MAIDS — a Framework for the Development of Multi-Agent Intentional Dialogue Systems

Débora C. Engelmann
School of Technology – PUCRS
Porto Alegre, Brazil
DIBRIS – University of Genoa
Genoa, Italy
debora.engelmann@edu.pucrs.br

Alison R. Panisson
Department of Computing – UFSC
Araranguá, Brazil
alison.panisson@ufsc.br

Renata Vieira
CIDEHUS – University of Évora
Évora, Portugal
renatav@uevora.pt

Jomi Fred Hübner
Automation and Systems Engineering – UFSC
Florianópolis, Brazil
jomi.hubner@ufsc.br

Viviana Mascardi
DIBRIS – University of Genoa
Genoa, Italy
viviana.mascardi@unige.it

Rafael H. Bordini
School of Technology – PUCRS
Porto Alegre, Brazil
rafael.bordini@pucrs.br

ABSTRACT

This paper introduces a framework for programming highly sophisticated multi-agent dialogue systems. The framework is based on a multi-part agent belief base consisting of three components: (i) the main component is an extension of an agent-oriented programming belief base for representing defeasible knowledge and, in particular, argumentation schemes; (ii) an ontology component where existing OWL ontologies can be instantiated; and (iii) a theory of mind component where agents keep track of mental attitudes they ascribe to other agents. The paper formalises a structured argumentation-based dialogue game where agents can "digress" from the main dialogue into subdialogues to discuss ontological or theory of mind issues. We provide an example of a dialogue with an ontological digression involving humans and agents, including a chatbot that we developed to support bed allocation in a hospital; we also comment on the initial evaluation of that chatbot carried out by domain experts. That example is also used to show that our framework supports all features of recent desiderata for future dialogue systems.

KEYWORDS

Agent-oriented programming; Argumentation; Ontological reasoning; Theory of mind; Dialogue systems

ACM Reference Format:
Débora C. Engelmann, Alison R. Panisson, Renata Vieira, Jomi Fred Hübner, Viviana Mascardi, and Rafael H. Bordini. 2023. MAIDS — a Framework for the Development of Multi-Agent Intentional Dialogue Systems. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023,* IFAAMAS, 9 pages.

1 INTRODUCTION

This paper addresses, in the context of a BDI agent programming language, three important aspects of a cognitive agent: the ability to argue, ontological knowledge and reasoning, and reasoning about mental attitudes of the agent itself as well as others. The paper puts forward the idea of a dialogue structure which allows agents, while arguing about a particular domain, to enter into *subdialogues* about ontological knowledge related to that domain, or about mental attitudes of others, much as humans often do. For example, when arguing about the best candidate in an election, we might digress into a discussion on whether a particular candidate should be classified as left or right wing, or digress into a discussion on whether one of our interlocutors holds progressive or conservative beliefs, given a possible disparity between the interlocutors theory of mind.

In this paper, we formalise, implement, evaluate, and demonstrate the expressivity of a framework for the development of dialogue systems built on top of Jason [9]. In that framework, agents have three separate components of their belief base: (i) argumentation schemes for the application domain that the dialogue system is aimed for, following a structured (rather than abstract) argumentation approach; (ii) an OWL ontology about that same domain; and (iii) a Theory of Mind (ToM) component storing presumed mental attitudes of other agents. With that multi-part belief base setting, our framework provides support for agents having a structured dialogue where the main line of argumentation is based on the argumentation schemes knowledge component but it can lead to subdialogues when ontological or ToM issues need to be resolved. This paper focuses on the expressivity of dialogue systems where agents have such a multi-part belief base together with the ability to engage in such structured dialogues.

The idea of subdialogues is in line with general ideas on nested dialogues (see, e.g., [7]), but we give in this paper a practical protocol limiting such "digressions", thus avoiding unnecessary computational burden. In fact, the multi-part belief base accompanied by the dialogue structure with subdialogues has a clear impact on efficiency, given that commitment stores of subdialogues can be deleted when they are completed. Importantly, because this is all in the context of an agent-oriented programming language that is formally based on the BDI architecture, we have precise and computationally-grounded [40] semantics for the mental attitudes that agents have and ascribe to others.

