On the Formal Semantics of Theory of Mind in Agent Communication

Alison R. Panisson[1(✉)], Ștefan Sarkadi[2], Peter McBurney[2], Simon Parsons[2], and Rafael H. Bordini[1]

[1] School of Technology, PUCRS, Porto Alegre, Brazil
alison.panisson@acad.pucrs.br, rafael.bordini@pucrs.br
[2] Department of Informatics, King's College London, London, UK
{stefan.sarkadi,peter.mcburney,simon.parsons}@kcl.ac.uk

Abstract. Recent studies have shown that applying Theory of Mind to agent technologies enables agents to model and reason about other agents' minds, making them more efficient than agents that do not have this ability or agents that have a more limited ability of modelling others' minds. Apart from the interesting results of combining Theory of Mind and agent technologies, an important premise has not been yet fully investigated in the AI literature: how do agents acquire and update their models of others' minds? In the context of multi-agent systems, one of the most natural ways in which agents can acquire models of other agents' mental attitudes is through communication. In this work, we propose an operational semantics for agents to update Theory of Mind through communication. We not only make our formalisation broadly applicable by defining a formal semantics based on components from the BDI architecture, but we also implement our approach in an agent-oriented programming language that is based on that architecture.

Keywords: Multi-Agent Systems · Theory of Mind · Agent-Oriented Programming Languages

1 Introduction

It seems reasonable to assume that agents will be more effective at achieving their goals during interactions if they understand the other entities involved. Understanding others requires the capability of modelling and reasoning about other agents' minds. These characteristics are intrinsic to Theory of Mind (ToM) [10]. ToM is the ability of humans to ascribe elements such as beliefs, desires, and intentions, and relations between these elements to other human agents. In other words, it is the ability to form mental models of other agents.

The Multi-Agent Systems (MAS) community is showing increased interest in ToM [6,7,24]. One reason for this interest might be that ToM could boost the quality of communication between agents that need to exchange information in order to make decisions and reach meaningful agreements. By meaningful

© Springer Nature Switzerland AG 2019
M. Lujak (Ed.): AT 2018, LNAI 11327, pp. 18–32, 2019.
https://doi.org/10.1007/978-3-030-17294-7_2

agreements we mean agreements that result from a mutual understanding. We consider mutual understanding to be represented by a certain set of shared beliefs reached through communication.

Various studies have investigated the use of ToM in MAS. Among them, [6,7] investigated the advantages of using different levels of ToM in games played by agents, and [1,11,12,20,29], even though ToM is not mentioned, show the advantages of modelling the opponent when considering strategies in argumentation-based dialogues. All that work shows that modelling other agents' minds is an important topic of research, and the results are important contributions to the MAS literature. However, as described in [32], most of the work on modelling other agents' minds assume ToM as given. This is an understandable assumption, but it is nevertheless unrealistic given that there are no readily-available, practical techniques for developing such agents. Also, as a result of relying on such unrealistic assumption, the question of how agents acquire the model of other agents' minds has not been fully investigated. In this work, we propose a formal semantics for updates that agents can effect to their ToM based on the communication that they have with other agents, thus allowing them to acquire a ToM.

Communication plays an important role in MAS [34], and takes places on multiple levels. Communicating content is only one part of the process of communication. It also includes forming the message in a way that will make the sender's purpose of communication clear to the receivers [8]. In order to make the sender's purpose clear, agent communication languages, such as FIPA-ACL [9] and KQML [8], have been proposed based on speech act theory. Both languages format message to include *performatives* in such a way that the sender's purpose will be clear to the agent that is receiving the communication, facilitating the correct interpretation of the content of that communication. In this work we show that, based on the semantics of the agent communication languages, agents are able to infer the likely model of other agents' minds, i.e., ToM, considering the meaning of each communication exchange. Using ToM acquired from communication, agents are able to reason and make decisions using other agents' models.

The main contributions of this paper are: (i) an operational semantics, formally defined, for updates that agents carry out on their ToM during communication—to the best of our knowledge, our work is the first to propose a formal model of how agents acquire and update ToM during communication in multi-agent systems, particularly in the practical context of an Agent-Oriented Programming Language (AOPL) based on the BDI architecture; (ii) an approach for agent reasoning and decision making, and, in particular, we show how agents can reach shared beliefs more efficiently than when they are not able to model ToM.

2 Background

2.1 Agent Communication Languages

Agent communication languages have been developed based on speech act theory [30]. Speech act theory is concerned with the role of language as actions. In speech act theory, a speech act is composed by (i) a *locution*, which represents the physical utterance; (ii) an *illocution*, which provides the speaker intentions to the hearer; and (iii) the *perlocution*, which describes the actions that occur as a result of the illocution. For example, "*I order you to shut the door*" is a *locution* with an *illocution* of a command to shut the door, and the *perlocution* may be that the hearer shuts the door. Thus, an illocution is considered to have two parts, the illocutionary force and a proposition (content). The illocutionary force describes the type speech act used, e.g., *assertive, directive, commissive, declarative, expressive.*

Among the agent communication languages which emerged based on speech act theory, FIPA-ACL [9] and KQML [8] are the best known. In this work, for practical reasons, we choose KQML, which is the standard communication language in the Jason Platform [3], the multi-agent platform we choose to implement this work.

The Knowledge Query and Manipulation Language (KQML) was designed to support interaction among intelligent software agents, describing the message format and message-handling protocol to support run-time agent communication [8,17]. In order to make KQML broadly applicable, in [16] a semantic framework for KQML was proposed. Considering the speech act semantics, they argue that it is necessary to consider the cognitive state of the agents that use these speech acts. Defining the semantics, the authors provided an unambiguous interpretation of (i) how the agents' states change after sending and/or receiving a KQML performative, as well as (ii) the criteria under which the illocutionary point of the performative is satisfied (i.e., the communication was effective).

