



KQML Based Communication Protocol for Multi Agent Systems

Dimple Juneja^a, Ankit Jagga^b, Aarti Singh^c

^aDronacharya Institute of Management and Technology, Kurukshetra, Haryana, India

^{b,c}M.M.Institute of Computer Technology & Business Management,
Maharishi Markandeshwar University,
Mullana, Ambala, Haryana, India

Abstract: *The premise of this paper is to design a communication protocol for the agents operating in different multiagent systems. The protocol thus proposed offers full autonomy to agents for deciding if they are interested in communicating and collaborating with peer agents. The work proposes six iterative rules for establishing the communication and five new performatives in KQML to support the working of protocol in multiagent systems using KQML as communication language.*

Keywords: *Multiagent Systems, Communication Protocols, Heterogeneous Agents, Inter-agent Communication, KQML*

I. Introduction

A multiagent system (MAS) [22] comprises of both homogeneous and heterogeneous agents. Homogeneous agents have a common understanding on ontology, messages and rules while on contrary heterogeneous agents need to develop a platform to exchange messages. Agents belonging to different groups of MAS are usually incompatible but on the other hand it is desired that agents though belonging to different categories will cooperate and coordinate to achieve the target. This expectation demands that there should be a communication protocol for heterogeneous group of agents, in particular. Although there exists variety of protocols for establishing communication amongst agents but these are silent towards heterogeneity among agents. Heterogeneous agents may not have same set of information, few may not understand how to communicate and further sustain the relationship and moreover, it is inevitable to avoid intra-agent and inter-agent communication. For these agents to interact, it is desired that these agents agree on a uniform set of syntax and semantics so that messages can be exchanged and interaction can be made possible. Moreover, any agent communication language such as FIPA [11], KQML[10] etc. do not offer semantically enhanced approach for expressing the meaning of messages being exchanged. It is highly desired that agents share a common convention of messages so as to make communication more smooth and efficient. The common convention as mentioned above lays the foundation for developing a new consensus based communication protocol for establishing messages between heterogeneous agents.

The review of existing solutions unveiled that these network protocols are not suitable for addressing the communication needs of software intelligent agents. Further, there exist very limited number of solutions addressing the need and the solutions in operation are neither flexible nor autonomous. The solutions addressing the cooperation and coordination do not address the heterogeneity factor amongst agents and thus degrading the performance of the entire system. For instance, DynWes [16] is a dynamic protocol for web based services but works poorly in a heterogeneous environment. A moderator based protocol was proposed by Sibertin [21] but the same failed to validate the ontology shared between the group of agents. Few other protocols [7][17] are too rigid and are application specific and hence the motivation for current work.

The paper is structured as follows: Section 2 presents the related work. Section 3 proposes the new protocol for establishing communication and presents the novel performatives in KQML to be used while establishing communicating using the proposed protocol. Section 5 concludes by discussing the future scope of work.

II. Related Work

The section throws light on the work of renowned researchers and protocols that have been proposed so far. Shaolong and Qiang [20] have proposed an application independent protocol specifying the sequence of steps that agents should follow while solving a problem. Authors have used KQML as agent communication language for establishing the communication. Agent Communication Transfer Protocol [1] is an application layer protocol facilitating communication between heterogeneous agents. Authors claim that the said protocol is flexible, reliable and offers platform independence.

Koes and her co-authors [3] highlighted that most of the agent communication protocols are application specific and these are not able to import the communication language and constrains the scalability of the system. The authors in their work have proposed to supplement the potential abilities of agents operating in MAS and support the backchannels of communication with flexible protocols.

Barbuceanu and Fox proposed a Coordination Language (COOL) [2] to be integrated it in a structured conversation based on the coordination rules among agents. COOL is based on speech act theory. Considering the latest specifications of KQML and application requirements, Yong et al. [24] have proposed a series of algorithms for agent communication. Work by Chen and Lien [6] highlights the technological challenges pertaining in the field of machine to machine world. The detailed survey about agent communication languages is available in [23] and the study clearly reflects that although some work has been done on developing agent communication languages but that is based on mental agency, But since agents are rarely programmed using BDI notion [12][18], therefore when it comes to practical implementation of a MAS, the integrity and atomicity of messages is challengeable. Hence, there is gap between theory and practice and it need to be bridged. There is strong desire of standardizing conversation policy and efforts by [5][8][9][19] are worth acknowledging. Despite the fact that researchers associated with agent community have been putting efforts to improve and develop new mechanisms to ease the communication and conversation between heterogeneous agents, still there exist challenges pertaining to communication as mentioned above. Therefore, the focus of current research work is to propose a design of a new communication protocol which is autonomous and application independent. The proposed protocol is being supported with few new KQML performatives.