Although all the knowledge of the multi-part belief base, if suitably translated from the various sources, could be merged and used

---

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

by argumentation systems as a single knowledge base, there are two main advantages of the modular approach we propose here: (i) it allows us to reuse existing ontologies on top of the more expressive (argumentation-based) reasoning that we may want to program for particular systems (i.e., encouraging reusability of existing ontologies in agent development); and (ii) it allows the agent strategy to "consciously" decide when to move on to an ontological argumentation[1] or argumentation about other agents' mental attitudes before returning to the main line of argumentation.

By putting together the ability to argue, to reason about ontological knowledge, and to represent a ToM, our framework supports the development of dialogue systems that satisfy all the desiderata for future dialogue systems recently put forward by P. Cohen [10] to overcome the limitations of current dialogue tools, as well as other desiderata appearing in the recent literature. The expressivity of our approach is demonstrated through a case study on a dialogue system including agents and humans for a healthcare scenario, more specifically, a MAS that supports hospital staff in making decisions about bed allocation through natural language dialogues; the system has been evaluated by staff responsible for bed management in a local hospital.

2 THE BASIS FOR ARGUMENTATION-BASED DIALOGUES

In our mechanism, agents argue using a subset of the speech acts found in the literature of argumentation-based dialogue [2, 33, 34]. The particular performative verbs used here and their informal meaning are as follows: (i) **assert**: an agent that performs this utterance declares, to all participants of the dialogue, that it is committed to defending this claim — the receivers of the message become aware of this commitment; (ii) **accept**: an agent that performs this utterance declares, to all participants of the dialogue, that it accepts the previous claim (assert) of another agent — the receivers of the message become aware of this acceptance; (iii) **question**: an agent that performs this utterance desires to know the reasons for a previous claim of another agent or, in case of an information-seeking dialogue, desires to know if the receiver can provide the information requested in the content of a *question* message; (iv) **challenge**: the receiver of the message, who previously committed to defending a claim, should now provide the support set for that claim; (v) **justify**: it is similar to the assert message but is used as a response to a challenge message previously received, whereby the agent provides the support to its previous claim.

We adopt the formal definition of the semantics of these speech acts from work by Panisson *et al.* [29, 31] which specify precisely the effect of the speech acts in the agent's mental state, as well as in the multi-agent dialogue as a whole[2]. The formal semantics allows for direct implementation of the effects of receiving and sending the speech-act in a BDI-based agent-oriented programming language based on the mental attitudes used in that specification [31]. From that work, we use the stated effects of each speech act on an agent's commitment store for the specification of our protocol, as described below. The Commitment Store (CS) consists of one or more structures, accessible to all agents in a dialogue, containing commitments made by the agents during the dialogue[3]. The CS is a subset of the knowledge base, and the union of the CSs can be viewed as the global state of the dialogue at a given time [34]. In the course of the dialogue, the agents use rules that define how the CS is updated. Such rules are part of the semantics used in this work. When an agent communicates, its CS is updated as follows: (i) *assert* (or *accept*): with the content p: $CS \leftarrow CS \cup \{p\}$; (ii) *question* and *challenge*: no effect on the CS; and (iv) *justify*: with the justified content contained in the set of rules and facts S (the support for a challenged claim p): $CS \leftarrow CS \cup S$;

Note that in our implementation, we support multi-agent interaction, so messages can be directed to a particular agent or to '$*$', which is used to denote all agents taking part in a particular dialogue. A message has the format `performative(sender, receiver, content)`. Besides the performative verbs used in individual messages, a *dialogue game protocol* restricts the moves allowed to agents. The dialogue game restricts the moves, but, as usual in such mechanisms, it also determines the alternative moves available to agents at any point in the multi-agent interaction. In fact, an interesting approach to determine an agent's individual strategy to participate in such interaction is through planning, as done, for example, in [6, 27].