2.2 Agent Oriented Programming Languages

Among the many AOPLs and platforms, such as Jason, Jadex, Jack, AgentFactory, 2APL, GOAL, Golog, and MetateM, as discussed in [2], we chose the Jason platform [3] for our work. Jason extends the AgentSpeak language, an abstract logic-based AOPL introduced by Rao [28], which is one of the best-known languages inspired by the BDI architecture.

Besides specifying BDI agents with well-defined mental attitudes, the Jason platform [3] has some other features that are particularly interesting for our work, for example, strong negation, belief annotations, and (customisable) speech-act based communication. Strong negation helps the modelling of uncertainty, allowing the representation of things that the agent: (i) believes to be true, e.g., $\texttt{about}(\texttt{paper1}, \texttt{tom})$; (ii) believes to be false, e.g., $\neg\texttt{about}(\texttt{paper2}, \texttt{tom})$; (iii) is ignorant about, i.e., the agent has no information about whether a paper is about `tom` or not. Also, Jason automatically generates annotations for all the beliefs

in the agents' belief base about the source from where the belief was obtained (which can be from sensing the environment, communication with other agents, or a mental note created by the agent itself). The annotation has the following format: $\texttt{about}(\texttt{paper1}, \texttt{tom})[\texttt{source}(\texttt{reviewer1})]$, stating that the source of the belief that `paper1` is about the topic `tom` is `reviewer1`. The annotations in Jason can be easily extended to include other meta-information, for example, trust and time as used in [19,21]. Another interesting feature of Jason is the communication between agents, which is done through a predefined (internal) action. There are a number of performatives allowing rich communication between agents in Jason, as explained in detail in [3]. Furthermore, new performatives can be easily defined (or redefined) in order to give special meaning to them[1], which is an essential characteristic for this work.

3 Running Example

As a running example, we will consider a scenario with five agents in a university. The first agent, named *John*, plays the role of a professor in the university, and the other agents, named *Bob, Alice, Nick*, and *Ted*, play the role of students. *John* has a relation of *supervisor* to the *students*. Also, *John* is responsible for distributing some tasks to the students. In order to distribute the tasks, *John* maintains information about the students, so as to distribute tasks to students that have the required knowledge for each task.

Our model can be described as $\langle Ag, \mathcal{T}, \mathcal{A}, \mathcal{S} \rangle$, in which Ag represents the set of agents, $\mathcal{T}$ the set of tasks of the kind $\mathcal{T} \subseteq \mathcal{A} \times \mathcal{S}$, representing an action from $\mathcal{A}$, requiring knowledge about a subset of subjects from $\mathcal{S}$, that might be executed to achieve the task $\mathcal{T}$. In our example, we consider the following actions, subjects, and tasks:

- $\mathcal{A} = \{\texttt{write\_paper}, \texttt{review\_paper}, \texttt{paper\_seminar}\}$
- $\mathcal{S} = \{\texttt{mas}, \texttt{kr}, \texttt{tom}\}$
- $\mathcal{T} = \left\{ \begin{array}{l} \texttt{task}(\texttt{write\_paper}, [\texttt{mas}, \texttt{tom}]) \\ \texttt{task}(\texttt{review\_paper}, [\texttt{kr}]) \\ \texttt{task}(\texttt{paper\_seminar}, [\texttt{tom}, \texttt{mas}]) \end{array} \right\}$

For example, the task for *writing a paper on the subjects multi-agent systems and theory of mind*, $\texttt{task}(\texttt{write\_paper}, [\texttt{mas}, \texttt{tom}])$, requires competence on both subjects (`mas` and `tom`). Thus, this task should be assigned to a student (or a group of students) who knows both subjects.

4 Semantics for ToM in Agent Communication

4.1 The Basis for the Operational Semantics

To define the semantics for the updates agents execute in their ToM, we extend the original operational semantics of AgentSpeak [33], which is based on a widely

[1] For example, [22,23] propose new performatives for argumentation-based communication between Jason agents.

used method for giving semantics to programming languages [27]. It is important to mention that we are interested in the operational semantics for the updates agents execute in their ToM, which considers the performatives (locutions) as computational instructions that operate successively on the states of agents [18]. The operational semantics is given by a set of inference rules. These inference rules define a transition relation between configurations represented by the tuple $\langle ag, C, M, T, s\rangle$, originally defined in [33], as follows:

