig 6 : Decomposition of Tasks

3.3 Contract Net Protocol

This is the best known and most widely applied protocol. Contract Net is the interaction protocol used for solving cooperative problems among agents. It is based on the contracting method used by businessmen to supervise the exchange of goods and services. Figure 7 defines the functioning of CNP:

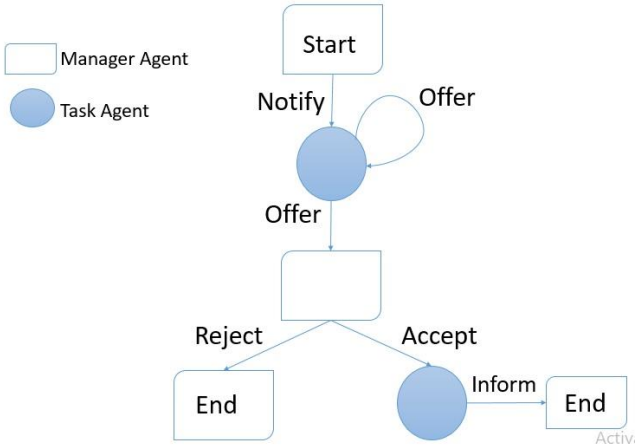
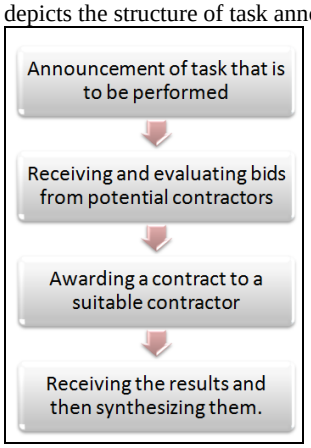
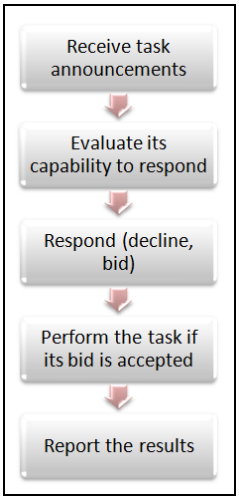


Fig 7: Contract Net Protocol

Agents in a contract are primarily of two types, one that wants to assign the task are managers and others are the agents having potential to perform the task are contractors. In a MAS role of an agent is not defined in advance. Any agent can make the task announcement and can act as manager. Agents that can respond to the announcement act as contractors. This allows further task decomposition. An agent acting as contractor can also act as manager for other responders. If a contractor does not respond appropriately then manager can assign task to another contractor. Figure 8 depicts the responsibilities of manager and contractors and table 6 depicts the structure of task announcement.



(a) : Manager's Tasks



(b) : Contractor's Tasks

Fig 8 : Responsibilities of Manager and Contractors

Table 6: Structure of Task Announcement

Data Structure	Description
Eligibility Specification	Tasks are assigned to some potential contractor that satisfies the eligibility specification
Task Abstraction	Brief description for the contractors so that they can rank the task from various task announcements
Bid Specifications	Potential contractors specify what information they should send along with bid.
Returned Bid Specification	Tells the manager the basis for comparison of bids sent by different promising contractors
Expiration time	Manager can consider only those bids that were received before the deadline.

Each potential contractor choose (ranks) tasks that are eligible for it from all the announced tasks and it accepts the most appropriate task (depending on some benchmark) and sends a bid to its manager. The manger receives and evaluates all the bids from all the task announcements. If any satisfactory bid is found then it accepts it before the expiration time. A manager then informs the potential contractor whose bid is accepted by announcement award message. Although CNP is a promising protocol, however; a manager may fails to receive bid for several reasons such as contractors are already overloaded and busy, contractor is idle but low prioritizes the proposed task, contractors are not capable of working on the task, just to list a few. Further, in certain cases manager either continues to wait indefinitely for a contractor to be free or assigns the task to a weaker contractor. Moreover, a manager does not take any obligation to inform other potential contractors that the task has been assigned.

3.4 Blackboard System

Figure 9 shows the architecture of a blackboard system.