The particular dialogue game approach we use in this paper is built upon fundamental ideas that appeared in [33, 34]. That work formalises the preconditions (called "rationality rules") for an agent to make each type of dialogue move and what commitment store updates ensue. Also, that work shows how those moves can be used to build dialogues for various purposes (see [39]), for example, information seeking, inquiry, or persuasion. Our case study in Section 4.3 shows in practice the sort of dialogue that the implementation of such rationality rules support. They provide the means for agents to engage in a dialogue, but our case study further shows when an agent chooses to move to an ontological subdialogue, following the rules we formally introduce in the next section.

3 FORMALISING MULTI-AGENT DIALOGUES WITH UNDERLYING ONTOLOGICAL AND ToM ARGUMENTS

We first informally present the structure of subdialogues we put forward in this paper, which can be seen in Figure 1. Agents engage in a dialogue about some subject (a claim put forward by the agent initiating the main dialogue). The dialogue proceeds normally following a particular protocol and using the knowledge base δ. In the case study reported here, for example, we use a multi-agent version of the dialogue protocol referred to in the previous section for both the main dialogue and each of the two types of subdialogues. What we formalise later in this section is precisely when an agent may digress from the main line of argumentation and move on to an ontological or ToM one. As seen in Figure 1, after a number of moves in either type of subdialogue, the agents involved in the dialogue must go back to discussing the main subject; that is, the

---

[1] *Ontological argumentation* term introduced in this paper refers specifically to multi-agent dialogues based on argumentation theory, where the content of the arguments being exchanged make explicit reference to a formal ontology.

[2] Due to space limitations, we cannot detail those formal definitions here, but the corresponding intuition given above is sufficient for understanding the approach proposed in this paper.

[3] CS is also referred to as *dialogue obligation store* in [22] and *dialogue store* in [35].

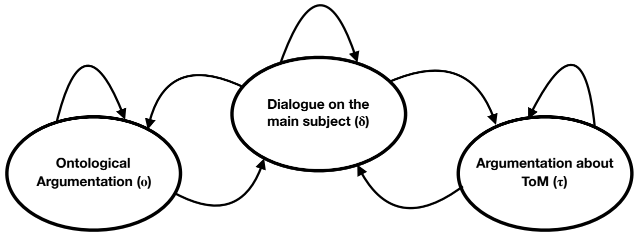

Figure 1: Dialogue Structure

main line of argumentation is suspended when a subdialogue starts, and it is only resumed when that subdialogue finishes.

The move towards a subdialogue is best explained by an example. Suppose we have $P(c)$ as a strict fact, $P(c) \wedge D(c) \longrightarrow Q(c)$ as a defeasible rule, $C(c)$ in the ABox, and $C \sqsubseteq D$ in the TBox of the o belief-base component. If, after asserting $Q(c)$, the agent is questioned about $D(c)$, the justification involves the ontological assertions. When presented with them, the other agent might disagree that $C(c)$ or disagree with the TBox statement if the ontologies are not correctly aligned. After that dialogue phase (i.e., a subdialogue) is finished, the main dialogue flow resumes. The result of the subdialogue, of course, will affect the main line of discussion. The agents may conclude the subdialogue by unanimously agreeing that $D(c)$, that $\neg D(c)$, or finishing the subdialogue inconclusively. In the latter case, the main dialogue will continue so that agents try to reach an agreement on the main subject despite being unable to agree on the ontological issue. Similarly, we might have a subdialogue to further inquire about ToM assumptions, in which case the subdialogue moves use knowledge from the ToM component. Support for ToM in our framework is done by incorporating the work on ToM for agent programming languages. Yet, those beliefs are particularly susceptible to being incorrect and incomplete. This is partly because of the intrinsic benevolence assumption in the rules for generating ToM but also because, in a dynamic environment, agent mental attitudes can change rapidly without further communication exchange that would have allowed for the ToM to be updated. Again, after a ToM subdialogue, the result will affect the main dialogue in the same ways mentioned above.