- ag is a set of beliefs bs, a set of plans ps, and a set of theories of minds ToM.
- An agent's circumstance C is a tuple $\langle I, E, A\rangle$ where:
  - I is a set of *intentions* $\{i, i', \ldots\}$. Each intention i is a stack of partially instantiated plans.
  - E is a set of *events* $\{(te, i), (te', i'), \ldots\}$. Each event is a pair (te, i), where te is a triggering event and i is an intention—a stack of plans in case of an internal event, or the empty intention T in case of an external event. An example is when the belief revision function (which is not part of the AgentSpeak interpreter but rather of the agent's overall architecture), updates the belief base, the associated events—i.e., additions and deletions of beliefs—are included in this set. These are called *external* events; internal events are generated by additions or deletions of goals from plans currently executing.
  - A is a set of *actions* to be performed in the environment.
- M is a tuple $\langle In, Out\rangle$ whose components characterise the following aspects of communicating agents (note that communication is typically asynchronous):
  - In is the mail inbox: the multi-agent system runtime infrastructure includes all messages addressed to this agent in this set. Elements of this set have the form $\langle mid, id, ilf, cnt\rangle$, where mid is a message identifier, id identifies the sender of the message, ilf is the illocutionary force of the message, and cnt its content: a (possibly singleton) set of AgentSpeak predicates or plans, depending on the illocutionary force of the message.
  - Out is where the agent posts messages it wishes to send; it is assumed that some underlying communication infrastructure handles the delivery of such messages. Messages in this set have exactly the same format as above, except that here id refers to the agent to which the message is to be sent.
- When giving semantics to an AgentSpeak agent's reasoning cycle, it is useful to have a structure which keeps track of temporary information that may be subsequently required within a reasoning cycle. In this particular work, we consider only T_ι, which records a particular intention being considered along the execution of one reasoning cycle.
- The current step within an agent's reasoning cycle is symbolically annotated by $s \in \{\mathsf{ProcMsg}, \mathsf{SelEv}, \mathsf{RelPl}, \mathsf{ApplPl}, \mathsf{SelAppl}, \mathsf{AddIM}, \mathsf{SelInt}, \mathsf{ExecInt}, \mathsf{ClrInt}\}$. These labels stand for, respectively: processing a message from the agent's mail inbox, selecting an event from the set of events, retrieving all relevant plans, checking which of those are applicable, selecting one particular applicable plan (the intended means), adding the new intended means to the set of

intentions, selecting an intention, executing the selected intention, and clearing an intention or intended means that may have finished in the previous step.
- The semantics of AgentSpeak makes use of "selection functions" which allow for user-defined components of the agent architecture. We use here only the S_M function, as originally defined in [33]; the *select message* function is used to select one message from an agent's mail inbox.

In the interests of readability, we adopt the following notation in the semantics rules:

- If C is an AgentSpeak agent circumstance, we write C_E to make reference to the E component of C, and similarly for other components of the multi-agent system and of the configuration of each agent.
- We write $b[s(id)]$ to identify the origin of a belief, where id is an agent identifier (s refers to *source*)

4.2 Tell Performative

It is important to note that when we consider agents that are able to model other agents' minds during communication, both sides, sender and receiver, execute updates in their ToM. The sender will be able to infer the likely model of the receiver's mind after receiving the message, and the receiver will be able to infer the likely model of the sender based on the message received. In the semantics presented in [33], there are separate semantic rules for sending and receiving a message. We follow the same approach here.

Considering the *Tell* performative, when the sender agent sends a message to a receiver agent sid with the content φ, first the sender checks if the receiver will believe that information $Bel_{sid}(\varphi)$, using a function $func\_send$ (which we assume as given and is domain dependent), based on ToM it already has about the receiver ag_{ToM} and the relevant beliefs in its belief base ag_{bs}. The sender will also annotate this ToM belief with a label γ that represents, for example, the likelihood of the belief (i.e., a certainty on the expected state of mind). Note that γ represents an estimation of the uncertainty given that no absolute inference is possible in regards to an agent's private state of mind.

$$\frac{\begin{array}{c} T_\iota = i[head \leftarrow \texttt{.send}(sid, Tell, \varphi); h] \\ func\_send(\varphi, ag_{ToM}, ag_{bs}) = Bel_{sid}(\varphi)_{[\gamma]} \end{array}}{\langle ag, C, M, T, \mathsf{ExecInt}\rangle \longrightarrow \langle ag', C', M', T, \mathsf{ProcMsg}\rangle} \quad \text{(SndTell)}$$

where:

$$\begin{array}{lll} M'_{Out} & = M_{Out} \cup \{\langle mid, sid, Tell, \varphi\rangle\} & \\ & \text{with } mid \text{ a new message identifier;} & \\ C'_I & = (C_I \setminus \{T_\iota\}) \cup \{i[head \leftarrow h]\} & \\ ag'_{ToM} & = ag_{ToM} + Bel_{sid}(\varphi)_{[\gamma]} & \\ C'_E & = C_E \cup \{\langle +Bel_{sid}(\varphi)_{[\gamma]}, \mathsf{T}\rangle\} & \end{array}$$

After the agent updates its mail outbox M_{Out} with the message, it updates its current intention to $i[head \leftarrow h]$ (considering the action $\texttt{.send}(sid, Tell, \varphi)$

that has already been executed), then it updates its ToM with the prediction of a belief $Bel_{sid}(\varphi)_{[\gamma]}$, creating an event $\langle +Bel_{sid}(\varphi)_{[\gamma]}, \mathsf{T}\rangle$ that may be treated in a later reasoning cycle, possibly forming a new goal for the agent based on this new information.

Conversely, when a receiver agent receives a *Tell* message from an agent *sid*, first it checks whether the sender believes φ based on its previous ToM about the sender and the relevant information in its belief base. This expectation of a state of mind results from function *func_rec*. A label γ is used to annotate relevant information such as the confidence on the projected state of mind.

$$\frac{\begin{array}{c} S_M(M_{In}) = \langle mid, sid, Tell, \varphi\rangle \\ func\_rec(\varphi, ag_{ToM}, ag_{bs}) = Bel_{sid}(\varphi)_{[\gamma]} \end{array}}{\langle ag, C, M, T, \mathsf{ProcMsg}\rangle \longrightarrow \langle ag', C', M', T, \mathsf{ExecInt}\rangle} \quad \text{(Tell)}$$

where:

$$\begin{array}{ll} M'_{In} & = M_{In} \setminus \{\langle mid, sid, Tell, \varphi\rangle\} \\ ag'_{bs} & = ag_{bs} + \varphi[s(sid)] \\ ag'_{ToM} & = ag_{ToM} + Bel_{sid}(\varphi)_{[\gamma]} \\ C'_E & = C_E \cup \{\langle +\varphi[s(sid)], \mathsf{T}\rangle\} \cup \{\langle +Bel_{sid}(\varphi)_{[\gamma]}, \mathsf{T}\rangle\} \end{array}$$

After that, the agent updates its mail inbox M_{In}, its belief base ag_{bs} with this new information $\varphi[s(sid)]$ (following the original semantics of AgentSpeak [33]), and it updates its ToM about the sender with $Bel_{sid}(\varphi)_{[\gamma]}$. Both of these updates (on the ToM and the belief base) generate events to which the agent is able to react.

