Lies, Bullshit, and Deception in Agent-Oriented Programming Languages

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

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

Abstract. It is reasonable to assume that in the next few decades, intelligent machines might become much more proficient at socialising. This implies that the AI community will face the challenges of identifying, understanding, and dealing with the different types of social behaviours these intelligent machines could exhibit. Given these potential challenges, we aim to model in this paper three of the most studied strategic social behaviours that could be adopted by autonomous and malicious software agents. These are dishonest behaviours such as lying, bullshitting, and deceiving that autonomous agents might exhibit by taking advantage of their own reasoning and communicative capabilities. In contrast to other studies on dishonest behaviours of autonomous agents, we use an agent-oriented programming language to model dishonest agents' attitudes and to simulate social interactions between agents. Through simulation, we are able to study and propose mechanisms to identify and later to deal with such dishonest behaviours in software agents.

1 Introduction

Agent-Oriented Programming Languages (AOPL) and platforms to develop Multi-Agent Systems (MAS) provide suitable frameworks for modelling agent communication in AI. We can reasonably say that one of the main purposes of AI research is to represent as accurately as possible the way humans use information to perform actions. Actions of humans are sometimes performed by applying dishonest forms of reasoning and behaviour such as lying, bullshitting, and deceiving.

In this paper, we model lies, bullshit and deception in an AOPL named Jason [3], which is based on the BDI (Belief-Desire-Intention) architecture. Modelling these dishonest attitudes in MAS allows us to simulate agent interactions in order to understand how agents might behave if they have reasons to adopt these dishonest behaviours. Understanding such behaviours also allows us to identify and deal with such phenomena, as proposed by [7].

Even though the AI community has investigated computational models of lies [31], "bullshit", and deception [6], to the best of our knowledge, our work is

one of the first attempts to model these types of agent attitudes in the practical context of an AOPL. AOPLs offer an attractive way of improving the research of dishonest agent behaviour through simulations of agent interactions with explicit representation of relevant mental states.

Our study has two main contributions: (i) A comparative model of lies, bullshit, and deception in an AOPL based on the BDI architecture, which allows us to define and simulate these dishonest behaviours. (ii) Making the respective model practical, by implementing an illustrative scenario to show how an agent called *car dealer* is able to deceive other agents called *buyers* in buying a car[3]. In this scenario, the *car dealer* also tells lies and bullshit in order to make the buyers believe a car is suitable for them, when in fact it is not.

2 Background

2.1 Lie, Bullshit, Deception

We will start by describing what lying is from an agent-based perspective. We define lying similar to [6]. A lie is a false statement about something that is intended to make someone believe the opposite of what is actually true. Lying cannot be reduced to linguistic communication only. Liars give out information to others in various forms, such as social behaviour, facial expressions, physiological responses to questions, and manipulation of the environment [11, 5].

Definition 1 (Lying). *The dishonest behaviour of an agent Ag_i to tell another agent Ag_j that $\neg\psi$ is the case, when in fact Ag_i knows that ψ is the case.*

Bullshit is different from lying in the sense that it is not intended to make someone believe the opposite from the truth. A bullshiter agent will give an answer to a question in such a way that the one who asked the question is left with the impression that the bullshiter agent knows the true answer [12], when in fact it does not.

Definition 2 (Bullshit). *The dishonest behaviour of an agent Ag_i to tell another agent Ag_j that ψ is the case, when in fact Ag_i does not know if ψ is the case.*

Deception is more complex than bullshit or lying. We define deception as the intention of an agent (Deceiver) to make another agent (Interrogator) believe something is true that it (the Deceiver) thinks is false, with the scope of reaching an ulterior goal or desire. The complexity arises due to the fact that an agent requires *Theory-of-Mind* (henceforth ToM) to deceive [17]. ToM is not needed to tell a lie or to bullshit (although there are cases in which liars or bullshitters can make use of ToM). The Deceiver has to let the Interrogator reach the conclusion by itself. For example, if the Deceiver wants the Interrogator to believe that q is the case, instead of directly telling the Interrogator that q is the case, the Deceiver uses some knowledge that the Interrogator possesses, let's say $p \rightarrow q$,

