Interaction Between Agents (Arguing and Cooperating Agents)

[1]Ng Kee Seng [2]Abdul Hanan Abdullah [3]Abdul Manan Ahmad
[1]School of Computer Science and Information System
University Technology Malaysia, 81310 UTM, Skudai, Johor Darul Takzim, Malaysia
Tel: +60-7-520-2911, Fax: +60-7-520-2911, E-mail: alvinngutm@yahoo.co.uk
[2]School of Computer Science and Information System
University Technology Malaysia, 81310 UTM, Skudai, Johor Darul Takzim, Malaysia
Tel: +60-7-550-3502, Fax: +60-7-556-5044, E-mail: hanan@fsksm.utm.my
[3]School of Computer Science and Information System
University Technology Malaysia, 81310 UTM, Skudai, Johor Darul Takzim, Malaysia
Tel: +60-7-550-3502, Fax: +60-7-556-5044, E-mail: manan@fsksm.utm.my

Abstract: Interaction builds up among agents in order to accomplish their goals. Argumentation is important for agent negotiation and interaction. In this paper, we discuss about the framework for multi-agent argumentation and the way multi-agents co-operates between each other. We identify aspects of classical argumentation theory that are suitable and useful for artificial agents.

1. Introduction

In today's landscape of open telecom markets, telecom operators have started to play new roles. At the same time, completely new players are appearing on the market. The business interactions between those roles have become much more complicated than before. The liberalisation process has also resulted in a mixture of heterogeneous systems, networks and terminals. On the other hand, telecom companies still want to meet the high expectations of their customers, asking for on-demand, robust and personalised services.

Before we can argue why software agents can elegantly solve the above challenges, we need to say a few words about what we mean with 'software agents'. A lot of discussion material already exists on this topic, resulting in quite different definitions of agents. For example, 'an agent should speak an agent communication language', or 'an agent has beliefs, desires and intentions'. Clearly, the chosen definition depends on the background of the author. For the sake of this discussion, we will use a rather broad definition and define agents as being 'autonomous pieces of software, which behave in the interest of their users'.

In this work, we view conversations as the means of representing the conventions adopted by agents when interacting through the exchange of utterances; "utterance suggests human speech or some analogue to speech, in which the message between sender and addressee conveys information about the sender". More precisely, such conventions define the *legal* sequence of utterances that can be exchanged among the agents engaged in conversation: what can be said, to whom and when. Therefore, *conversation protocols* are coordination patterns that constrain the sequencing of utterances during a conversation.

2. Argumentation in Philosophy and Knowledge Representation

Researchers have been investigating how to automate argumentation. A problem is that many figures of arguments cannot be described formally. The Encyclopaedia Britannica lists for example the following figures:

- Semantically figures: Arguing by example, authority, or analogy,
- Syntactical figures: Arguing from the consequences, arguing from similar propositions, arguing from an accepted conclusion to an even more evident one, arguing from an accepted conclusion to the rejection of its contrary, undercut (Attacking premises), or rebut (Attacking conclusions)

The syntactical figures can be formally described by their form, i.e. syntax, and can therefore be easily automated. Although semantically figures such as arguing by authority may be formalised for particular domains, they are in general, not formalisable. This should not put us off, because it turns out that the syntactical figures of undercut and rebut are already sufficient to define the semantics of logic programs, which in turn makes logic programming the implementation language of choice for argumentation tools.

The relevance of an argument, i.e. should an agent accept it or immediately reject it, is an important issue in classical argumentation. Copi and Cohen list, for example, 17 fallacies of relevance of arguments, only three of which can be expressed formally:

- An *argument from ignorance* argues that a proposition is true simply on the basis that it has not been proved false, or that it is false because it has not been proved true.
- *Begging the question* also called *circular argumentation* assumes the truth of what it seeks to prove in the effort to prove it.
- *Division* assumes that what is true as a whole must be true in its parts and vice versa.

Interestingly, these three examples of fallacies involve all non-monotonic reasoning and require two kinds of negation:

- Implicit negation *not a* to express the lack of evidence for *a*
- Explicit negation *–a* to state that there is an explicit proof for *-a*.
- The two negations are related in that *-a* implies *not a*

With the two kinds of negation we can express the three fallacies mentioned above:

- Arguments from Ignorance have the form *a <- not -a* or *-a <- not a*
- Begging the question has the form of a positive loop, e.g. *a <- a* or *a <- not a* in its most reduced form.
- Division requires non-monotonic reasoning (NMR) and contradiction removal. A typical example dealt with extensively in the NMR literature is a contradiction between flying birds and not-flying penguins.