Note that the predictions resulting from *func_send* and *func_rec* can be different from the actual state of mind of the other agents. Therefore, a good prediction model, considering both the ToM and relevant information from the agents' belief base, plays an important role when modelling ToM based on agent communication. Such models might consider the uncertainty present in agent communication, agents' autonomy and self interest, trust relations, reliability, etc. Thus, there are many different ways to instantiate such a model, and our approach allows different models to be implemented through the user-defined *func_send* and *func_rec* functions. Proposing a particular model for uncertainty on ToM is out of the scope of this work. Therefore, we will omit γ in our examples. A model for uncertain ToM can be found in our work presented in [31].

Example: Considering the scenario introduced in Sect. 3, imagine that *John* meets his students every week in order to supervise their work. In a particular meeting with *Alice*, *Alice* asks *John* about the definition of ToM, and *John* responds *Alice* with the following message: $\langle alice, \texttt{tell}, \texttt{definition}(\texttt{tom}, \text{“an approach to model others' minds”})\rangle$. At that moment, *John* is able to model that *Alice* believes the definition of Theory of Mind as “*an approach to model others' minds*”, i.e., *John* models $Bel_{Alice}\texttt{definition}(\texttt{tom}, \text{“an approach to model others' minds”})$ according to the SndTell semantic rule. Also, when *Alice* receives the message, *Alice* is able to model that *John* believes on that definition

for ToM, i.e., *Alice* models $Bel_{John}\texttt{definition}(\texttt{tom},$ "an approach to model others' minds")) according to the TELL semantic rule.

4.3 Achieve Performative

Considering the *Achieve* performative, when a sender agent sends a message with the content φ, it expects that the receiver agent will likely desire φ. It can predict this result using its previous ToM about the receiver, ag_{ToM}, and the relevant information in its belief base, ag_{bs}, resulting in $Des_{sid}(\varphi)_{[\gamma]}$ (where again γ is an estimation of how likely the receiver is to adopt that goal).

$$\frac{\begin{array}{c} T_\iota = i[head \leftarrow .\texttt{send}(sid, Achieve, \varphi); h] \\ func\_send(\varphi, ag_{ToM}, ag_{bs}) = Des_{sid}(\varphi)_{[\gamma]} \end{array}}{\langle ag, C, M, T, \mathsf{ExecInt}\rangle \longrightarrow \langle ag', C', M', T, \mathsf{ProcMsg}\rangle} \quad \text{(SNDACHIEVE)}$$

where:

$$\begin{array}{ll} M'_{Out} & = M_{Out} \cup \{\langle mid, sid, Achieve, \varphi\rangle\} \\ & \text{with } mid \text{ a new message identifier;} \\ C'_I & = (C_I \setminus \{T_\iota\}) \cup \{i[head \leftarrow h]\} \\ ag'_{ToM} & = ag_{ToM} + Des_{sid}(\varphi)_{[\gamma]} \\ C'_E & = C_E \cup \{\langle +Des_{sid}(\varphi)_{[\gamma]}, \mathsf{T}\rangle\} \end{array}$$

The sender agent updates its mail outbox M_{Out}, its current intention, its ToM about the receiver with the prediction $Des_{sid}(\varphi)_{[\gamma]}$, and an event is generated from the update in its ToM.

On the other hand, when a receiver agent receives an *Achieve* message, it can safely conclude that the sender desire φ itself, using its previous ToM about the sender and the relevant information from its belief base.

$$\frac{\begin{array}{c} S_M(M_{In}) = \langle mid, sid, Achieve, \varphi\rangle \\ func\_rec(\varphi, ag_{ToM}, ag_{bs}) = Des_{sid}(\varphi)_{[\gamma]} \end{array}}{\langle ag, C, M, T, \mathsf{ProcMsg}\rangle \longrightarrow \langle ag', C', M', T, \mathsf{ExecInt}\rangle} \quad \text{(ACHIEVE)}$$

where:

$$\begin{array}{ll} M'_{In} & = M_{In} \setminus \{\langle mid, sid, Achieve, \varphi\rangle\} \\ ag'_{ToM} & = ag_{ToM} + Des_{sid}(\varphi)_{[\gamma]} \\ C'_E & = C_E \cup \{\langle +!\varphi, \mathsf{T}\rangle\} \cup \{\langle +Des_{sid}(\varphi)_{[\gamma]}, \mathsf{T}\rangle\} \end{array}$$

The receiver agent updates its mail inbox M_{In} and its ToM about the sender, which generates an event $\langle +Des_{sid}(\varphi)_{[\gamma]}, \mathsf{T}\rangle$. Also, another event $+!\varphi$ is generated, and the agent is able to autonomously decide whether to achieve φ or not. In case it decides to achieve φ, then the agent will look for a plan that achieves φ and make that plan one of its intentions.

