ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 4, April 2015*

# Agent Commitments and Ranking of Commitment Protocols

**Harjot Kaur[1], Harpreet Kaur[2]**

Department of Computer Science Engineering, Guru Nanak Dev University, Gurdaspur, India[1]

**Abstract**: The commitments play a central role in multi agent systems. The protocols specifies step by step rules of interaction between communicating agents and thus is a fundamental part of agent communication. The paper reviews the concept of agent communication and agent commitments. The agent commitment protocols are used for interaction between agents and can deliver messages between them to achieve their objectives. These protocols may be defined at design time by the designer, but it poses various problems**.** So protocols are generated at runtime by the agent depending on the situation. The ranking of protocols is done so as to select one protocol amongst all generated protocols. Accordingly, this paper proposes the method that can be used for the ranking of the commitment protocols, the factors that can be considered for ranking of the commitment protocols, the one with the highest rank is selected. .

**Keywords**: Agent, communication, commitments, protocols, ranking.

## I. INTRODUCTION

An agent is simply a computer system that performs certain actions in order to achieve its design objectives. Agents have the ability to communicate and they can coordinate their actions and behaviour by communicating with each other. The agents can work without the intervention of humans, they can also control their behaviour and their internal state[11]. The agents are generally capable of sending and receiving messages and understand the messages send by others. The agents form a society, where they interact in an open environment. A simple example of an agent is of an air conditioner. It operates in two modes- either it is off or it is on. It is provided with a sensor that detects the room temperature. When the temperature of a room rises above the limit set by the user then it switches on automatically and when it attains a given temperature level then it turns off. So in this way, the agents take output from environment and responds accordingly.

Temperature higher than limit set → turn on
Temperature reaches the limit set → turn off

In an open environment, like for example the Internet, agents are usually designed by different constructors. So it is very important to define a standard that could be widely accepted and used. The interaction between different agents in an open environment depends on the adoption of a common standard Agent Communication Language (ACL)[4]. The Agent Communication Language (ACL) is based on *speech act theory*, messages are actions, or communicative acts, as they are intended to perform some action by virtue of being sent[6]. This approach to the study of communication helps in making it possible to treat communicative acts and other types of action in a similar way. Practically, all existing proposals in the field of ACL's are based on concept of speech act theory as it appears to be suitable to describe communicative interactions. Moreover, given that it provides an adequate approach to human communication, speech act theory allows one to treat communication among artificial and human agents in a uniform way - a crucial point to obtain successful mixed interactions among human beings and software agents. The communicative acts are performed by gents to manipulate and create commitments [16, 12]. We express the meaning of messages using the social notion of commitment. In the following sections, the agent commitments concept is explained, which also includes agent commitment protocols. Then the ranking of the protocols criteria is explained, the various factors that are considered while ranking are discussed.

## II. AGENT COMMITMENTS

A commitment develops a relationship between two parties. They are usually directed from one party to another and they can also be manipulated. Commitments are defined operationally within an object-oriented paradigm. The formal definition of commitments can be expressed as:

A *commitment C(debtor, creditor, antecedent, consequent)* state expresses the social contract between the agents debtor and creditor, such that if the antecedent holds, then the debtor is committed to the creditor to bring about the consequent. Each commitment has a state that represents the current state of the commitment in its life cycle. The *state* of a commitment evolves depending on the value of the antecedent and the consequent and also according to the operations performed by the debtor and the creditor of the commitment. The important property of commitments is that they can be manipulated. A debtor may create or cancel a commitment; a creditor may release it. More interestingly, a debtor may delegate a commitment to a new debtor, and a creditor may assign it to a new creditor. Six operations are defined on commitments. These are the create, cancel, release, assign, delegate and discharge[19]. A commitment is said to be a *precommitment* when it has been proposed, but not yet accepted or refused[12]. In such a case, we say that the (potential) debtor is