[3] The implementation of this work is available at `https://tinyurl.com/ybrmkqg9`.

and tells the Interrogator that p is the case. Having told the Interrogator that p is the case, the Deceiver then knows that if the Interrogator is a rational agent that has the ability to apply *Modus Ponens*, then it will conclude that q is the case. Levine and McCornack call this interplay *Pars Pro Toto* (the information the Deceiver decides to feed the Interrogator) and *Totum Ex Parte* (the knowledge the Interrogator derives from the information sent by the Deceiver) [21].

One can argue that liars and bullshitters might have some types of motivations or goals. However, compared to deceivers, these goals do not contain ulterior motives. A liar, for example, can have the goal to speak falsely about a state of the world without taking into consideration the state of mind of the agent it speaks to. It can also be argued that a good liar would take into account its target's mind, although by definition a liar is constrained by one single strategy which is to speak falsely about a state of the world. A bullshitter can have the goal to make the agent it speaks to believe it (the bullshitter) is speaking the truth independently of the state of the world it is speaking about. Most of the times, however, bullshitters do not take into consideration the target's mental activity in order to deliver a bullshit.

Definition 3 (Deception). *The **intended** dishonest behaviour of an agent Ag_i to tell another agent Ag_j that ψ is the case, when in fact Ag_i knows that $\neg\psi$ is the case, in order to make Ag_j conclude that φ given that Ag_i knows that Ag_j knows that $\psi \rightarrow \varphi$ and Ag_i also knows that Ag_j is rational.*

2.2 Agent Oriented Programming Language

Among the many AOPL 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 [27], which is one of the best-known languages inspired by the BDI architecture. In Jason, the agents are equipped with a library of pre-compiled plans that have the following syntax:

```
triggering_event : context <- body.
```

where the `triggering_event` represents the way agents react to events, for example, a new goal for the agent to pursue, or a new belief in case the plan is to be triggered by reaction to perceived changes in the world; the `context` has the preconditions for the plan to be deemed applicable for achieving that goal given the current circumstances, and the `body` is a sequence of actions and sub-goals to achieve the goal.

Besides specifying agents with well-defined mental attitudes based on the BDI architecture, 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., `safe(car1)`; (ii) believes to be false, e.g., ¬`safe(car1)`; (iii) is ignorant about, i.e., the agent has no information about whether the car is safe 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: `safe(car1)[source(seller)]`, stating that the source of the belief that `car1` is safe is the agent `seller`. The annotations in Jason can be easily extended to include other meta-information, for example trust and time as used in [22, 25]. 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]. Further, new performatives can be easily defined (or redefined) in order to give special meaning to them[4].

3 Running Example

To show the difference between the agents' attitudes of telling a lie, telling bullshit and deceiving, we will present an approach to model these three agent attitudes in an agent-oriented programming language using a running example of a car dealer scenario[5], inspired by [20, 23, 31]. In our scenario, an agent called *car dealer*, *cd* for short, has the desire to sell as many cars as it can. Thus, the car dealer will use all its available strategies, including lying, bullshitting, and attempting to deceive the customers to buy the cars it has for sale.

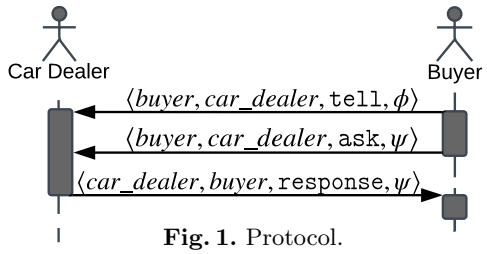

Fig. 1. Protocol.

An illustration of the communication protocol for our scenario is shown in Figure 1. The protocol states that: a *buyer* agent will tell to another agent, the *car dealer*, the set ϕ of characteristics they desire in buying a car. For example, $\phi =$ `inference(buy(car),[safe(car),comfortable(car)])`, means that the *buyer* considers safety and comfort to be the most desirable characteristics for buying a car. After that, *buyers* ask the *car dealer* about the cars they have an interest to buy. The *car dealer* answers the questions based on its own interest (i.e., it is a self-interested agent).

