

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/64709>

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

A Dialogue Game Approach to Multi-Agent System Programming

Henk-Jan Lebbink ^{abc} Cilia Witteman ^b
John-Jules Meyer ^a

^a Institute of Information and Computing Sciences, Utrecht University

^b Diagnostic Decision Making, Faculty of Social Sciences, University of Nijmegen

^c Emotional Brain research and consultancy, Almere

henkjan@cs.uu.nl

Abstract

This paper approaches multi-agent system programming with dialogue games allowing the semantics of communicative acts to be a component in multi-agent architectures. We present a dialogue game for enquiry enabling agents to answer questions in a distributed fashion. In addition, we propose a reasoning game that defines when agents are allowed to make decisions, in the current case, decisions to accept to believe propositions. These games are brought together in a deliberation cycle and are implemented in Prolog.

1 Introduction

This paper proposes a programming approach for multi-agent system (MAS) development in which distributed problem solving emerges from communication and reasoning protocols. This approach provides relief for developers from developing questioning, inquiring and answering processes allowing them, for example, to focus on knowledge acquisition and representation issues.

FIPA proposed ACLs in which the semantics of communicative acts are specified with pre and post-conditions. Our semantics is similar; however, with the difference that the post-conditions define how the listener *perceives* the speaker, and not what the actual state of the speaker is. Other approaches include goal directed agent programming in 3APL; our approach does not use goals but instead unbalanced cognitive states to motivate communication. We contribute to the MAS development, reasoning and communication protocols.

In Section 2, the agent's cognitive state (CS) is given and a deliberation cycle is provided stating when communication and reasoning is to be done. Next, dialogue games and reasoning games are presented that define, given the agent's CS, whether communicative acts are allowed to be uttered (Section 3) or decisions to be made (Section 4). In the current paper, communicative acts are about question handling, and decisions are about deciding to believe propositions. These games are to a certain extent independent of the agents' specification. In Section 5, conclusions and a sketch of the implementation are given.

2 Agent Programming with Games

2.1 The Agent's Cognitive State

An agent's cognitive state (CS) consists of a finite number of mental states, which are theories of multi-valued logic¹. For our current needs it is sufficient to know that theories are sets of propositions; see [3] for a formal specification. We will not present a full repertoire of all possible mental states agents have regarding themselves and others; only those are identified that are used in the present paper. In the remainder, set \mathcal{A} denotes the set of agent identifiers.

Agent x 's private belief state is denoted B_x ; $\psi \in B_x$ states that x believes proposition ψ . An agent x 's private desire to believe $D_x B_x$ is the set of propositions that x desires to believe; $\psi \in D_x B_x$ states that x desires that it believes ψ . $B_x B_y$ is the set of manifested beliefs of y that x is aware of; $\psi \in B_x B_y$ states that x is aware that y believes ψ . An agent can be aware of other agents' desires; $\psi \in B_x D_y B_y$ states that x is aware that y desires to believe ψ . Manifested ignorance state $B_x I_y$ is the set of propositions that x is aware that y does not believe; $\psi \in B_x I_y$ states that x is aware that y does not believe ψ . In addition, higher-order manifested mental states are defined likewise, e.g. $B_x B_y B_x$, $B_x B_y I_x$; for manifested desires we use $B_x B_y D_x B_x$ and $B_x B_y B_x D_y B_y$.

2.2 Deliberation Cycle

An agent consists of its CS and a deliberation cycle that describes when changes are made to its CS. The deliberation cycle contains three choices and the execution of five rules. See Figure 1 for a graphical depiction.

Step 1. Choose an applicable reasoning rule and go to step 2 to execute this rule. If there are no applicable rules, go to step 4 to check whether communicative acts have been received.

Step 2. Execute the selected reasoning rule from step 1. Note that execution of reasoning rules does not have observable effects for other agents. Go to step 3 to update the agent's CS accordingly.

Step 3. Execute the appropriate update rule for the selected reasoning rule from step 2. Go to step 1 to check whether more reasoning can be done.