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 4, April 2015*

precommitted to the (potential) creditor. The social commitments are used in Multi agent systems. Social commitments have some interesting features distinguishing them from psychological commitments[5]. First, a social commitment is directed from one party (its debtor) to another (its creditor). This terminology reflects the fact that the debtor is committed to do something for the creditor. The second interesting, and also less common, aspect of social commitments is the idea of incorporating an organizational context into the notion of commitments. The organizational context of a commitment describes the organization or "system" in which the commitment arises, providing support of the commitments and interactions among autonomous parties. The debtor and creditor of the commitment would thus generally be members of the context organization. An example would be a commitment from a seller to a buyer operating within the jabong marketplace wherein the seller is committed to shipping some goods. That commitment references jabong as its organizational context. Here, jabong might penalize a seller who doesn't discharge the commitment. Commitments are defined operationally within an object-oriented paradigm. An agent that wants to engage in an interaction considers its own goals, makes assumptions about the other agents' goals, and proposes a set of commitments such that if accepted by the other agent, will lead the initial agent to realize its goal. When doing this generation, the agent also considers its own capabilities, so that it generates commitments that it can realize.

### *B. Agent Commitment Protocols*

The protocols specifies step by step rules of interaction between communicating agents and thus is a fundamental part of agent communication. The protocols are used for interaction that can be used to deliver messages between agents. The agents participate in a protocol to realize a service engagement. A protocol describes a pattern of communication between agents [20]. Mostly, protocols are defined at the design time. So the agents need not worry about which protocol to use at run time. But nowadays, agents may leave or enter the system and the same protocol cannot be used in every situation. For example, consider a situation in which a shopkeeper maybe knowing about the interaction protocol to be used to talk to the buyer but he may not be knowing about the interaction protocol to be used to talk to the deliverer [1]. So if these two agents meet, then they need to figure out a protocol to be used to complete their process.

The protocols defined at the design time have many drawbacks. The protocol library may not have the required protocol to be used. The agents are heterogeneous in nature, they may not comply with the available protocol even if the required task can be fulfilled. So an agent needs to generate a commitment protocol at run time itself. The agent generates a set of alternative protocols. The exact protocol that will be used is chosen after deliberations with the other agent. Having alternative protocols is also useful for recoverability. That is, if a protocol is chosen by the agents, but if one of the agents then violates a commitment, the goals will not be realized as expected. In this case, agents can switch to an alternative protocol. The commitment-based protocols cover a large context of agents by relating commitments to the agents goals, capabilities, and their knowledge of other agents' goals and capabilities.

## III. RANKING OF THE PROTOCOLS

Mostly, protocols are defined at the design time, therefore the agents need not worry about which protocol to use at run time. But in some situations the same protocol cannot be used. For example, consider a situation in which a seller maybe knowing about the communication protocol to be used to talk to the buyer but he may not be knowing about the communication protocol to be used to talk to the deliverer [1]. So if these two agents meet, then they need to figure out a protocol to be used to complete their process. Also nowadays in an open system, the agents may leave or enter the system. It is not necessary that the agents would use the same protocol as the agents inside. Even if the agents are fixed, the *environment* can also change in a way that requires run-time adaptation of protocols. For example, if the online payment service is temporarily unavailable, then the previous pay online protocol will not be sufficient.

Hence, the agents need to find and use a different protocol that involves pay cash on delivery service, where customer is expected to pay in cash. Such cases occur because the environment is dynamic. Moreover, the agents' preferences may also change with time [3]. For example, an agent that doesn't has necessary goods to be delivered, may come up with a protocol in which the agent either delays its delivery of goods or cancels the orders. Since the need becomes evident only at run-time, the agent should have the means to adapt and formulate a protocol that reflects its preferences.

It is difficult to account for all such cases when designing a protocol at design-time. The various protocols are generated at the run time by the agent but only one protocol is to be used to attain the goals and objectives. To choose the best protocol amongst all, the ranking of the protocols is done. The one with the highest rank is chosen and followed. There are various factors based on which ranking can be done. The two major factors that are already being used for ranking are risk and benefit[2,3]. Besides these factors few others can also be considered in future like the time taken to achieve a certain goal. The selection can be based on a range of factors such as the number of the commitments required (the fewer the better), or the strength of the argument for accepting a commitment. For example, one protocol may have a commitment which asks a merchant to ship goods should the customer pay (by any means), whereas another possible protocol may be asking the merchant to commit to shipping goods should the customer pay cash. It could be argued that the second commitment is more attractive to the merchant, and hence more likely to be accepted.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
Vol. 4, Issue 4, April 2015