Example: Continuing our scenario, imagine that during a meeting with *Bob*, *John* realises that it could be interesting for *Bob* to read a paper about multi-agent systems, so *John* sends the following message to *Bob*: $\langle bob, \texttt{achieve}, \texttt{read}(\texttt{bob}, \texttt{paper\_mas})\rangle$. At that time, *John* is able to model

that *Bob* desires to read the paper, i.e., $Des_{Bob}(\texttt{read}(\texttt{bob}, \texttt{paper\_mas}))$ according to the SndAchieve semantic rule. Also, *Bob* is able to model that *John* desires that *Bob* reads the paper, i.e., $Des_{John}(\texttt{read}(\texttt{bob}, \texttt{paper\_mas}))$ according to the Achieve semantic rule. *Bob* is able to react to the event $+!\texttt{read}(\texttt{bob}, \texttt{paper\_mas})$, searching for a plan to achieve that goal and turning the plan into one of *Bob's* intentions. A simple plan, written in Jason, that *Bob* could use to achieve this goal is shown below:

```
+!read(Ag,Paper)
  :  .my_name(Ag) & desires(Sup,read(Ag,Paper)) & supervisor(Sup,Ag)
  <- read(Paper).
```

The plan above says that, when an event of the type `+!read(Ag,Paper)` is generated, then if `Ag` unifies with the name of the agent executing this plan (obtained with `.my_name(Ag)`), and if the agent believes that its supervisor desires that it reads that paper (`desires(Sup,read(Ag,Paper))` and `supervisor(Sup,Ag)`), then the agent will proceed to execute the action `read(Paper)`. Note that the Achieve semantic rule provides the context (precondition) necessary for *Bob* to execute this plan, considering the unification $\{\texttt{Ag} \mapsto \texttt{bob}, \texttt{Paper} \mapsto \texttt{paper\_mas}, \texttt{Sup} \mapsto \texttt{john}\}$ and that `desires(john,read(bob,paper_mas))` is the code representation for $Des_{John}(\texttt{read}(\texttt{bob}, \texttt{paper\_mas}))$.

4.4 Ask-If Performative

Considering the *AskIf* performative, when the sender agent sends a message with the content φ, the only inference the agent can make is that the other agent will believe that the sender desires to know φ, i.e., $Bel_{sid}(Des_{ag}(\varphi))_{[\gamma]}$.

$$\frac{T_\iota = i[head \leftarrow \texttt{.send}(sid, AskIf, \varphi); h] \quad func\_send(\varphi, ag_{ToM}, ag_{bs}) = Bel_{sid}(Des_{ag}(\varphi))_{[\gamma]}}{\langle ag, C, M, T, \mathsf{ExecInt}\rangle \longrightarrow \langle ag', C', M', T, \mathsf{ProcMsg}\rangle} \quad \text{(SndAskIf)}$$

where:

$$\begin{aligned} M'_{Out} &= M_{Out} \cup \{\langle mid, sid, AskIf, \varphi\rangle\} \\ &\quad \text{with } mid \text{ a new message identifier;} \\ C'_I &= (C_I \setminus \{T_\iota\}) \cup \{i[head \leftarrow h]\} \\ ag'_{ToM} &= ag_{ToM} + Bel_{sid}(Des_{ag}(\varphi))_{[\gamma]} \\ C'_E &= C_E \cup \{\langle +Bel_{sid}(Des_{ag}(\varphi))_{[\gamma]}, \mathsf{T}\rangle\} \end{aligned}$$

The sender agent updates its mail outbox M_{Out}, its current intention and its ToM about the receiver with the prediction $Bel_{sid}(Des_{ag}(\varphi))_{[\gamma]}$, thus an event is generated from the update in its ToM. Conversely, when a receiver agent receives the message, it is able to infer that the sender desires to know φ. After that, in both cases the agent updates its mental state similarly to the other semantic rules.

$$\frac{\begin{array}{c} S_M(M_{In}) = \langle mid, sid, AskIf, \varphi \rangle \\ func\_rec(\varphi, ag_{ToM}, ag_{bs}) = Des_{sid}(\varphi)_{[\gamma]} \end{array}}{\langle ag, C, M, T, \mathsf{ProcMsg} \rangle \longrightarrow \langle ag', C', M', T, \mathsf{ExecInt} \rangle} \quad (\textsc{AskIf})$$

$$\begin{array}{l} \textit{where:} \\ M'_{In} \;\; = M_{In} \setminus \{\langle mid, sid, AskIf, \varphi \rangle\} \\ ag'_{ToM} = ag_{ToM} + Des_{sid}(\varphi)_{[\gamma]} \\ C'_E \;\;\; = C_E \cup \{\langle +Des_{sid}(\varphi)_{[\gamma]}, \mathsf{T} \rangle\} \end{array}$$

Example: Continuing our scenario, imagine that during a group meeting, *John* asks all students if they like paper seminars, using the following message: $\langle \{bob,\ alice,\ nick,\ tom\}, \texttt{AskIf}, \texttt{like}(\texttt{Ag}, \texttt{paper\_seminar}) \rangle$. At that moment *John* considers that all students believe that *John* desires to know who likes paper seminars, $Bel_{Alice}(Des_{John}(\texttt{like}(\texttt{Ag}, \texttt{paper\_seminar})))$, according to the SndAskIf semantic rule. Also, all students think that *John* desires to know who likes paper seminars, $Des_{John}(\texttt{like}(\texttt{Ag}, \texttt{paper\_seminar}))$, according to the AskIf semantic rule. Two simple plans, written in Jason, that students could use to react the event generated by adding $Des_{John}(\texttt{like}(\texttt{Ag}, \texttt{paper\_seminar}))$ to their ToM is shown below:

```
+!desires(Sup,like(Ag,Task))
  :  .my_name(Me) & like(Me,Task) & supervisor(Sup,Me)
  <- .send(Sup,tell,like(Me,Task)).

+!desires(Sup,like(Ag,Task))
  :  .my_name(Me) & ¬like(Me,Task) & supervisor(Sup,Me)
  <- .send(Sup,tell,¬like(Me,Task)).
```