Our work includes the formalisation of a novel dialogue-subdialogue structure, using an existing protocol for each of the (sub)dialogues. Besides implementing the rules that support the dialogue protocol, our framework requires derivation of conclusions to be obtained for each of the three belief-base components when the agent needs to respond to a `challenge` message. For the defeasible component, the existing d-Prolog-based implementation already produces an AgentSpeak list with the sequence of rules used to derive a particular conclusion. For the ontology component, it is obtained from a description logic reasoner through an API integrated into our system. Finally, for the ToM component, it makes direct reference to the rules of the operational semantics that govern how ToM is updated in an agent language [32] that we incorporated into our system. However, this paper focuses on ontological subdialogues, particularly for the case study in Section 4.3.

3.1 Formalisation of Participating Agents

As seen in the previous section, our work builds on three other separate pieces of work in the literature: domain-specific strict and defeasible rules and facts, one or more ontologies, and a ToM (i.e., the information about other agents' state of mind that is kept updated through communication); note that all messages exchanged by agents may contribute to ToM updating, including the messages exchanged following the overall dialogue protocol we present in this section and the associated protocol governing (sub)dialogues. An agent in our framework is formalised as follows.

Definition 3.1 (Agent). An agent that takes part in our structured-dialogue argumentation protocol is defined as a tuple $\langle \delta, o, \tau, \pi, \varepsilon, \iota \rangle$, where δ is a set of defeasible and strict rules and facts (in the AgentSpeak style based on d-Prolog); o is a CooL-AgentSpeak style ontology-based belief base; τ is an AgentSpeak representation for ToM following the approach described in the previous section; π is the set of plans to achieve goals forming the agent's know-how (i.e., its plan library); ε is a set of AgentSpeak events which include, for example, recent goal adoptions (i.e., goals that are not yet intentions); and ι is the agent's current set of intentions (partially executed, partially instantiated plans to achieve goals).

Note that (δ, o, τ) are three now separate components replacing what would normally be simply one set of beliefs representing the agent's current belief base. We use C_i to refer to component C of agent i. Introducing further notation, we say that an agent can build an acceptable argument S that supports a claim p (denoted as $S \models p$) from one of its knowledge bases and the commitment store of the other participants. For example, agent i can build an acceptable argument S, which supports a conclusion p, from its defeasible knowledge base (δ_i) and the commitment store of j (CS_j) (denoted $(\delta_i \cup CS_j) \models S$).

3.2 Subdialogue Rules

We now introduce the rules governing the high-level dialogue structure, that is, the rules that allow agents to initiate the two types of subdialogues we would like them to have in our framework. They should be interpreted in the context of normal *dialogue rules* [20, 21] determining a protocol that governs the interactions between the agents, given their strategies whereby each agent moves by performing one of the utterances allowed by the protocol. Such rules, effectively determining a dialogue game [21], are often expressed as if-then rules, which are then easy to implement.

The dialogue rules specify the moves that each player can make, and so specify the *protocol* under which the dialogue takes place [2]. As mentioned before, the permitted moves in each (sub)dialogue follow the existing protocol discussed in Section 2. Instead of the usual if-then rules, we use a different style, similar to operational semantic rules, to formalise new performatives that are required to support the dialogue structure. In order to do so formally, we first define the overall dialogue setting.

Definition 3.2 (Subdialogue Game). A subdialogue game is formally represented as a tuple $\langle MD, SD_1, \ldots, SD_n, MS, DR \rangle$, where MD is the main dialogue, SD_i $(1 \leq i \leq n)$ are n possible subdialogues, MS is a finite set of allowed moves between any of the dialogues, and DR a set of dialogue rules governing the moves between the various (sub)dialogues. It is assumed in our model that digressing to a subdialogue suspends the dialogue on the main subject, which is only resumed when the subdialogue finishes.

We propose one particular subdialogue game, as follows.

Definition 3.3 (Ontological-ToM Subdialogue Game). An Ontological-ToM subdialogue game, denoted by SDG^{OT}, is formally defined by $\langle MD^{OT}, SD^{O}, SD^{T}, MS^{OT}, DR^{OT} \rangle$.

Arguments can be formed from the commitment store of the main dialogue and the knowledge in δ of each agent. The SD^{O} subdialogue uses o plus its commitment store and SD^{T} uses τ and another particular commitment store as well as ε and ι (so that the agent may refer to its own desires and intentions, as well as beliefs[4]). The formalisation of the two other components is given below in this section. First, we formalise a particular running instance of dialogue following our Ontological-ToM Subdialogue Game.