III. The Proposed Protocol

Agents in a multiagent system exchange number of messages to reach to specific goal. Further, in order to achieve a target, agents from different multiagent systems may also coordinate and get into conversation. When agents within a multiagent system communicate, it is known as intra-agent communication, while communication between agents belonging to different MAS is termed as inter-agent communication. The proposed protocol is generic as it defines the set of rules applicable to both types of communication.

Conventionally, in order to communicate, heterogeneous agents must either share a common ontology and communication language or there should be an interface that would be able to establish the communication. Primary challenge here would be that the interface should be in the position to corroborate the communication requirements at both ends. Moreover, the communication thus happening would not be a single message rather it would be a progression of messages. Therefore ensuring the progression of messages i.e. what is being communicated has been delivered to the receiving agent, is also equally important.

Considering the above two challenges, this work proposes a protocol which is independent of any interface and allows direct communication. Further, by implementing clock synchronization algorithm [4][15], order of messages is being ensured and the proposed protocol is based on extending the existing set of KQML performatives.

A. Abstract View of the Proposed Protocol

The protocol focuses on agent communication in multiagent systems and further assumes that all agents in communication are registered with certifying agency and are carrying a unique id for identification. To present a simplified view, communication between only two agents is being considered. When two agents are willing to communicate, they will be henceforth termed as partnering agents and will get into following five states as depicted in figure 1. The five states are:

- **Active State:** Agent is willing to initiate a communication with another agent.
- **Acquire State:** Agent starts acquiring the desired resources to establish the communication with partnering agent.
- **Waiting State:** Agent has sent the request to partnering agent and waiting to hear the response.
- **Busy State:** Connection between two agents has been established and both agents are busy communicating.
- **Sleep State:** Communication is over and agents get into sleep mode.

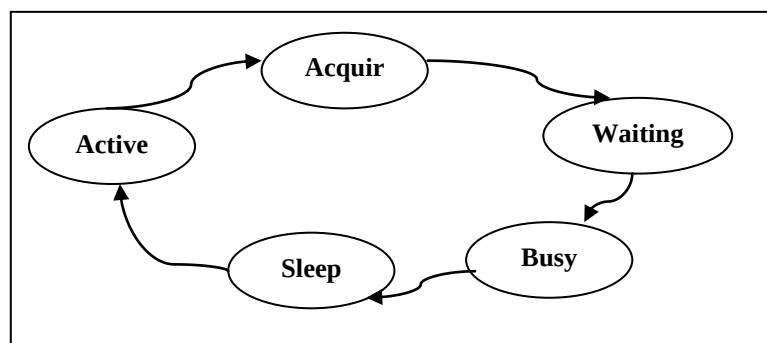


Figure 1: Partnering Agent States

The communication protocol thus proposed is the set of rules that governs the change in the state of partnering agents. Moreover, it also deals with managing the order of messages being communicated and notifying the

potential loss of any message. The following subsections propose the algorithms for the above mentioned objectives.

B. Rules Governing the Agent States

The state of agent would initially change when it initiates to establish a connection. Let us assume all agents not involved in any communication remain in sleep mode. Let $agent_i$ wants to communicate with $agent_j$. The rules for establishing the communication between $agent_i$ and $agent_j$ are listed as follows:

Rule 1: Change the state of $agent_i$ from sleep state to active state.

Rule 2: $agent_i$ pings $agent_j$ with its unique id leading to the state change of $agent_j$ from sleep to active state.

Rule 3: $agent_j$ if already in a state other than sleep state (may be communicating with other peers agents), delays the acknowledgement to $agent_i$ else changes its state from sleep to active and sends “ok” message to $agent_i$.

Rule 4: On receiving this message, both agents makes attempt to acquire all necessary resources and once acquired, changes the states from active to acquire. When both the agents in acquire state, only then communication can be further carried forward else both agents wait (leading to state change from active/acquire to waiting state) for a limited threshold time ($\tau > 0$). If communication fails, both agent changes their state from active/acquire to sleep/active states respectively else the connection is said to be established and communication can start leading to busy state.