The plans above say that an agent will tell *John* that it likes a particular task if it likes the task. Otherwise, an agent will tell *John* that it does not like that task. For example, *Alice* likes paper seminars, answering *John* with the following message: $\langle john, \texttt{tell}, \texttt{like}(\texttt{alice}, \texttt{paper\_seminar}) \rangle$. In this case, *John* will update its ToM stating that *Alice* likes paper seminars, and *Alice* will update its ToM stating that *John* believes that she likes paper seminars $Bel_{john}(\texttt{like}(\texttt{alice}, \texttt{paper\_seminar}))$, according to the Tell and SndTell semantic rules. In the future, as *John* has this information, it would be able to allocate a task to a student who likes that task.

5 Reaching Shared Beliefs Using ToM

In [33], the authors showed how agents are able to reach shared beliefs. That approach for agents reaching shared beliefs starts with an agent ag_i, which believes in φ, sending to another agent ag_j a `tell` message with the content it desires to became a shared belief, i.e., $\langle ag_j, \texttt{tell}, \varphi \rangle$. Thus, following the semantics in [33], agent ag_j will receive the message and update its belief base with $\varphi[\texttt{source}(\texttt{ag}_\texttt{i})]$. Then, agent ag_i needs to send a message to agent ag_j to

achieve that shared belief, i.e., $\langle ag_j, \texttt{achieve}, \varphi \rangle$, thus the agent ag_j is able to execute the same procedure, sending a tell message to the agent ag_i with φ, i.e., $\langle ag_i, \texttt{tell}, \varphi \rangle$. Finally, agent ag_i receives this message and updates its belief base to $\varphi[\texttt{source(itself)}, \texttt{source}(\texttt{ag}_\texttt{j})]$, reaching the state of shared beliefs.

Definition 1 (Shared Beliefs [33]). *An agent ag_i will reach a state of shared beliefs with another agent ag_j when, for a belief $\varphi[S]$ with S the different sources of φ, both itself and ag_j are sources of φ, i.e.,* $\texttt{source(self)}, \texttt{source}(\texttt{ag}_\texttt{j}) \in \texttt{S}$.

Considering agents that are able to model ToM, we are able to redefine the idea of shared beliefs, including the model of other agents' minds, i.e., a ToM.

Definition 2 (Shared Beliefs using ToM). *An agent ag_i will reach a state of shared beliefs with another agent ag_j when, for a belief φ, it is able to match its own belief φ with a ToM about ag_j believing φ, i.e., $\varphi \wedge Bel_{ag_j}(\varphi)_{[\gamma]}$, with γ the parameter describing, for example, the certainty on ToM required to consider φ a shared belief.*

When we assume that agents are cooperative, they trust each other, and the network infrastructure guarantees that messages will reach their intended receivers, we also are able to assume that there is no uncertainty of the ToM agents model about each other. Thus, we are able to ignore the label γ, which aims to model uncertainty of ToM.

Proposition 1 (Reaching Shared beliefs—ToM without Uncertainty). *Without uncertainty of ToM, agents able to model ToM are able to reach a state of shared beliefs faster (with fewer messages) than agents without this ability.*

Proof (sketch). Following the semantic rule SndTell, when an agent ag_i believes in φ and it is able to model ToM, then it is able to reach a state of a shared belief φ with another agent ag_j communicating a single message $\langle ag_j, \texttt{tell}, \varphi \rangle$ to ag_j. When the agent ag_i sends this message, it updates its ToM with $Bel_{ag_j}(\varphi)$, reaching the state of shared beliefs according to the Definition 2. Agents that are not able to model ToM will need at least two messages, i.e., a `tell` message each, according to the semantics from [33] and Definition 1.

Example: Following the scenario introduced in Sect. 3, imagine that during the meetings *John* has had with his students, the students tell *John* which subjects they know more about, and *John* has the following information of his students, according to the Tell semantic rule:

$$\left\{ \begin{array}{ll} \texttt{knows(alice, tom)} & \texttt{knows(bob, mas)} \\ \texttt{believes(alice, knows(alice, tom))} & \texttt{believes(bob, knows(bob, mas))} \\ \texttt{knows(nick, kr)} & \texttt{knows(ted)} \\ \texttt{believes(nick, knows(nick, kr))} & \texttt{believes(ted, knows(tom, [tom, mas]))} \end{array} \right\}$$

Given this knowledge and the tasks *John* wants to allocate to his students, *John* decides to assign the tasks as follows: $\texttt{task(write\_paper, [mas, tom])}$

to *Ted*, who knows about both subjects needed for completing that task, `task(review_paper, [kr])` to *Nick*, who is the only student able to execute that task, and grouping *Alice* and *Bob* for the task `task(paper_seminar, [tom, mas])`. If *Bob* only knows `mas` and *Alice* only knows `tom`, then they need to share their knowledge in order to successfully perform the task.