Step 4. Check whether communicative acts are received, that is, acts that are directed to the agent, take the oldest act from the queue of received acts. Go to step 5 to update the CS accordingly. If the queue of received acts is empty, go to step 6 to check whether communication is allowed.

Step 5. Execute the appropriate update rule for the selected communicative act from step 4. Go to step 1 to check whether reasoning can be done.

Step 6. Choose an applicable dialogue rule and go to step 7 to execute this rule. If there are no applicable rules, go to step 4 to check whether communicative acts have been received.

¹A multi-valued logic allows to represent a lack of belief, partial belief, and inconsistent belief; these states are used in our dialogue games and therefore need to be represented explicitly.

Step 7. Execute the selected dialogue rule from step 6. Execution has the effect of uttering a communicative act directed at some other agent. Go to step 8 to update the CS accordingly.

Step 8. Execute the appropriate update rule for the uttered communicative act from step 7. Go to step 1 to check whether reasoning can be done.

The deliberation cycle combined with reasoning and dialogue games result in a domain independent multi-agent system. The system is independent of the agent’s actual beliefs and desires, as well as the agents’ use of decision and communication protocols. Knowledge can be added to a MAS in the form of agents, and in a similar fashion reasoning and communication protocols can be added as components to agents.

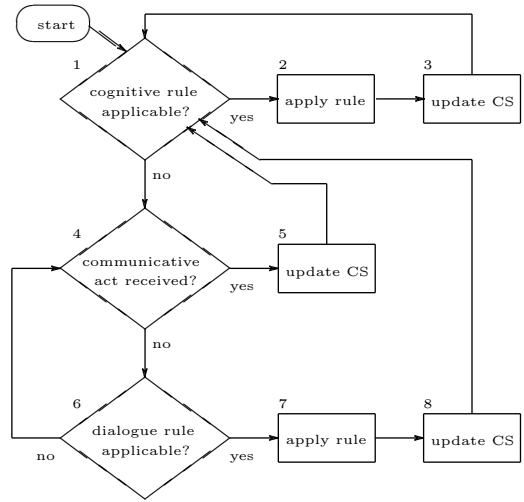


Figure 1: The agent’s deliberation cycle.

3 Dialogue Game for Handling Questions

Two agents participate in the following games: unless states otherwise, Sarah is the speaker and John the listener; denoted with variables s and j respectively.

3.1 Posing Questions

An agent is in the state of being motivated to utter a question when she has an unbalanced belief and desire state, cf. [1]. Stated differently, if Sarah desires to believe a proposition ψ , and she does not yet believe ψ , then she has a motive to utter a question to John whether she may add ψ to her belief state, because answers to the question may balance her belief and desire state. This communicative act is called a question for belief addition (qba for short). These two criteria are part of the pre-conditions of the qba .

$$(\psi \in D_s B_s), (\psi \notin B_s) \in pre(qba(s, j, \psi))$$

The Gricean maxims [2] are principles of cooperative dialogue and provide restrictions for uttering acts. These maxims state that utterances of communicative acts should be informative: i.c. Sarah may not already be aware of John’s answer, that is, she may ask for ψ if she does not believe that John believes ψ , and that

she does not believe that John is ignorant of ψ .

$$(\psi \notin B_s B_j), (\psi \notin B_s I_j) \in pre(qba(s, j, \psi))$$

Given the motivation to utter a $qba(s, j, \psi)$, listener John can derive properties of Sarah's CS: he may derive that Sarah has the desire to believe ψ , and that she does not believe ψ ; John's CS should change yielding the following post-conditions.

$$(\psi \in B_j D_s B_s), (\psi \in B_j I_s) \in post(qba(s, j, \psi))$$

In addition, John is aware that Sarah is not aware of his answer; however, these post-conditions are not used in the dialogue game.