In our scenario, we will focus on characteristics of cars such as: safety, speed, comfort, and storage size which are defined in Δ_{car_dealer}:

[4] For example, [26] proposes new performatives for argumentation-based communication between Jason agents.

[5] We do not assume that in real life car dealers are deceptive agents, we just use this particular scenario as an illustrative example.

$$\Delta_{car\_dealer} = \left\{ \begin{array}{ll} \texttt{safe(ford)} & \neg\texttt{comfortable(ford)} \\ \texttt{safe(bmw)} & \neg\texttt{comfortable(bmw)} \\ \neg\texttt{safe(renault)} & \neg\texttt{comfortable(renault)} \\ \neg\texttt{fast(ford)} & \texttt{large\_storage(bmw)} \\ \texttt{fast(bmw)} & \neg\texttt{large\_storage(renault)} \end{array} \right\}$$

Here, there are two important considerations for our model. The first consideration is about the diversity of information the *car dealer* knows, which is fundamental when simulating the agents' behaviours. We set up our scenario with different cars, in which each car has different characteristics. The second consideration is that the *car dealer* may be able to model the *buyers*' mental state, which means the *car dealer* is able to model the characteristics *buyers* consider important to buy a car, i.e., ϕ. Given the knowledge of the characteristics the *buyers* consider important, the *car dealer* is able to simulate the influence of the information it provides, choosing the best answer according to its own interest or desire, i.e., to sell the cars. Thus, an agent may be able not only to model the initial states of other agents minds, but also to simulate how the minds of these other agents change over time[6].

4 Modelling Buyers' Minds

In this work, we set up the notation based on the Jason agent-oriented programming language [3] and a standard representation for messaging. Our model will consist of predicates which represent the mental state of agents, the event model which is the set of perceptions and possible messages that the agents can communicate such as asking and answering questions, the belief update rules for each kind of message and perception, and inference rules that allow agents to execute belief update and reasoning simulation.

4.1 Modelling the Minds of Other Agents

Agents will model others agents' minds according to inferences they are able to make, that are based on the perceptions they have of the target agents and the communication they have with the target. These ideas come from studies in Theory of Mind (ToM) [14]. Thus, based on the BDI architecture, we use the following predicates to allow an agent to model the other agents' minds:

- `believes(`*ag*`,prop)` means that an agent *ag* believes on proposition `prop`. For example, `believes(john,safe(ford))` means that `john` believes that `ford` are safe. A *car dealer* agent *cd* is able to model the beliefs of a *buyer* agent *ag* after receiving a `tell` message from *ag*, i.e., $\langle ag, cd, \texttt{tell}, \texttt{prop}\rangle$. We use $\Delta_{cd} \models \texttt{believes}(ag, \texttt{prop})$ to describe that the car dealer *cd* knows that the buyer *ag* believes on `prop`. A particular case for this predicate

[6] These abilities of our agents reflect their capacity of using both Theory-Theory of Mind and Simulation-Theory of Mind for modelling the minds of their targets [13].

is `believes(ag,inference(prop,S))` representing that an agent *ag* believes on the inference from `S` (a set of predicates) to `prop`. For example, `believes(john,inference(buy(bmw),[safe(bmw)])` means that `john` believes that if a `bmw` is `safe` it could buy a `bmw`.
- `desires(ag,prop)` means that an agent *ag* desires `prop`. For example, `desires(john,buy(bmw))` means that `john` desires to buy a `bmw`. We use $\Delta_{cd} \models \texttt{desires}(ag, \texttt{prop})$ to describe that the car dealer *cd* knows that the buyer *ag* desires `prop`.