Reaching Shared Beliefs: *Alice* and *Bob* need to work together to accomplish this particular task, which requires the subjects `mas` (Multi-Agent Systems) and `tom` (Theory of Mind). *Bob* only knows the subject of `mas` and *Alice* only knows the subject of `tom`. Considering that together *Alice* and *Bob* know both topics in order to help each other during the paper seminar, they decide to exchange knowledge about these topics. Thus, they might reach some shared beliefs (knowledge) about both topics. Note that, in this scenario, *Alice* and *Bob* assume that both are cooperating and both are rational. Thus, *Alice* starts the dialogue telling *Bob* that "*Theory of Mind is an approach to model others' minds*", i.e., $\langle bob, \texttt{tell}, \texttt{def(tom, "an approach to model others' minds")}\rangle$. At that moment, following the semantic rule SndTell, *Alice* updates its ToM with the following information $Bel_{bob}(\texttt{def(tom, "an approach to model others' minds")})$. When *Bob* receives this message, following the semantic rule Tell, *Bob* updates its belief base with the following information `def(tom, "an approach to model others' minds")`, as well as its ToM about *Alice* with $Bel_{alice}(\texttt{def(tom, "an approach to model other minds")})$. By now, both *Alice* and *Bob* have reached a state of shared belief about the definition of `tom`, according to Definition 2. They proceed sharing the relevant information about each topic until they both feel confident about both topics. Reaching shared beliefs (knowledge) is important for this particular task, in which, when the audience asks them questions about the topics `tom` and `mas`, both *Alice* and *Bob* are able to answer the questions because they both have sufficient knowledge about the topics.

6 Future Work

The relation of trust between agents [19,25,26] is an interesting property agents could consider in a model for uncertain ToM. Our approach allows us to model uncertainty through the functions *func_rec*() and *func_send*(), labelling the uncertainty of that information using γ. Even though our approach allows us to model ToM that reflects uncertainty, we believe that the modelling of uncertain ToM is a task that falls beyond the scope of this particular paper and we thus leave it as future work.

Another aspect of ToM to be considered in future work is that ToM can also be inferred by agents from the environment by observing other agents' actions. The modelling of ToM based on these aspects is part of our ongoing research and it faces some more complex issues such as the ones mentioned in [5]: "*the slamming of a door communicates the slammer's anger only when the intended observer of that act realises that the slammer wanted both to slam the door in his*

face and for the observer to believe that to be his intention". This means that there is both uncertainty about the slammer's intentions and uncertainty about the act of slamming the door, which could be caused either by an accidental shove or by natural means, which would not represent a communicative act and, therefore, observing such an event occur should not cause the observer to make any inference about the slammer's mental state.

7 Related Work and Conclusions

As mentioned before, to the best of our knowledge, there is no work that explicitly and formally describes how agents acquire and update ToM during communication. However, our work is inspired by others who have investigated agents that use models of other agents in reasoning and decision making, e.g., [1,6,7,11,12,20,29]. Also, we took some inspiration from the STAPLE language, that seems to have ceased to be used. The STAPLE (Social and Team Agents Programming Language) language has its logic semantics based on joint intention theory [13]. STAPLE has the goal of reaching a fault-tolerant approach to programming teamwork, in which the authors argue that a team is more than a collection of individuals working together to achieve a common goal. The agents in a team must have shared goals as well as a shared mental state [15]. Thus, STAPLE enables agents to specify the models of other agents, as well as temporal properties of actions and events, allowing them to reason about group beliefs, team intentions, and team commitments [14]. Note that our approach is more general than that, in which ToM could be used to implement similar approaches to teamwork and scalable cooperation, which is a likely research direction for our work.

In this paper, we have defined the formal semantics for updates agents execute on their ToM during communication. The formal semantics uses components based on the BDI model and it is, therefore, broadly applicable to any BDI based AOPL. To the best of our knowledge, our work is the first to address a formal model for ToM in agent communication. We have showed not only how agents acquire and update ToM based on agent communication, but we have also shown how agents reason and make decisions using ToM through an illustrative scenario. The modelling, the implementation and the study of agents that are able to model other agents' minds (i.e., ToM) goes beyond the current interests of the AI community, in which the main research scope is to implement rational and efficient software agents that are able to reason and make decisions in order to simulate and study the social behaviour of intelligent entities [24]. ToM is also regarded as important by other research communities that engage in the interdisciplinary study of communication, negotiation, social behaviour, and developmental psychology [4]. We consider that it would be very useful for these interdisciplinary communities to have the possibility to use AOPLs in order to study ToM or any other problems in which ToM plays a significant role.

Acknowledgements. We gratefully acknowledge the partial support from CAPES and CNPq.