The speaker may assume that for a listener equal post-conditions hold. Consequently, after uttering a $qba(s, j, \psi)$, Sarah may derive that John is aware that she desires to believe ψ , and that she may be aware that John is aware that she does not believe ψ . After the utterance, Sarah's CS has changed according to the following post-conditions.

$$(\psi \in B_s B_j D_s B_s), (\psi \in B_s B_j I_s) \in post(qba(s, j, \psi))$$

Communicative acts should be informative, i.e. agents may not utter communicative acts more than once. To realize this restriction, at least one of the previous post-condition must *not* hold. A criterion is added to the set of pre-conditions to restrict situations in which a question may be uttered.

$$(\psi \notin B_s B_j D_s B_s) \in pre(qba(s, j, \psi))$$

3.2 Affirmative Answers to Questions

Apart from giving restrictions, Gricean maxims provide motivations to answer questions: a question should be answered either affirmative or negative. Sarah is motivated to respond affirmative to a question from John regarding ψ , if Sarah believes that John has the desire to believe ψ , and Sarah believes ψ . $gqba$ is short for granting a question for belief addition.

$$(\psi \in B_s D_j B_j), (\psi \in B_s) \in pre(gqba(s, j, \psi))$$

Given the motivation for an affirmative response, listener John may derive properties of Sarah's CS. If a $gqba(s, j, \psi)$ is uttered, John may deduce that Sarah believes ψ , and that Sarah is aware that John desires to believe ψ . Based on the post-conditions for John's CS, Sarah's post-conditions are given next.

$$\begin{aligned} &(\psi \in B_j B_s), (\psi \in B_j B_s D_j B_j), \\ &(\psi \in B_s B_j B_s), (\psi \in B_s B_j B_s D_j B_j) \in post(gqba(s, j, \psi)) \end{aligned}$$

To prevent the $gqba$ from being superfluous, Sarah may not be aware she uttered the act before; she can be sure about this if at least one of her previous post-conditions does not hold.

$$(\psi \notin B_s B_j B_s) \in pre(gqba(s, j, \psi))$$

3.3 Posing Counter-Questions

Counter-questions are questions about propositions that are related to other propositions that agents desire to believe. These questions are syntactically indistinguishable from the question defined in Section 3.1. However, counter-questions are different communicative acts from a semantic perspective: they have a different motivation; nevertheless, post-conditions and other pre-conditions are no different from the ordinary question. To differentiate between questions, counter-questions are indexed 2 and 3.

Sarah is motivated to utter counter-question 2 regarding ψ , if she desires to believe φ , she does not believe φ , and adding ψ to her belief base results in believing φ . Addition to a belief state is adding a proposition set theoretically and taking the closure (Cn) under a set of deduction rules², for details, see [3].

$$(\varphi \in D_s B_s), (\varphi \notin B_s), (\varphi \in Cn(B_s \cup \{\psi\})) \in pre(qba_2(s, j, \psi))$$

Sarah is motivated to utter a counter-question 3 regarding ψ , if she is aware that John desires to believe φ , she is not aware that John believes φ , and she does not herself believe φ , however, adding ψ to her belief state results in believing φ , then we necessarily have $\psi \notin B_s$.

$$(\varphi \in B_s D_j B_j), (\varphi \notin B_s B_j), (\varphi \notin B_s), (\varphi \in Cn(B_s \cup \{\psi\})) \in pre(qba_3(s, j, \psi))$$

Note that only motivations are presented and that the other pre-conditions of a qba are also applicable. Also note that these counter-questions create new enquiry dialogues that are related to existing dialogues; initial unbalanced belief-desire pairs introduce new unbalanced pairs which results in distributed enquiry yielding a distributed solution.