We are also able to use nested representations for beliefs and desires. For example, we are able to express that the *car dealer cd* believes that the *buyer ag* desires to buy a car, i.e., `believes(cd,desires(ag,buy(_)))`, which it is the same $\Delta_{cd} \models \texttt{desires}(ag,\texttt{buy}(\_))$.[7]

4.2 Modelling Agents' Actions and Communication Updates

Agents will update their ToM about others when communicating with them, as well as when perceiving them in the environment. For simplicity, in this work we will consider only a few communication actions, based on the protocol described in Section 3. Thus, the possible actions and belief updates of the agents are the following:

- $\langle ag, cd, \texttt{tell}, \texttt{prop}\rangle$ means a message sent by the agent *ag* to the agent *cd*, with the performative `tell`, and the content `prop`. When *cd* receives this message, it executes the following update in its ToM:

$$\Delta_{cd} = \Delta_{cd} \cup \texttt{believes}(ag, \texttt{prop})$$

- $\langle ag, cd, \texttt{ask}, \texttt{prop}\rangle$ means a message sent by the agent *ag* to the agent *cd*, with the performative `ask` and the content `prop`. When *cd* receives this message, it executes the following update in its ToM:

$$\Delta_{cd} = \Delta_{cd} \cup \texttt{desires}(ag, \texttt{prop})$$

- $\langle cd, ag, \texttt{response}, \texttt{prop}\rangle$ means a message sent by the agent *cd* to the agent *ag*, with the performative `response` and the content `prop`. To execute this action, it requires that a previous message $\langle ag, cd, \texttt{ask}, \texttt{prop}\rangle$ has been communicated. Thus, the agents *cd* and *ag* execute the following updates in their ToM and knowledge base, respectively:

$$\Delta_{cd} = \Delta_{cd} \cup \texttt{believes}(ag, \texttt{prop})$$

$$\Delta_{ag} = \Delta_{ag} \cup \texttt{prop}[\texttt{source}(cd)]$$

Note that the semantics for a `response` message is different from the `tell` message, given that a `tell` message expresses the opinion of the sender, and the `response` message represents an information previously requested, which means

[7] To investigate different levels of ToM in multi-agent systems is out of the scope of this paper, thus we use only first-order ToM, i.e., we do not model ToM about others' ToM, and ToM about others' ToM about others' ToM, and so forth.

it represents a desired update the receiver wants to execute in its knowledge base.

Finally, an agent also is able to update its ToM perceiving other individuals that are situated in the same environment. In this work, the *car dealer* is able to perceive the *buyers* when they enter in the sale room, i.e., an event (perception) of the type `+client(`*`ag`*`)` is generated by the environment, enabling the *car dealer* to infer that the *buyer ag* desires to buy a car. The *car dealer cd*'s ToM is updated as follows:

$$\Delta_{cd} = \Delta_{cd} \cup \texttt{desires}(ag, \texttt{buy}(\_))$$

Note that, while the perceptions from the environment are domain dependent, the communication semantics are independent of the domain. This is because the meaning of the performatives guides the way in which an agent executes its belief updates. That is, for different environments, the agents' perceptions from the environment may have different meanings, and by extension beliefs will be updated in different ways.

4.3 Making Inferences from the Models of Other Agents' Minds

It is important to model when an agent is *ignorant* about the truth of a proposition. That is, considering multi-agent systems that model a *open* world, when an agent does not know if ϕ is true, that does not mean that ϕ is false, i.e., when an agent cannot infer either ϕ or $\neg\phi$, the only conclusion it may reach is that it is *ignorant* about the truth of ϕ. An agent is able to infer that it is ignorant about a proposition using the following inference rule:

```
ignorant_about(Prop) :- not(Prop) & not(¬Prop).
```

Similarly, an agent is able to infer that it is ignorant about other agents' mental states, using the following inference rules:

```
ignorant_about(believes(Ag,Prop)) :-  not(believes(Ag,Prop)) &
                                      not(¬believes(Ag,Prop)).
ignorant_about(desires(Ag,Prop)) :-  not(desires(Ag,Prop)) &
                                      not(¬desires(Ag,Prop)).
```