*A. Risk*

The risk is always associated with the protocol, whether choosing a particular protocol would result in successful execution or not. The agents involved in execution may be unreliable [2]. The trust factor is very important for proper execution. For example, a buyer must trust the bookseller for the delivery of the books. The trust can be related to a particular service. We consider the trust of an agent a for another agent a′ with respect to a particular service S, denoted as Ta(a′, Sa′(d,u)) ∈ [0, 1]. This trust value represents how likely a′ is to complete service Sa′ (d, u) from agent a's perspective. A trust value of 1 would mean that agent a believes a′ would definitely carry out the service, whereas a trust value of 0 would mean that a believes a′ will definitely *not* carry out the service[2].The trust of an agent a in a protocol p(denoted as Ta(p) ) is calculated as inverse of risk.

Ra(p) = 1 − Ta(p).

In order to quantify the risk of a protocol, we start from the trust relation among the agents that are involved in that protocol. In general, an agent's trust in another depends on the particular service in question. An agent might trust a retailer for delivering furniture but may not trust him or her for actually assembling the furniture[3]. After an agent calculates a risk then it can decide whether the risk is acceptable or not for a current situation. The different risk ranges can be accepted on the various situations and agent characteristics.

*B. Benefit*

The benefit of using a particular protocol is considered here. The actual benefit that occurs if a particular protocol is used. The benefit of each protocol may vary depending on the particular agent. Each agent would want to maximize its utility, given that two protocols achieve the same goals. There may also be some protocols that are not acceptable to a given agent, i.e., if the utility is negative (i.e. the cost exceeds the benefit).The different agents may have different perspective and hence can gain different benefit by using a protocol as compared to other agent.

To calculate the overall cost (or utility) of a protocol we consider the overall set of propositions involved, and then sum their costs (or utilities). We consider all the propositions for which the agent is a debtor in the cost summation and all the propositions for which the agent is a creditor in the utility summation[2,3]. We then find the benefit as benefit(p) = utility(p) − cost(p)

Given the benefit of a protocol, we can then consider whether a protocol is acceptable to an agent. Basically, a protocol is acceptable if it has a benefit greater than zero.

*C. Other factors*

The ranking of the protocols is being done by using the above two factors that is the risk and benefit. In addition to these, various other factors can also be considered. The time factor may be considered in the ranking of the protocols. The different agents may take different time by different protocols to execute. The protocol should be chosen such that it takes the least time to execute by agent to achieve its objective. The problem can be that the time considered here is the estimated time, not the actual time to use a particular protocol, so the errors might occur in calculating the time. The time should be measured properly and the protocol with the least estimated time should be selected.

The selection can be based on a range of factors such as the number of the commitments required (heuristic: the fewer the better), or the strength of the argument for accepting a commitment. For example, one protocol may have a commitment which asks a merchant to ship goods should the customer pay (by any means), whereas another possible protocol may be asking the merchant to commit to shipping goods should the customer pay cash. It could be argued that the second commitment is more attractive to the merchant, and hence more likely to be accepted. The agents should be able to generate protocols at run time. For this the two algorithms have been developed already. One is goal based and the other is protocol based algorithm[3]. The goal based algorithm is based on divide and conquer strategy while the protocol based algorithm is based on depth first search strategy.

## IV. PROPOSED WORK

The agent can be considered as computer systems designed to attain a certain objective or a goal. The agent commitments develop a relationship between the two parties. The various commitment protocols are used by the agents. These can be defined at the design time by the designer or these can be generated at the run time by one of the agents. Now more than one protocol is generated at run time and only a single protocol can be used for execution. So in order to choose the best protocol, the ranking of the protocols is done. The ranking can be done based on various factors which have already been explained in paper. The risk and benefit factor is generally considered. But in future the various other factors may also be considered. The time factor can be included while ranking of the protocols. The different agents may take different time by different protocols to execute. The protocol should be chosen such that it takes the least time to execute by agent to achieve its objective. The other problem can be that the time considered here is the estimated time, not the actual time to use a particular protocol, so the errors might occur in calculating the time. The time should be measured properly and the protocol with the least estimated time should be selected. In future, the various other factors may also be considered for ranking of the protocols so that the best protocol can be chosen amongst all other protocols.