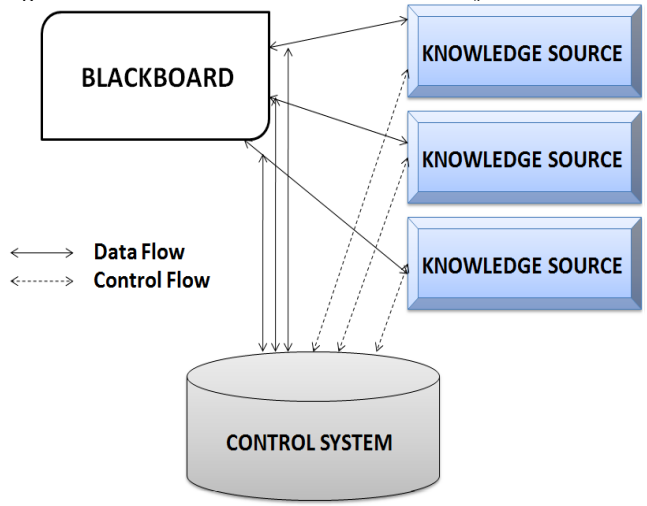


Fig 9: Architecture of a Blackboard System

A blackboard system comprises of variety of knowledge sources and a centralized control system. It requires specialized agents to works cooperatively to find the solution of the task. Here, blackboard acts as workplace where problem is to be solved. All agents start solving the problem as soon as the problem is delegated and its initial values are written onto the blackboard. The specialists watch the blackboard, waiting for an opportunity to apply their expertise to the developing solution. As soon as the specialist agent gets the sufficient information to contribute, the agent also records the contribution on the blackboard. With this additional information another specialist agent may be able to contribute. This adding of information by the specialists continues till the solution of the problem is obtained. Blackboard systems offer significant characteristics such as independence of expertise, diversity in problem-solving techniques, flexible representation of blackboard information, common interaction language, event-based activation and need for control [26, 35].

3.5 Negotiation Protocol

When agents try to fulfill individual goals and objectives, there may arise a condition where two or more agents need to communicate and take a joint decision, this is termed as negotiation. Firstly, the agents share their positions with each other, which might be conflicting. To provide a solution to it, they try to move towards an agreement by providing concessions or looking for the alternate decisions. The major requirements of negotiation are the language using which the agents are programmed, the negotiating protocol used by the agents, and the process by which the agents determine its positions, concessions, and criteria for agreement. Figure 10 depicts the terms of negotiation.

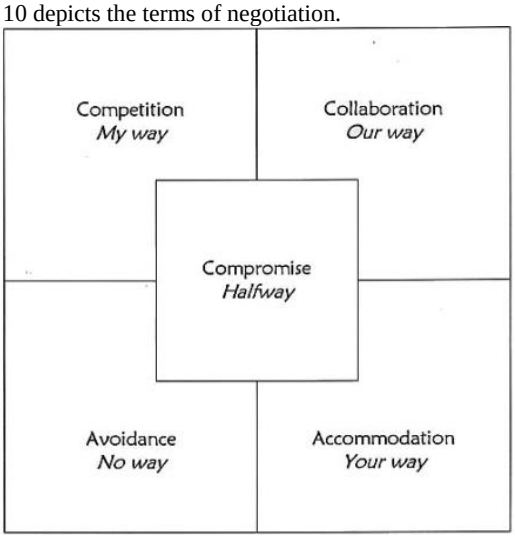


Fig 10: Terms of Negotiation

The negotiation protocols are mostly developed for specific problems. Hence there are few general principles of negotiation. In few cases, the negotiating protocols and their respective components are formalized through the speech-act classifiers together with a possible world semantics. On the other hand, it is assumed that the agents are economically rational and hence negotiation can be carried out easily.

Table 7: depicts a comparative view of various interaction protocols discussed in the previous section.

	Coordination	Cooperation	Negotiation	Contract Net
Year	Mid 70's	Mid 70's	1983	1980s
Author	Unknown	Unknown	Reid Smith	Reid Smith
Function	Commitments and Conventions are the corner stones	Decomposition and distribution is the key	Joint decision reached by two or more agents.	Modelled on the contracting mechanism
Limitation	If change in commitments not conveyed then the system may crash	Identification of potential agent is difficult	The decision may be biased	It is not iterative.