Furthermore, an agent is able to infer new information about other agents' mental state from the information it already has on its ToM. For example, if the *car dealer* agent *cd* knows that the *buyer* agent *ag* believes that `ford` are `safe`, i.e., `believes(`*`ag`*`,safe(ford))`, and that *ag* also believes in the inference that safe cars are good options to buy, i.e., `believes(`*`ag`*`,inference(buy(X),[safe(X)]))`, *cd* is able to infer that *ag* also believes that the `ford` is a good option to buy, i.e., `believes(`*`ag`*`,buy(ford))`:

```
believes(Ag,C) :- believes(Ag,inference(C,P)) & believes(Ag,P).
```

The *car dealer* will not know the beliefs of the *buyers* about each car in advance. An interesting way for the *car dealer* to gain this knowledge is for the *car dealer* to be able to simulate the conclusions a *buyer* might reach based on the information the *car dealer* provides and the inferences the *buyer* is able to execute:

```
implies(believes(Ag,N),believes(Ag,C)) :- believes(Ag,inference(C,N)).
```

Thus, if the *car dealer cd* knows that the *buyer ag* believes that safe cars are a good option to buy, i.e., `believes(ag,inference(buy(X),[safe(X)]))`, then *cd* also knows in advance that *ag* will believe that `ford` are good options to buy, i.e., `believes(ag,buy(ford))`, if and only if *cd* provides *ag* the information that `ford` are safe, i.e., `believes(ag,safe(ford))`.

5 Modelling Lies in AOPL

Using our model, we are able to model a lie following the scenario of when the *car dealer cd* knows that $\neg\psi$ (ψ is not true), but it responds either ψ or `ignorant_about(cd,`ψ`)` to *buyer ag*.

Table 1. Conditions for a Lie.

Car Dealer (*cd*)	**Buyer (*ag*)**
Beliefs: $\neg\psi$	Beliefs: `ignorant_about(`ψ`)`
Actions: $\langle cd, ag,$ `response`$, \psi\rangle$	Desires: ψ
ToM: `desires(ag,`ψ`)`	Actions: $\langle ag, cd,$ `ask`$, \psi\rangle$

As described, a liar could tell lies without any particular goal, but the most common situation requires some motivation that makes an agent tell a lie, in order to achieve a particularly desired state of the world and/or a state of mind. We will discuss this motivation further in this paper. For now let's assume that the a *buyer ag* asks the *car dealer* if `renault` are safe, i.e., $\langle ag, cd,$ `ask`$,$ `safe(renault)`$\rangle$. In this case, based on *cd*'s knowledge base represented in Δ_{car_dealer}, *cd* has two options: either telling the truth, i.e., $\langle cd, ag,$ `response`$,\neg$`safe(renault)`$\rangle$, or telling a lie, i.e., either $\langle cd, ag,$ `response`$,$ `ignorant_about(cd,safe(renault))`$\rangle$ or $\langle cd, ag,$ `response`$,$ `safe(renault)`$\rangle$.

6 Modelling Bullshit in AOPL

Using our model, we are able to model a bullshit based on the scenario of when the *car dealer cd* is ignorant about ψ, i.e., `ignorant_about(`ψ`)`, but it responds either ψ or $\neg\psi$ to the *buyer ag*.

Table 2. Conditions for Bullshit.

Car Dealer (*cd*)	**Buyer (*ag*)**
Beliefs: `ignorant_about(`ψ`)`	Beliefs: `ignorant_about(`ψ`)`
Actions: $\langle cd, ag,$ `response`$, \psi\rangle$	Desires: ψ
ToM: `desires(ag,`ψ`)`	Actions: $\langle ag, cd,$ `ask`$, \psi\rangle$

Similarly to a liar, a bullshiter could tell bullshit without a particular goal, but the most common situation requires some motivation, as we will discuss further in this paper. For now, let's assume that the *buyer ag* asks to the *car dealer*

if `renault` are fast, i.e., $\langle ag, cd, \texttt{ask}, \texttt{fast(renault)}\rangle$. In this case, based on *cd*'s knowledge base represented in Δ_{car_dealer}, *cd* has two options: either telling the truth, i.e., $\langle cd, ag, \texttt{response}, \texttt{ignorant\_about(fast(renault))}\rangle$, or telling a bullshit, i.e., either $\langle cd, ag, \texttt{response}, \texttt{fast(renault)}\rangle$ or $\langle cd, ag, \texttt{response}, \neg\texttt{fast(renault)}\rangle$.