Rule 5: Assuming the communication is allowed, $agent_i$ pings $agent_j$ delegating the desired task and changes its state from acquire to waiting. It now waits for a limited period. $agent_j$ is now required to inform $agent_i$ if it can take up the task by sending a “Yes” message. On receiving “Yes” from $agent_j$, $agent_i$ changes its state to busy implying that it is now busy in communicating with $agent_j$ and exchange of messages as required is allowed.

Rule 6: $agent_i$ on receiving the final response from $agent_j$, terminates the connection by sending the “Thank You” message to $agent_j$. $agent_j$ now changes its state to sleep again while $agent_i$ may change to any state according the task it is executing.

The working flowchart of the above protocol is given in figure 2. The proposed protocol makes use of clock synchronization algorithm [15] and logically time-stamps every message exchanged in order to maintain the integrity of message. Both $agent_i$ and $agent_j$ assign a successive timestamp to any message sent to the respective agent. For example, $agent_i$ time-stamps its first message (after the communication is allowed and agents are in ready state) sent to $agent_j$ as msg1 and $agent_j$ subsequently time-stamps the response message as msg2 and so on. This implies receiving agent always expects one higher time-stamp. If a message with time-stamp greater than one is received, the receiving agent assumes the potential loss of corresponding message and may request retransmission. Now, one agent might communicate with many agents in parallel. The packet being sent as message is used to identify the receiving agent and also the time-stamped message identifies the sequence of messages thus sent and received. Table 1 depicts the format of message being exchanged between $agent_i$ and $agent_j$.

Table 1: Packet Format of Message

Sender agent	Receiver Agent	Timestamp	Message Content
$agent_i$	$agent_j$	$t1$	Agent id

In order to implement the above stated protocol, KQML as language of communication formed the basis. Further, few new performatives are being proposed to implement the same in practice.

C. The Proposed KQML Performatives

KQML is one of the prominently used agent communication language and is known to be more powerful than other ACLs. A KQML message can only be sent either in the form of ASCII strings or binary notations [20]. Agents using KQML make use of set of pre-defined performatives for communication where performatives help in deciding the type of message content as an assertion, a query and a command. Literature survey [13][14] indicates that although reserved performatives addressing the communication needs are available in KQML but those controlling the order of messages are lacking.