3. The Argumentation Framework

There are two fundamental types of interaction for multiple agents: they cooperate and argue. An agent, which does not know anything about a certain literal, cooperates with others, which help out and possibly provide the knowledge. As for argumentation, an agent believes in something and argues with other agents to determine whether this belief is valid or has to be revised. When arguing we can distinguish sceptical and credulous agents. The former is more critical towards its own beliefs than the latter. Technically, sceptical agents accept undercuts and rebuts to their own arguments, whereas credulous agents accept only undercuts.

To reduce the number of messages exchanged an agent will ask only its co-operation partners for help and only those whose domain of expertise covers the issue in question. Similarly, an agent proposes its beliefs only to its argumentation partners with the corresponding domain of expertise. All in all, an agents is defined by

- a set of arguments
- a set of predicate names defining the agent's domain of expertise
- a flag indication whether the agent is credulous or sceptical
- a set of cooperation partners
- a set of argumentation partners

4. Conversation Protocol

A conversation protocol (CP) defines a class of legal sequences of utterances that can be exchanged between two agents holding a conversation. We model and implement a CP as a special type of Pushdown Transducer (PDT), which can be seen in turn as a combination of a Finite-State Transducer (FST) and a Pushdown Automaton (PDA):

- An FST is simply a Finite State Automaton (FSA) that deals with two tapes. To specify an FST, it suffices to augment the FSA notation so that labels on arcs can denote pairs of symbols;
- A PDA is composed of an input stream and a control mechanism like an FSA along with a stack on which data can be stored for later recall.

Therefore, a PDT is essentially a pushdown automaton that deals with two tapes. A PDA can be associated to a PDT by considering the pairs of symbols on the arcs as symbols of a PDA. The choice of PDTs as the mechanism for modeling CPs is motivated by several reasons. First, analogously to other finite-state devices a few fundamental theoretical basis make PDTs very flexible, powerful and efficient. They have been largely used in a variety of domains such as pattern matching, speech recognition, cryptographic techniques, data compression techniques, operating system verification, etc.

They offer a straightforward mapping from specification to implementation. The use of pairs of symbols allows to label arcs with (*p*/*d*) pairs (where *p* stands for a performative, and *d* stands for a predicate). This adds expressiveness to the representation of agent conversations. PDTs, unlike other finite state devices, allow us to store, and subsequently retrieve, the contextual information of ongoing conversations.

5. Overview of Interagents

Interagents mediate the interaction between an agent and the agent society wherein this is situated. The main task of an interagent is the management of conversation protocols. Here we differentiate two roles for the agents interacting with an interagent: *customer*, played by the agent exploiting and benefiting from the services offered by the interagent; and *owner*, played by the agent endowed with the capability of dynamically establishing the policies that determine the interagent's behaviour. Needless to say that an agent can possibly play both roles at the same time. Moreover, several owners can even share their property (*collective ownership*), whereas several customers can make use of the same interagent (*collective leasing*).

In what follows, we provide a detailed account of the full functionality of interagents, mostly from the point of view of customers to illustrate how they undertake conversation management. An interagent is responsible for posting utterances of its customer to the corresponding addressee and for collecting the utterances that other agents address to its customer. This *utterance management* abstracts customers from the details concerning the agent communication language and the network protocol. Each interagent owns a collection of relevant conversation protocols (CP) used for managing its customer conversations. When its customer intends to start a new conversation with another agent the interagent instantiates the corresponding conversation protocol. Once the conversation starts, the interagent becomes responsible for ensuring that the exchange of utterances conforms to the CP specification.

Before setting up, any conversation the interagent must perform a *CP negotiation* process with the interagent of the addressee agent. The goal of CP negotiation is to reach an agreement with respect to the conversation protocol to be used. Moreover, before starting a conversation, the interagent performs a *CP verification* process. This process checks whether the CP to be used verifies the necessary conditions (liveliness, termination, deadlock and race condition free) for guaranteeing the correct evolution of an interaction. Finally, an interagent allows its customer to hold several conversations at the same time. This capability for *multiple conversations* is important because, although in the paper we consider only conversations with two participants (dialogues), conversations with any number of participants are built as a collection of simultaneous CP instances. In other words, the agent views a conversation as involving *n* participants while its interagent views such conversation as a collection of simultaneous dialogues represented as multiple CP instances.