7 Modelling Deception in AOPL

One question that arises from Sections 5 and 6 is: how does the *car dealer cd* decide what to answer? For example, How does it choose between lying by telling ψ or lying by telling $\texttt{ignorant\_about(}cd\texttt{,}\psi\texttt{)}$, when it knows $\neg\psi$ is true? We argue that the answer for that question is the *motivation* or *ulterior goal* of the *car dealer cd*. In this particular piece of work, we model deception using the motivation of the *car dealer cd* of making the *buyers* to buy a car which is not suitable for the *buyers* according to the buyers' requirements communicated in the first interaction of our protocol.

There are two major reasons we consider the scenario in which car dealers are deceivers. The first reason is because car dealers usually have an ulterior goal, that is to sell cars. This goal is related to both the state of the world (usually the properties of the car the dealer is trying to sell) and to the mind of the target. The dealer needs to take into account the preferences and attitudes (considered by us as beliefs of the target) in order to provide the information that will make the target believe it should buy the car. The second reason is because car dealers do not care if the target agent believes they (the car dealing agents) know the truth about the state of the world (or state of the car in this particular case). Their ulterior goal is not to make the buyer believe they have true knowledge about the car (as a bullshitter would want the buyer to believe). The car dealer's goal is to make the buyer reach the conclusion that it (the buyer) should buy the car by itself. In order to make the buyer reach that particular conclusion, the dealer needs to feed the buyer a set of particular pieces of information (true or false).

Table 3. Conditions for Deception.

Car Dealer (*cd*)	**Buyer (*ag*)**
Beliefs: $\neg\psi$	Beliefs: $\texttt{believes(inference(}\varphi\texttt{, }\psi\texttt{))}$, $\texttt{ignorant\_about(}\psi\texttt{)}$
Desires: $\texttt{believes(}ag\texttt{,}\varphi\texttt{)}$	Desires: ψ
Actions: $\langle cd, ag, \texttt{response}, \psi\rangle$	Actions: $\langle ag, cd, \texttt{tell}, \texttt{inference(}\varphi\texttt{,}\psi\texttt{)}\rangle$, $\langle ag, cd, \texttt{ask}, \psi\rangle$
ToM: $\texttt{believes(}ag\texttt{,inference(}\varphi\texttt{,}\psi\texttt{))}$, $\texttt{desires(}ag\texttt{,}\psi\texttt{)}$	

Imagine that a *buyer ag* starts a dialogue with the *car dealer cd* by telling *cd* that it considers safety and speed to be the most important characteristics when buying a car, i.e., $\langle ag, cd, \texttt{tell}, \texttt{inference(buy(X),[fast(X),safe(X)])}\rangle$. When *ag* asks *cd* if `renault` are safe, i.e., $\langle ag, cd, \texttt{ask}, \texttt{safe(renault)}\rangle$, it makes *cd* model that $\texttt{desires(}ag\texttt{,safe(renault))}$. Thus, *cd* satisfies the precondition necessary for deceiving *ag* (see *cd*'s ToM in Table 3). Imagine also that *cd*'s

desire is for *buyers* to believe that they should buy the car *cd* is selling. Then, the agent *cd* models that a buyer *ag* considers safety and speed the essential characteristics to buy a car, and that *ag* desires to know if `renault` are safe, i.e., *cd* models `believes(`*ag*`,inference(buy(X,[safe(X),fast(X)]))` and `desires(`*ag*`,safe(renault))` in its ToM. What follows from this is that now, *cd* is able to infer that if it gives a positive answer `safe(renault)`, then this will determine *ag* to believe `buy(renault)`. Therefore, *cd* decides to send the message $\langle cd, ag, \texttt{response}, \texttt{safe(renault)}\rangle$, lying about `safe(renault)`. What happens next is that *ag* asks if `renault` are fast, i.e., $\langle ag, cd, \texttt{ask}, \texttt{fast(renault)}\rangle$. Again, *cd* executes the same reasoning process as before. Therefore, *cd* will answer $\langle cd, ag, \texttt{response}, \texttt{fast(renault)}\rangle$, telling bullshit about `fast(renault)`. In the final step, *cd* is able to conclude that it has managed to deceive *ag* because *cd* is able to model in its ToM that `believes(`*ag*`,safe(renault))`, `believes(`*ag*`,fast(renault))` and `believes(`*ag*`,inference(buy(X),[safe(X),fast(X)]))`. This allows *cd* to conclude [8] `believes(`*ag*`,buy(renault))` that corresponds to *cd*'s ulterior goal.