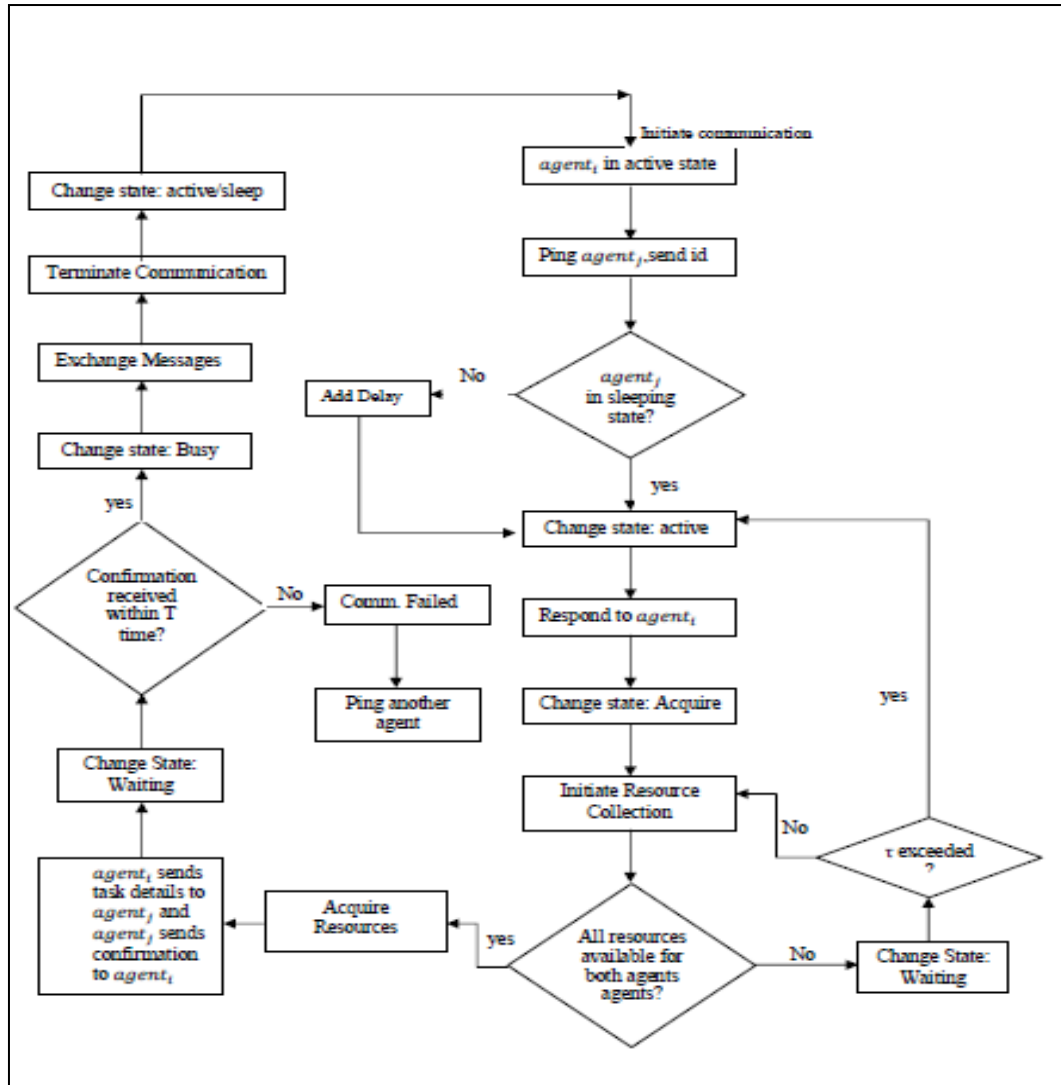


Figure 2: Working Flowchart of Proposed Protocol

Therefore, the section proposes few new performatives representing the state of agent as explained above and parameters for controlling the order of messages. The proposed extension also includes new reserved parameters BELIEF, DESIRE and INTENTION as discussed below.

The proposed five performatives representing the state of agents engaged in communication is *active*, *acquire*, *waiting*, *busy* and *sleep*. In order to explain the semantics associated with these performatives, following are the few assumptions made.

- $agent_i$ is the initiator agent and it sends the message to $agent_j$.
- $agent_j$ only communicates in response to messages received from $agent_i$.
- $msg_{timestamp}$ represent that message being communicated with logical timestamp.
- $agent_i$ and $agent_j$ must agree on preconditions, post-conditions and completion conditions as described in [14] as described below.
 - Preconditions :
 - $Pre(agent_i) : DESIRE(agent_i)$
 - $Pre(agent_j) : BELIEF(agent_j)$
 - Postconditions:
 - $Post(agent_i) : INTENSION(agent_j, Desire(agent_i))$
 - $Post(agent_j) : BELIEF(agent_i, Desire(agent_i))$
 - Completion Conditions
 - $Completion(agent_i) : DESIRE(agent_i)$
 - $Completion(agent_j) : INTENSION(agent_j)$

On the basis of previous discussions, the proposed performatives are described as follows.

- i. *active* ($agent_i, agent_j, msg_{timestamp}$)

- agent_i* wants communicate with *agent_j* and *msg_{timestamp}* contains the unique id of *agent_i* timestamped as *t1* since the first message initiating the communication.
- ii. *acquire(agent_i, agent_j, desire_{list}, belief_{list})*
agent_j agrees to communicate with *agent_i* and hence both agents are now required to acquire the desired aims and targets.
- iii. *wait(agent_i, threshold_time)*
agent_i has delegated the desire to *agent_i* and waits for the threshold time $\tau > 0$
- iv. *busy(agent_i, agent_j, msg_{timestamp})*
communication between both agents have been established and messages are being exchanged i.e. belief set is being mapped with desire set and finally intentions achieved would be delivered. Both agents continue to communicate till post conditions are met.
- v. *sleep(agent_i, agent_j, msg_{timestamp})*
Postconditions have been met. *agent_i* communicates “Thankyou” as the last message indicating the termination of communication. Both agents get into the sleep state. Ideally, it is expected that the all desires of *agent_i* have been mapped to belief set of *agent_j* but since practically, it is not achievable, therefore completion condition is required which states that maximum possible intentions i.e. targets could be achieved with the help of *agent_j*.

IV. Conclusions

The paper proposed a novel communication protocol which is not only flexible but also offers full autonomy to agents to decide if they are willing to communicate. The work introduced five new communication decisive performatives in KQML on the basis of which the state of agent changes and communication is established. Using this protocol, the messages could be transferred in sequential order and further missing sequence number could easily detect potential loss of message. Challenges pertaining to security of the messages and ensuring trustworthiness of participating agents are still untouched and can be taken up as future research.