## V. CONCLUSION

The agent commitments develop a relationship between the two parties. The various commitment protocols are used by the agents. These can be defined at the design time by the designer or these can be generated at the run time by one of the agents. In order to choose the best

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 4, April 2015*

protocol, the ranking of the protocols is done. The risk and benefit factor is generally considered. The time may be considered in the ranking of the protocols. The different protocols may take different time by different agents to execute. The protocol should be chosen such that it takes the least time to execute by agent to achieve its objective. The other problem can be that the time considered here is the estimated time, not the actual time to use a particular protocol, so the errors might occur in calculating the time. The time should be measured properly and the protocol with the least estimated time should be selected. So, it is concluded that the best protocol should be chosen by considering various factors for ranking. Many other factors can also be used in future for ranking of the commitment protocols.

## REFERENCES

[1]. A. Günay, M. Winikoff, and P. Yolum. Commitment protocol generation. In 10th International Workshop on Declarative Agent Languages and Technologies (DALT), 2012
[2]. A. Günay, M. Winikoff and P. Yolum, Generating and Ranking Commitment Protocols, In Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1323-1324, Saint Paul, MN, US, 2013.
[3]. A. Günay, M. Winikoff and P. Yolum, Dynamically Generated Commitment Protocols in Open Systems, Journal of Autonomous Agents and Multiagent Systems(JAAMAS), Volume 29, Number 2, pages 192-229. March 2015
[4]. FornaraN, ColombettiM(2002), Operational specification of a commitment based agent communication language. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM Press, Melbourne, pp 535–542.
[5]. Chopra AK, Artikis A, Bentahar J, Colombetti M, Dignum F, FornaraN, Jones AJI, Singh MP, Yolum P (2013) Research directions in agent communication. ACMTransactions on Intelligent Systems and Technology (TIST).
[6]. FIPA (2000). Agent Communication Language. FIPA 2000 Specification. Foundation for Intelligent Physical Agents, http://www.fipa.org, 2000.
[7]. Artikis, A. (2009). Dynamic protocols for open agent systems. In Proceedings of the eighth international conference on autonomous agents and multiagent systems (AAMAS), (Vol. 1, pp. 97–104).
[8]. Mallya, A. U., & Singh, M. P. (2007). An algebra for commitment protocols. Autonomous Agents and Multi-Agent Systems, 14(2), 143–163.
[9]. Singh, M.P.: An Ontology for Commitments in Multiagent Systems. Artificial Intelligence and Law 7(1), 97–113 (1999).
[10]. M. Wooldridge. Semantic issues in the verification of agent communication. Autonomous Agents and Multi-Agent Systems, 3(1):9–31, 2000.
[11]. Gerhard W. Multiagent Systems A Modern Approach to Distributed Modern Approach to Artificial Intelligence , The MIT Press Cambridge, Massachusetts London, England
[12]. Verdicchio M. and Colombetti M. A logical model of social commitment for agent communication. In J.S. Rosonschein, T. Sandholm, M.Wooldridge editors, Proc. AAMAS 2003, pages 528-535, Melbourne,Australia. 2003.
[13]. Singh MP (1998) Agent communication languages: Rethinking the principles. IEEEComputer31(12):40-47.
[14]. Labrou Y. and Finin T., Semantics and conversations for an agent communication language. In Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97), Nagoya, Japan, 1997.
[15]. Chaib-draa, B., and Dignum, F. Trends in agent communication languages. In Computational Intelligence, Vol. 18(2) (2002) 1-14 .
[16]. Verdicchio, M., Colombetti, M.: A logical model of social commitment for agent communication. In Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM Press (2003)
[17]. Singh, M.P.: Multiagent systems as spheres of commitment. In Proceedings of the International Conference on Multiagent Systems (ICMAS), Workshop on Norms, Obligations, and Conventions. (1996).
[18]. F. Dignum and M. Greaves. Issues in Agent Communication, volume LNCS(1916). Springer-Verlag, 2002.
[19]. Ashok U. Mallya and Singh MP, An Algebra for Commitment Protocols Autonomous Agents and Multi-Agent Systems. volume 14, number 2, April 2007, pages 143–163.
[20]. Scott N.G and Singh MP, Protocol Refinement: Formalization and Verification, .Proceedings of the AAMAS Workshop on Agent Communication (AC). May 2010.