3.4 Negative Answers to Questions

Sarah is motivated to utter a negative response to a question from John regarding ψ , if she believes that John has the desire to believe ψ , she does not believe ψ , and she ran out of options i.e. counter-questions to help him. The formal treatment of the last pre-condition is left out due to lack of space. $dqba$ is short for denying a question for belief addition.

$$(\psi \in B_s D_j B_j), (\psi \notin B_s) \in pre(dqba(s, j, \psi))$$

Listener John may derive properties of Sarah's CS from the motivations of a $dqba(x, y, \psi)$. Sarah may derive properties of John's CS.

$$(\psi \in B_j B_s D_j B_j), (\psi \in B_j I_s), (\psi \in B_s B_j B_s D_j B_j), (\psi \in B_s B_j I_s) \in post(dqba(s, j, \psi))$$

To prevent that the $dqba$ is superfluous, at least one of Sarah's post-conditions must not hold.

$$(\psi \notin B_s B_j I_s) \in pre(dqba(s, j, \psi))$$

²We only use R3: If $\psi \rightarrow \varphi \in B_s$ and $\psi \in B_s$ then also $\varphi \in B_s$.

3.5 Dialogue and Update Rules

Dialogue rules prescribe which communicative acts agents are allowed to utter given their CS's. A dialogue rule for a communicative act λ states that if all criteria in the set of pre-conditions of λ hold according to the agent's CS, then the rule may be executed by the agent, that is, λ may be uttered. Update rules define the contents of an agent's CS after λ is uttered or received. Update rules states that if λ is received by an agent, then the post-conditions of λ hold for that agent. Dialogue games consist of these rules.

We analysed dialogue games for desirable properties such as termination, confluence, or whether unbalanced belief-desire states are resolved in the terminating CS's. The dialogue game for handling questions is proven to be terminating, and confluent for belief states with term rewriting systems [5] (proofs not presented). Termination property is essential in the deliberation cycle: if it is violated decisions may never be made and communicative acts may never be uttered due to eternal cycling. Confluence makes the choice of dialogue rule in step 1 unimportant from a semantic point of view which allows straightforward implementation.

3.6 Other Communication Protocols

Another communication protocols concern offering of beliefs, i.e. requesting agents to add propositions to their belief states. Offering propositions has the intended effect of the propositions being part of the listeners' belief states, opposed to questions which have the intended effect of propositions being part of the speakers' belief states. The motivation to utter an offering of belief addition (*oba* for short) is given next, the dialogue game is elaborated in Lebbink *et al.* [4].

$$(\psi \in D_s B_j), (\psi \notin B_s B_j) \in pre(oba(s, j, \psi))$$

Protocols for belief retraction can be given in a similar fashion: a question for belief retraction (*qbr* for short) directed at John is motivated when Sarah has the desire to be ignorant about ψ and she believes ψ .

$$(\psi \in D_s I_s), (\psi \in B_s) \in pre(qbr(s, j, \psi))$$

A request for a belief retraction (*rbr* for short) directed at John is motivated when Sarah has the desire that John is to be ignorant about ψ , and she is not aware that John is ignorant about ψ .

$$(\psi \in D_s I_j), (\psi \notin B_s I_j) \in pre(rbr(s, j, \psi))$$

Special care must be taken that introducing other games does not break termination properties of existing games.

4 Reasoning Game for Believing

Our objective is to propose a programming approach to MAS development with dialogue games. These dialogue games presuppose reasoning rules defining when agents are allowed to accept to believe propositions, i.c. when questions for belief addition are answered affirmative.

4.1 Protocol for Deciding to Believe Propositions

The agent’s reasoning processes are based on her private beliefs, desires and her beliefs of other agents’ mental states. Similar to the semantics of communicative acts, the agents’ decisions to add propositions to their belief states are sets of criteria that need to hold in their current CS’s. We use a simple conformism protocol. Sarah is allowed to add ψ to her belief state, if she believes that John believes ψ , and she does not believe ψ itself. *ba* is short for belief addition.

$$(\psi \in B_s B_j), (\psi \notin B_s) \in pre(ba(s, \psi))$$

The effects of the *ba* is that Sarah believes the propositions.

$$(\psi \in B_s) \in post(ba(s, \psi))$$

Other criteria may restrict the *ba* to add only those beliefs that have a purpose, i.e. contribute to balance belief-desire pairs, or add only those beliefs coming from an authority, like users or sensors, but this is beyond the scope of the paper.

Reasoning rules and associated update rules are defined like dialogue rules. Similar issues for termination and confluence hold for reasoning games.