5.1 Example

Next, we introduce an example of multi-agent conversation, an agent-mediated electronic market that will serve to illustrate both the use and functionality of interagents. Brokerage system is an electronic auction house based on the traditional night market auctions in which trading (buyer and seller) heterogeneous (human and software) agents can trade. The main activity within the system is the auctioning of goods. When the auctioneer opens a new *bidding round* to auction a good among a group of agents, he starts quoting offers downward from the chosen good's starting price.

For each price called, three situations might arise during the open round: i) several buyers submit their bids at the current price. In this case, a collision comes about, the good is not sold to any buyer, and the auctioneer restarts the round at a higher price; ii) only one buyer submits a bid at the current price. The good is sold to this buyer whenever his credit can support his bid. Whenever there is an unsupported bid the round is restarted by the auctioneer at a higher price, the unsuccessful bidder is punished with a fine, and he is expelled out from the auction room unless such fine is paid off; and iii) no buyer submits a bid at the current price.

If the reserve price has not been reached yet, the auctioneer quotes a new price which is obtained by decreasing the current price according to the price step. If the reserve price is reached, the auctioneer declares the good *withdrawn* and closes the round.

The conventions that buyer agents have to comply with when interacting with the auctioneer during a bidding round are represented by means of a conversation protocol managed by the so-called trading interagents. These are owned by the institution, the auction house, but used by trading agents.

6. Conclusion

Interaction in between agents in a multi-agents system environment involves complex or simple argumentations and co-operates among each other to accomplish each others goals.

It involves and interagent where the conversation among the agents are based on the discussed conversation protocol earlier. Each agent has it own set of arguments and are able to identify its co-operation and argumentation partners.

To understand better and the get the feel of the above framework, try developing an argumentation and co-operating scenarios on your own based on the framework.

References

[1] Michael Schroder, *An efficient argumentation framework for negotiating autonomous agents,* In Proceedings of the *workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW99*. Valencia, Spain, Springer-Verlag, July 1999

[2] Michael Schroeder, Daniela A. Plewe, and Andreas Raab, *Ultima Ratio - Should Hamlet kill Claudius?,* In Proceedings of the *second International Conference on Autonomous Agents AA98*, Minneapolis, USA, May 1998. ACM Press.

[3] Iara Mora, Jose Julio Alferes, and Michael Schroeder, *Argumentation and Cooperation for Distributed Extended Logic Programs,* In *Working notes of the workshop on Non-monotonic Reasoning*, Trento, Italy, May 1998.

[4] Iara Mora, Jose Julio Alferes, and Michael Schroeder, *Argumentation for Distributed Extended Logic Programs,* Proceedings of the *1st International Workshop on Logic Programming and Multi-Agents*, Leuven, July 1997.

[5] Michael Schroeder, Iara Mora, and Jose Julio Alferes. *Vivid Agents arguing about Distributed Extended Logic Programs*, Proceedings of the *Portuguese Conference on Artificial Intelligence* EPIA97, Coimbra, Oct 1997, LNAI 1323, Springer-Verlag.

[6] Francisco J. Martín , Enric Plaza, Juan A. Rodríguez-Aguilar, *Composing Multi-Agent Systems with Interagents*, In SEID'99*Proceedings of the Simposio Español* de Informática Distribuida , Santiago de Compostela, Spain, February 24-26, 1999

[7] Francisco J. Martín, Enric Plaza, Juan A. Rodriguez-Aguilar, *An Infrastructure for Agent-Based Systems: an Interagent Approach,* In IJIS *International Journal of Intelligent Systems.* 1999

[8] T. Kawamura, N. Kase, D. Araki, A. Ohsuga, *"Development of a Distributed Cooperative Scheduling System based on Negotiations between Scheduling Agents"*, Systems and Computers in Japan, John Wiley & Sons, Inc., Vol. 31, No. 1, pp. 92-101.

[9] Einsiedler, H.J., Leger, A. Gleizes, M-P. *ABROSE: A cooperative multiagent-based framework for the e-marketplace.* Agents Technology in Europe. ACTS Activities. Infowin '99.

[10] T. Kawamura, S. Joseph, A. Ohsuga, S. Honiden, 2001. *"Designing Multi-Agent Systems based on Pairwise Agent Interactions"*, IEICE Trans. Inf. & Syst.