4. Discussion

The compendious study of MAS enlightens few important observations which are worth discussion. MAS are finding attention of researchers of varied domains such as Wireless Sensor Networks, Social Networks, Big Data, Cloud Computing, just to list a few. The performance of MAS not only depends on the application of employment but also on the language and protocols chosen for

communication and also on the platform of development. In fact, the research indicates that CNP is the most used protocol to establish communication among agents [12, 13, 14, 15]. During 1980s, CNP was one of the most important paradigms developed in DAI for decentralized task allocation. In 1993, the TRACONET (TRAnsportation COoperation NET) system was presented; the formalization is based on marginal cost calculations based on local agent criteria. The TRACONET system was seen as an extension of CNP in commitment strategy. Still the communication amongst MAS is based on CNP and its extended versions. Although CNP has its benefits and promises that make t the most competitive protocol, however it also comes with certain limitations. For instance, it can only entertain one request at a time. Another agent's request cannot be handled. At any time during the communication, the receiver can inform the sender that it did not understand what was communicated. This may terminate the entire protocol. Hence the semantics of canceling the interaction protocol should be terminated in a manner acceptable to both the initiator and the participant. Further it offers no multi-round iterative bidding. The current study reflects that there is ample scope of improvement in CNP and hence the motivation for future works.

5. Conclusion

The paper presented an exhaustive study about multiagent systems. The understanding of multiagent systems primarily focused on two domains namely, agent communication languages and protocols. The engraved study of literature reflected that KQML and FIPA-ACL are the two most popular agent communication languages while CNP dominates the segment of interaction protocols. This is so primarily due to the simplicity of the protocol, the performative used for communication and unambiguous semantics it offers. However, CNP also has certain barriers to cope with such as non-iterative nature, failure to handle multiple agents at the same time and failure to distribute to tasks to a cluster of agents. The authors believe that CNP based on clustering with iteration would be a better approach offering performance upgrade to MAS.

References

[1] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2):115-152, 1995.

[2] R. P. Bonasso, D. Kortenkamp, D. P. Miller, and M. Slack. Experiences with an architecture for intelligent, reactive agents. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, Intelligent Agents II (LNAI Volume 1037), pages 187202. Springer-Verlag: Berlin, Germany, 1996.

[3] S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice-Hall, 1995.

[4] M. Wooldridge. Agent-based software engineering. IEE Transactions on Software Engineering, 144(1):26-37, February 1997.

[5] Randall Davis and Reid G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. Artificial Intelligence, Vol. 20, No. 1, pages 63-109, January 1983.

[6] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Coherent Cooperation among Communicating Problem Solvers. IEEE Transactions on Computers, C-36(11):1275-1291, 1987.

[7] Les Gasser and Michael N. Huhns, editors.Distributed Artificial Intelligence, Volume II. Pitman Publishing, London, 1989.

[8] Afsaneh Haddadi. Towards a Pragmatic Theory of Interactions.In Proc. International Conference on MultiAgent Systems (ICMAS), San Francisco, 1995.

[9] M. N. Hubris, U. Mukhopadhyay, L. M. Stephens, and R. D. Bonnell. DAI for Document Retrieval: The MINDS Project. InM. N. Huhns, editor, Distributed Artificial Intelligence. Pittman, London, 1987.

[10] Nick R. Jennings. Coordination Techniques for distributed Artificial Intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, Foundations of Distributed Artificial Intelligence, pages 187-210. John Wiley & Sons, Inc., New York, 1996.

[11] Reid G. Smith. The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers, Vol. C-29, No. 12, pages 1104-1113, December 1980.

[12] Singh, Rashmi, Aarti Singh, and Saurabh Mukherjee. "A Critical Investigation of Agent Interaction Protocols in Multiagent Systems." International journal of Advancements in Technology5.1 (2014): 72-81.

[13] Xu, Lai, and Hans Weigand. "The evolution of the contract net protocol." International Conference on Web-Age Information Management. Springer, Berlin, Heidelberg, 2001.