4.2 Other Reasoning Protocols

Other reasoning protocols may concern agents becoming aware that they have irresolvable disagreements and that they want to agree to disagree [4]. Yet another protocol may define when agents are allowed to forget beliefs, i.e. remove propositions from their belief state. *br* is short for belief retraction.

$$(\psi \in B_s), (\forall j \in \mathcal{A} (\psi \in B_s I_j)) \in pre(br(s, \psi))$$

This protocol will interfere with the protocol for belief addition; future research centres on the question what a protocol for retraction should look like that if it is to work with a protocol for belief addition. Agents performing belief revision in a distributed fashion could be implemented with these reasoning protocols.

5 Implementation and Conclusions

The agents’ CS’s, dialogue and reasoning games, and the deliberation cycle are implemented in SWI-Prolog [6] resulting in a multi-agent implementation. For every mental state a separate Prolog engine with database is created that captures the state of an agent. The database is used to store the propositions part of the mental state. Engines with databases are called ‘modules’ in SWI-Prolog and are used in the following fashion: e.g. $mc(b(s))$ is the identifier for Sarah’s belief state. Reasoning and dialogue rules are implemented by taking the pre-conditions of decisions and communicative acts as the body of Prolog clause. Update rules for listener and speaker are the actions of asserting propositions if they were not yet present in the modules. The dialogue rule to utter a *qba*, and the corresponding

update rules are implemented in Figure 2. Reasoning rules are implemented in analogous fashion. Preliminary tests have been performed with up to 40 agents resulting in good performance.

Dialogue and reasoning games are implemented enabling agents to communicate and make decisions. The semantics of communicative acts and decisions are defined by formulating the rules of usage, being pre-conditions that need to hold in the speaker's or reasoning agent's CS, and post-conditions that need to hold when communicative acts are uttered or decisions made. The agent's ability to utter counter-questions results in distributed enquiry and a distributed answer. Different dialogue games and reasoning games can be deployed for different tasks, like problem solving or belief revision. Confluence of the dialogue game for question handling makes the order of permissible communicative act utterance unimportant. And termination property states that answers to questions (if any) are found. Tests resulted in good performance providing an indication in the feasibility of the programming approach.

```

dialogue_rule(qba(S, J, Prop)) :-
    mc(d(S), b(S)):Prop,
    mc(b(S)):not Prop,
    mc(b(S), b(J)):not Prop,
    mc(b(S), i(J)):not Prop,
    mc(b(S), b(J), d(S), b(S)):not Prop.

update_listener(qba(S, J, Prop)) :-
    add(mc(b(J), d(S), b(S)), Prop),
    add(mc(b(J), i(S), b(J)), Prop),

update_speaker(qba(S, J, Prop)) :-
    add(mc(b(S), b(J), d(S), b(S)), Prop),
    add(mc(b(S), b(J), i(S)), Prop).

add(Prop, MC) :-
    MC:not Prop -> MC:assert(Prop); true.

```

Figure 2: Prolog listings.

References

- [1] Robbert-Jan Beun. On the Generation of Coherent Dialogue: A Computational Approach. *Pragmatics & Cognition*, 9(1):37–68, 2001.
- [2] H. P. Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Speech Acts*, volume 11 of *Syntax and Semantics*, pages 41–58. Academic Press, New York, USA, 1975.
- [3] Henk-Jan Lebbink, Cilia Witteman, and John-Jules Meyer. Dialogue Games for Inconsistent and Biased Information. *Electronic Lecture Notes of Theoretical Computer Science*, 52(2), 2003.
- [4] Henk-Jan Lebbink, Cilia Witteman, and John-Jules Meyer. A dialogue game to offer an agreement to disagree. In *Third international joint conference on Autonomous Agents & Multi-Agent Systems (AAMAS'04)*, pages 1238–1239, New York, NY, USA, July 19–23 2004.
- [5] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
- [6] Jan Wielemaker. Swi-prolog home page. <http://www.swi-prolog.org/>.