8 Related Work

Various studies have investigated the use of ToM in multi-agent systems. Among them, [10, 9] investigate the advantages of using different levels of ToM in games played by agents. Others have applied the idea of modelling the opponent in order to evaluate strategies for argumentation-based dialogues [1, 15, 16, 24, 28], for example, in [1] agents consider the recipient's model in order to choose the most persuasive arguments, based on the recipient's values. Regarding dishonest attitudes, many works are found in the AI literature. [18] models self-deception using epistemic logic. [30] defines multiple types of deception using a modal logic of belief and action. [19] builds a cognitive model of deception based on human-computer interaction. [4] introduces a framework for agents to enable them to make socially aware inferences in order to detect deception. [17] demonstrates that a crucial condition for agents to deceive and detect deception is a ToM. [31] examines the notion of lying in agent-based systems dialogues, including situations and dialogues when it is acceptable for agents to communicate locutions that contradict their beliefs, i.e., situations in which it is acceptable for agents to lie. [6] describes the difference among three classes of dishonesty: lies, bullshit and deception. [29] studies a computational logic for *dishonest reasoning*.

The work that is closest to our approach is [8], where the author defines a formal framework that represents a theoretical machine which uses ToM to formulate deceptive sophistic arguments. The framework has been proved to work through psychological experiments on human subjects.

9 Conclusion

In this work, we proposed a representation for modelling and simulating other agents' minds using an AOPL. Our representation is based on literature on

[8] This scenario corresponds to the `buyer1` in our implementation.

Theory of Mind and the BDI architecture. Furthermore, using the proposed representation, we described a model for three of the most studied dishonest attitudes in AI literature, i.e., *lying*, *bullshitting* and *deceiving*. In particular, we modelled and implemented these attitudes in Jason [3], which is a well-known agent-oriented programming language inspired by the BDI architecture.

Modelling and implementing such attitudes in an AOPL allows us to investigate agents' dishonest behaviours through simulations in a high-level, declarative approach. On one hand, in this particular piece of work, we have used a *car dealer* scenario, which, given its simplicity, allowed us to focus on the main contribution of this paper, i.e., the representation of other agents' minds and the modelling and simulation of lies, bullshit and deception in MAS. On the other hand, our approach is generic and can be easily used to model and simulate other scenarios of dishonest agent behaviour, which we highlight as one strand of future work.

Acknowledgements

This research was partially funded by CNPq and CAPES.

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. Bordini, R.H., Dastani, M., Dix, J., Seghrouchni, A.E.F.: Multi-Agent Programming: Languages, Tools and Applications. Springer Publishing Company, Incorporated, 1st edn. (2009)
3. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology). John Wiley & Sons (2007)
4. Bridewell, W., Isaac, A.: Recognizing deception: A model of dynamic belief attribution. In: AAAI Fall Symposium: Advances in Cognitive Systems (2011)
5. Burgoon, J.K., Buller, D.B., Floyd, K., Grandpre, J.: Deceptive realities: Sender, receiver, and observer perspectives in deceptive conversations. Communication Research 23(6), 724–748 (1996)
6. Caminada, M.: truth, lies and bullshit; distinguishing classes of dishonesty. In: In: Social Simulation Workshop at the International Joint Conference on Artificial Intelligence (SS@ IJCAI. Citeseer (2009)
7. Castelfranchi, C., Tan, Y.H.: Trust and deception in virtual societies. Springer (2001)
8. Clark, M.H.: Cognitive illusions and the lying machine: a blueprint for sophistic mendacity. Ph.D. thesis, Rensselaer Polytechnic Institute (2010)
9. De Weerd, H., Verheij, B.: The advantage of higher-order theory of mind in the game of limited bidding. In: Proc. Workshop Reason. About Other Minds, ceur workshop proceedings. vol. 751, pp. 149–164 (2011)
10. de Weerd, H., Verbrugge, R., Verheij, B.: Higher-order social cognition in rock-paper-scissors: A simulation study, pp. 218–232 (2012), m1 - Book, Section
11. Ekman, P., Friesen, W.V.: Nonverbal Leakage and Clues to Deception . Psychiatry 32(1), 88–106 (1969)