Definition 3.4 (Dialogue Instance). A particular dialogue instance following our Ontological-ToM Subdialogue Game is defined as $\langle dID, \mathcal{A}, SDG^{OT} \rangle$ where dID is a unique dialogue instance ID, $\mathcal{A}$ is the set of agents (in this paper we assume the same set of agents participates in the main as well as all subdialogues), and SDG^{OT} is as per Definition 3.3.

Definition 3.5 (Dialogue Moves). We denote a move in MS^{OT} as $v(i, j, \varphi)$, where v is the performative verb used for that move, made by agent i, addressed to agent j, regarding content φ. We consider the following set of performatives, denoted by P (see Section 2): `assert`, `accept`, `question`, `challenge`, `justify`, `closedialogue`, `ontoargsubdlg`, `tomsubdlg`, `closesubdlg`, and `failsubdlg`. The content of a move (φ) can be an argument (a set of formulæ) or just a formula (e.g., in an `assert` move, the content is a formula and in a `justify` move, the content will be a support set for a claim made in a previous `assert` move).

The dialogue rules in DR^{OT} indicate the possible moves that an agent can make following a previous move by another agent. They are presented here in the form of an inference rule in a similar presentation style as used in operational semantics of programming languages, except that here the conclusion part of the rule state which dialogue move (or *transition*) is allowed when the premises of the rule hold. A dialogue transition $l \longrightarrow r$ means making the r move in response to a previously received message l. When necessary to make that clear, a move r may be written r_M, r_O, or r_T depending on whether it took place in the main, ontological, or ToM (sub)dialogue. In the premises, existential quantification is assumed, and horizontal space between formulæ denotes conjunction. When multiple rules can fire, those are precisely the points where an individual agent strategy will determine how the dialogue unfolds (and as mentioned before, planning is one possible technique to help determine optimal dialogue strategies). We use $*$ to denote messages that are not directed towards a particular agent but to all agents taking part in the dialogue. The specific rules DR^{OT} that govern our subdialogue structure as follows.

$$\frac{f \in \delta_j \qquad C(t) \in f \qquad o \vdash C(t)}{\texttt{challenge}(i,j,f)_M \longrightarrow \texttt{ontoargsubdlg}(j,*,C(t))_O} \;(\textsc{OAsdlg1})$$

Rule OAsdlg1 says that if an agent challenges, in the main dialogue M, a formula in which $C(t)$ appears, and C is related to an ontology class, we can enter a subdialogue to discuss whether t indeed is an instance of class C. Rule OAsdlg2 is not shown because it is exactly like OAsdlg1 but for an ontology relation $R(t_1, t_2)$ rather than a class (line 17 of our example in Section 4.3 exemplifies the use of this rule). Note that it is assumed in the formalisation, without loss of generality, that the participating agents have only one ontology, which they have individually aligned using CooL-AgentSpeak. In practice, a `ontoargsubdlg` message could include a parameter for the URI of the particular OWL ontology referred to by the agent starting the subdialogue. When agents receive an `ontoargsubdlg` message, they know they have to switch their moves to a fresh instance of the subdialogue protocol.

$$\frac{\forall a \in \mathcal{A}.o_a \models \varphi}{\texttt{closedialogue}(i,*,\varphi)_O \longrightarrow \texttt{closesubdlg}(i,*,\varphi)_M} \;(\textsc{CloseOAsdlg1})$$

$$\frac{a \in \mathcal{A} \qquad b \in \mathcal{A} \qquad o_a \models \varphi \qquad o_b \models \neg\varphi}{\texttt{closedialogue}(i,*,\varphi)_O \longrightarrow \texttt{failsubdlg}(i,*,\varphi)_M} \;(\textsc{FailAOsdlg})$$

Rule CloseOAsdlg1 states that when the `closedialogue` performative is used by one of the agents to finish a dialogue which was an ontological subdialogue, that leads to the closing of the subdialogue with success (`closesubdlg`), in case all agents agreed on φ, and thereafter to the resuming of the main dialogue. Note that although we specify the condition from the point of view of the belief base of the participating agents, that can also be checked from the commitment stores of the subdialogue. Rule CloseOAsdlg2 is exactly like CloseOAsdlg1 except that it applies when all agents accept $\neg\varphi$ instead. It should also be noted that following a $\texttt{closesubdlg}(i, *, \varphi)$ message, the commitment store of the main dialogue is updated with the fact that now all agents accept φ (i.e., they reach an agreement about whether that ontological issue holds or not). When instead rule FailAOsdlg applies, the main dialogue is resumed with no alteration in the CS. The dialogue will have to continue despite the disagreement on φ.