[14] Juneja, Dimple, Ankit Jagga, and A. Singh. "A review of FIPA standardized agent communication language and interaction protocols." Journal of Network Communications and Emerging Technologies 5.2 (2015): 179-191.

[15] Fip, A. "FIPA Iterated Contract Net Interaction Protocol Specification." (2001).

[16] Specifications, F. I. P. A. "Fipa Contract Net Interaction Protocol Specification, version H." Architecture (2003).

[17] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. Developing Multi-Agent Systems with JADE. Wiley-Blackwell, 2007.

[18] Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, Reading, MA, 3rd edition, 2003.

[19] Baldoni, Matteo, et al. "Declarative Agent Languages and Technologies VI." 6th International Workshop DALT. 2008.

[20] Benatallah, Boualem, Fabio Casati, and Farouk Toumani. "Analysis and management of web service protocols." ER3288 (2004): 524-541.

[21] Yellin, Daniel M., and Robert E. Strom. "Protocol specifications and component adaptors." ACM Transactions on Programming Languages and Systems (TOPLAS) 19.2 (1997): 292-333.

[22] Huget, Marc-Philippe, and James Odell. "Representing agent interaction protocols with agent UML." International Workshop on Agent-Oriented Software Engineering. Springer Berlin Heidelberg, 2004..

[23] Bauer, Bernhard, Jörg P. Müller, and James Odell. "Agent UML: A formalism for specifying multiagent software systems." International journal of software engineering and knowledge engineering 11.03 (2001): 207-230.

[24] Chopra, Amit K., and Munindar P. Singh. "Contextualizing commitment protocol." Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. ACM, 2006.

[25] Huhns, Michael N., and Larry M. Stephens. "Multiagent systems and societies of agents." Multiagent systems: a modern approach to distributed artificial intelligence 1 (1999): 79-114.

[26] Wei, Gerhard. "Learning to coordinate actions in multi-agent systems." Proceedings of the international joint conference on artificial intelligence. 1993.

[27] Brazier, F. M., et al. "Formal specification of multi-agent systems: a real world case." (1995).

[28] Lee, Lyndon C., et al. "The stability, scalability and performance of multi-agent systems." BT Technology Journal16.3 (1998): 94-103.

[29] Bellifemine, Fabio, Agostino Poggi, and Giovanni Rimassa. "Developing multi-agent systems with JADE." International Workshop on Agent Theories, Architectures, and Languages. Springer, Berlin, Heidelberg, 2000.

[30] Marin, Olivier, et al. "Towards adaptive fault-tolerance for distributed multi-agent systems." Proceedings of ERSADS. 2001.

[31] Goldman, Claudia V., and Shlomo Zilberstein. "Optimizing information exchange in cooperative multi-agent systems." Proceedings of the second international joint conference on Autonomous agents and multiagent systems. ACM, 2003.

[32] Singh, Rashmi, Aarti Singh, and Saurabh Mukherjee. "A critical investigation of agent interaction protocols in multiagent systems." International journal of Advancements in Technology 5.2 (2014): 72-81.

[33] Balaji, P. G., and D. Srinivasan. "An introduction to multi-agent systems." Innovations in multi-agent systems and applications-1. Springer Berlin Heidelberg, 2010. 1-27.

[34] Weiss, Gerhard, ed. Multiagent systems: a modern approach to distributed artificial intelligence. MIT press, 1999.

[35] Corkill, Daniel D. "Blackboard systems." AI expert 6.9 (1991): 40-47.

[36] Lau, Billy Pik Lik, Ashutosh Kumar Singh, and Terence Peng Lian Tan. "A review on dependence graph in social reasoning mechanism." Artificial Intelligence Review 43.2 (2015): 229-242.

[37] Lau, Billy Pik Lik, Ashutosh Kumar Singh, and Terence Peng Lian Tan. "Weighted voting game based algorithm for joining a microscopic coalition." TENCON 2013-2013 IEEE Region 10 Conference (31194). IEEE, 2013.

[38] Lau, B., A. Singh, and Terence Peng Lian Tan. "T-DepExp: Simulating transitive dependence based coalition formation." Pertanika Journal of Science and Technology 23.1 (2015): 105-117.