12. Frankfurt, H.G.: On bullshit. Princeton University Press, (2009)
13. Goldman, A.I.: Theory of mind. In: The Oxford Handbook of Philosophy of Cognitive Science, vol. 1. Oxford Handbooks Online, 2012 edn. (2012)
14. Gopnik, A., Glymour, C., Sobel, D.M., Schulz, L.E., Kushnir, T., Danks, D.: A theory of causal learning in children: causal maps and bayes nets. Psychological review 111(1), 3 (2004)
15. Hadidi, N., Dimopoulos, Y., Moraitis, P., et al.: Tactics and concessions for argumentation-based negotiation. In: COMMA. pp. 285–296 (2012)
16. 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)
17. Isaac, A., Bridewell, W.: White lies on silver tongues: Why robots need to deceive (and how). Oxford University Press (11 2017)
18. Jones, A.J.: On The Logic of Self-deception. South American Journal of Logic 1, 387–400 (2015)
19. Lambert, D.: A cognitive model for exposition of human deception and counterdeception. Tech. rep., DTIC Document (1987)
20. McBurney, P., Van Eijk, R.M., Parsons, S., Amgoud, L.: A dialogue game protocol for agent purchase negotiations. Autonomous Agents and Multi-Agent Systems 7(3), 235–273 (2003)
21. McCornack, S.A., Morrison, K., Paik, J.E., Wisner, A.M., Zhu, X.: Information manipulation theory 2: a propositional theory of deceptive discourse production. Journal of Language and Social Psychology 33(4), 348–377 (2014)
22. 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)
23. Modgil, S., Toni, F., Bex, F., Bratko, I., Chesnevar, C.I., Dvořák, W., Falappa, M.A., Fan, X., Gaggl, S.A., García, A.J., et al.: The added value of argumentation. In: Agreement Technologies, pp. 357–403. Springer (2013)
24. Oren, N., Norman, T.J.: Arguing using opponent models. In: ArgMAS. pp. 160–174. Springer (2009)
25. Panisson, A.R., Melo, V.S., Bordini, R.H.: Using preferences over sources of information in argumentation-based reasoning. In: Brazilian Conference on Intelligent Systems, BRACIS (2016)
26. Panisson, A.R., Meneguzzi, F., Fagundes, M., Vieira, R., Bordini, R.H.: Formal semantics of speech acts for argumentative dialogues. In: Thirteenth Int. Conf. on Autonomous Agents and Multiagent Systems. pp. 1437–1438 (2014)
27. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : agents breaking away: agents breaking away. pp. 42–55. MAAMAW '96, Springer-Verlag New York, Inc., Secaucus, NJ, USA (1996)
28. 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)
29. Sakama, C.: Dishonest reasoning by abduction. In: IJCAI Proceedings-International Joint Conference on Artificial Intelligence. vol. 22, p. 1063 (2011)
30. Sakama, C., Caminada, M.: The many faces of deception. Proceedings of the Thirty Years of Nonmonotonic Reasoning (NonMon@ 30) (2010)
31. Sklar, E., Parsons, S., Davies, M.: When is it okay to lie? a simple model of contradiction in agent-based dialogues. In: ArgMAS. pp. 251–261. Springer (2004)