V. References

- [1] Artikis, A., Pitt, J., & Stergiou, C. (2000, June). Agent communication transfer protocol. In Proceedings of the fourth international conference on Autonomous agents (pp. 491-498). ACM.
- [2] Barbuceanu, M., & Fox, M. S. (1995, June). COOL: A Language for Describing Coordination in Multi Agent Systems. In ICMAS (pp. 17-24).
- [3] Berna-Koes, M., Nourbakhsh, I., & Sycara, K. (2004, April). Communication efficiency in multi-agent systems. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on (Vol. 3, pp. 2129-2134). IEEE.
- [4] Carli, R., & Zampieri, S. (2014). Network clock synchronization based on the second-order linear consensus algorithm. Automatic Control, IEEE Transactions on, 59(2), 409-422.
- [5] Chaib-draa, B., & Dignum, F. (2002). Trends in agent communication language. Computational intelligence, 18(2), 89-101.
- [6] Chen, K. C., & Lien, S. Y. (2014). Machine-to-machine communications: Technologies and challenges. Ad Hoc Networks, 18, 3-23.
- [7] Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). Web services description language (WSDL) 1.1.
- [8] Dignum, F., & Greaves, M. (Eds.). (2000). Issues in agent communication (No. 1916). Springer Science & Business Media.
- [9] Ervin-Tripp, S. (2014, June). Speech acts, social meaning and social learning. In Language: Social Psychological Perspectives: Selected Papers from the First International Conference on Social Psychology and Language held at the University of Bristol, England, July 1979 (p. 389). Elsevier.
- [10] Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994, November). KQML as an agent communication language. In Proceedings of the third international conference on Information and knowledge management (pp. 456-463). ACM.
- [11] Fipa, A. C. L. (2002). Fipa acl message structure specification. Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00061/SC00061G.html> (30.6. 2004).
- [12] Kinny, D., Georgeff, M., & Rao, A. (1996). A methodology and modelling technique for systems of BDI agents. In Agents breaking away (pp. 56-71). Springer Berlin Heidelberg.
- [13] Labrou Y, Finin T W. A semantics approach for KQML: a general purpose communication language for software Agents. In Proc. of the Third International Conference on Information and Knowledge Management. 1994, 447—455].
- [14] Labrou, Y., & Finin, T. (1997). A Proposal for a new KQML Specification. Technical Report Technical Report TR-CS-97-03, University of Maryland Baltimore County.
- [15] Liu, J., Li, X., Liu, M., Cui, X., & Xu, D. (2014). A New Design of Clock Synchronization Algorithm. Advances in Mechanical Engineering, 2014.
- [16] McKinlay, M., & Tari, Z. (2002). DynWES-a dynamic and interoperable protocol for Web services. In Electronic Commerce, 2002. Proceedings. Third International Symposium on (pp. 74-83). IEEE.
- [17] Ouyang, C., Verbeek, E., Van Der Aalst, W. M., Breutel, S., Dumas, M., & Ter Hofstede, A. H. (2007). Formal semantics and analysis of control flow in WS-BPEL. Science of Computer Programming, 67(2), 162-198.
- [18] Rao, A. S., & Georgeff, M. P. (1995, June). BDI agents: From theory to practice. In ICMAS (Vol. 95, pp. 312-319).
- [19] Searle, J. R. (1969). Speech acts: An essay in the philosophy of language. Cambridge university press..
- [20] Shaolong, Z., & Qiang, W. (2010, May). A Conversation Protocol Design between Heterogeneous Agents Using KQML. In E-Business and E-Government (ICEE), 2010 International Conference on (pp. 1-4). IEEE.
- [21] Sibertin-Blanc, C., Hanachi, C., & Cardoso, J. (2000). Communication protocols as first-class components of multiagent systems. In MultiAgent Systems, 2000. Proceedings. Fourth International Conference on (pp. 437-438). IEEE.
- [22] Wooldridge, M. (2009). An introduction to multiagent systems. John Wiley & Sons.
- [23] Wooldridge, M., & Jennings, N. R. (1995). Agent theories, architectures, and languages: a survey. In Intelligent agents (pp. 1-39). Springer Berlin Heidelberg.
- [24] Yong, L., Congfu, X., Weidong, C., & Yunhe, P. (2004, June). KQML realization algorithms for agent communication. In Fifth World Congress on Intelligent Control and Automation (pp. 15-19).