References

1. Black, E., Atkinson, K.: Choosing persuasive arguments for action. In: The 10th International Conference on Autonomous Agents and Multiagent Systems, pp. 905–912 (2011)
2. El Fallah Seghrouchni, A., Dix, J., Dastani, M., Bordini, R.H. (eds.): Multi-Agent Programming: Languages, Tools and Applications, 1st edn. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-89299-3
3. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. Wiley Series in Agent Technology. Wiley, Chichester (2007)
4. Carr, A., Slade, L., Yuill, N., Sullivan, S., Ruffman, T.: Minding the children: a longitudinal study of mental state talk, theory of mind, and behavioural adjustment from the age of 3 to 10. Soc. Dev. **27**(4), 826–840 (2018)
5. Cohen, P.R., Perrault, C.R.: Elements of a plan-based theory of speech acts. In: Readings in Distributed Artificial Intelligence, pp. 169–186. Elsevier (1988)
6. de Weerd, H., Verheij, B.: The advantage of higher-order theory of mind in the game of limited bidding. In: Proceedings of the Workshop Reasoning About Other Minds, CEUR Workshop Proceedings. vol. 751, pp. 149–164 (2011)
7. de Weerd, H., Verbrugge, R., Verheij, B.: Higher-order social cognition in rock-paper-scissors: a simulation study. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, pp. 1195–1196 (2012)
8. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an agent communication language. In: Proceedings of the 3rd International Conference on Information and knowledge management, pp. 456–463. ACM (1994)
9. TCC FIPA: FIPA communicative act library specification. Foundation for Intelligent Physical Agents (2008). http://www.fipa.org/specs/fipa00037/SC00037J.html. 15 Feb 2018
10. Goldman, A.I.: Theory of mind. In: The Oxford Handbook of Philosophy of Cognitive Science, vol. 1. Oxford Handbooks Online, 2012 edn. (2012)
11. Hadidi, N., Dimopoulos, Y., Moraitis, P., et al.: Tactics and concessions for argumentation-based negotiation. In: COMMA, pp. 285–296 (2012)
12. Hadjinikolis, C., Siantos, Y., Modgil, S., Black, E., McBurney, P.: Opponent modelling in persuasion dialogues. In: International Joint Conference on Artificial Intelligence IJCAI, pp. 164–170 (2013)
13. Kumar, S., Cohen, P.R.: Staple: an agent programming language based on the joint intention theory. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1390–1391 (2004)
14. Kumar, S., Cohen, P.R., Huber, M.J.: Direct execution of team specifications in STAPLE. In: Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 567–568. ACM (2002)
15. Kumar, S., Cohen, P.R., Levesque, H.J.: The adaptive agent architecture: achieving fault-tolerance using persistent broker teams. In: Proceedings of the 4th International Conference on MultiAgent Systems, pp. 159–166 (2000)
16. Labrou, Y., Finin, T.: A semantics approach for KQML - a general purpose communication language for software agents. In: Proceedings of the 3rd International Conference on Information and Knowledge Management, pp. 447–455. ACM (1994)
17. Mayfield, J., Labrou, Y., Finin, T.: Evaluation of KQML as an agent communication language. In: Wooldridge, M., Müller, J.P., Tambe, M. (eds.) ATAL 1995. LNCS, vol. 1037, pp. 347–360. Springer, Heidelberg (1996). https://doi.org/10.1007/3540608052_77

18. McBurney, P., Parsons, S.: Dialogue games for agent argumentation. In: Simari, G., Rahwan, I. (eds.) Argumentation in Artificial Intelligence, pp. 261–280. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-98197-0_13
19. Melo, V.S., Panisson, A.R., Bordini, R.H.: Argumentation-based reasoning using preferences over sources of information. In: Fifteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2016)
20. Oren, N., Norman, T.J.: Arguing using opponent models. In: McBurney, P., Rahwan, I., Parsons, S., Maudet, N. (eds.) ArgMAS 2009. LNCS (LNAI), vol. 6057, pp. 160–174. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12805-9_10
21. Panisson, A.R., Melo, V.S., Bordini, R.H.: Using preferences over sources of information in argumentation-based reasoning. In: 5th Brazilian Conference on Intelligent Systems, pp. 31–36 (2016)
22. Panisson, A.R., Meneguzzi, F., Fagundes, M., Vieira, R., Bordini, R.H.: Formal semantics of speech acts for argumentative dialogues. In: 13th International Conference on Autonomous Agents and Multiagent Systems, pp. 1437–1438 (2014)
23. Panisson, A.R., Meneguzzi, F., Vieira, R., Bordini, R.H.: Towards practical argumentation in multi-agent systems. In: Brazilian Conference on Intelligent Systems, pp. 98–103 (2015)
24. Panisson, A.R., Sarkadi, S., McBurney, P., Parsons, S., Bordini, R.H.: Lies, bullshit, and deception in agent-oriented programming languages. In: Proceedings of the 20th International Trust Workshop, pp. 50–61 (2018)
25. Parsons, S., Sklar, E., McBurney, P.: Using argumentation to reason with and about trust. In: McBurney, P., Parsons, S., Rahwan, I. (eds.) ArgMAS 2011. LNCS (LNAI), vol. 7543, pp. 194–212. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33152-7_12
26. Parsons, S., Tang, Y., Sklar, E., McBurney, P., Cai, K.: Argumentation-based reasoning in agents with varying degrees of trust. In: The 10th International Conference on Autonomous Agents and Multiagent Systems, pp. 879–886 (2011)
27. Plotkin, G.D.: A structural approach to operational semantics (1981)
28. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Van de Velde, W., Perram, J.W. (eds.) MAAMAW 1996. LNCS, vol. 1038, pp. 42–55. Springer, Heidelberg (1996). https://doi.org/10.1007/BFb0031845
29. Rienstra, T., Thimm, M., Oren, N.: Opponent models with uncertainty for strategic argumentation. In: International Joint Conference on Artificial Intelligence IJCAI, pp. 332–338 (2013)
30. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
31. Sarkadi, S., Panisson, A.R., McBurney, P., Parsons, S., Bordini, R.H.: Towards an approach for modelling uncertain theory of mind in multi-agent systems. In: 6th International Conference on Agreement Technologies (2018)
32. Thimm, M.: Strategic argumentation in multi-agent systems. KI-Künstliche Intelligenz **28**(3), 159–168 (2014)
33. Vieira, R., Moreira, A., Wooldridge, M., Bordini, R.H.: On the formal semantics of speech-act based communication in an agent-oriented programming language. J. Artif. Int. Res. **29**(1), 221–267 (2007)
34. Wooldridge, M.: An Introduction to Multiagent Systems. Wiley, New York (2009)