The closing rules for ToM subdialogues are very similar, so for our purposes here, we only need to formalise the rules for starting a ToM subdialogue.

$$\frac{f \in \delta_j \qquad \mathsf{Mod}_{a\in\mathcal{A}}(\varphi) \in f \qquad \tau_j \vdash \mathsf{Mod}_{a\in\mathcal{A}}(\varphi)}{\texttt{challenge}(i,j,f)_M \longrightarrow \texttt{tomsubdlg}(j,*,\mathsf{Mod}_{a\in\mathcal{A}}(\varphi))_T} \;(\textsc{OTsdlg})$$

where $\mathsf{Mod} \in \{\mathsf{Bel}, \mathsf{Des}, \mathsf{Int}\}$. Rule OTsdlg says that if a formula f is challenged by an agent and that formula involves a subformula which is associated with the ToM component of the belief base, we may start a subdialogue to discuss specifically whether the mental attitude of a particular agent does in fact hold, i.e., there is a divergence between their ToMs.

Definition 3.6 (Divergence between agents' ToM). Considering two agents $i, j \in \mathcal{A}$, there is divergence between their ToM about some mental attitude $\mathsf{Mod}_k(\varphi)$, for some agent $k \in \mathcal{A}$, when $\tau_i \models \mathsf{Mod}_k(\varphi)$ and $\tau_j \not\models \mathsf{Mod}_k(\varphi)$.

We assume that agents have a consistent ToM about their own mental attitudes (they have perfect introspection about their own mental attitudes), i.e., $\forall\varphi \in \{\delta_i \cup \iota_i\}$ then $\mathsf{Mod}_i(\varphi) \in \tau_i$. Also, they have a consistent ToM about other agents, i.e., $\mathsf{Mod}_j(\varphi)$ and $\mathsf{Mod}_j(\neg\varphi)$ does not hold in τ_i simultaneously. Thus, we have the

---

[4] For a formalisation of the BDI modalities for AgentSpeak agents, see [8].

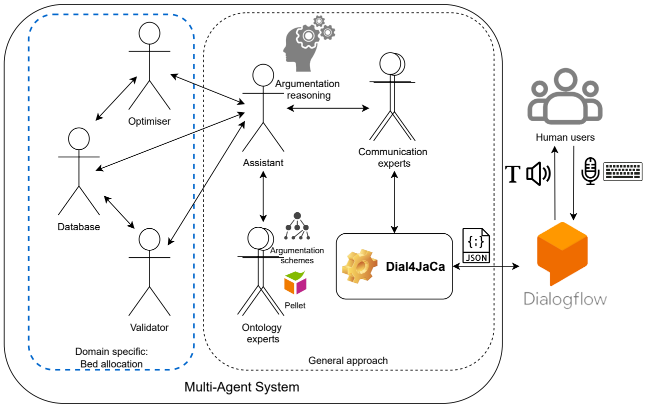

Figure 2: MAIDS Overview

following scenarios for ToM subdialogues: (i) When $j = k$, i.e., agent i has a divergent model about j's mental attitude $\text{Mod}_j(\varphi)$, agent j can inform its current mental attitude $\text{Mod}_j(\varphi)$ to i. (ii) When $i = k$, i.e., agent j has a divergent model about i's mental attitude $\text{Mod}_i(\varphi)$, agent i can inform its current mental attitude $\text{Mod}_i(\varphi)$ to j. (iii) When $j \neq k$ and $i \neq k$, i.e., agents i and j have a divergence about another agent k's mental attitude $\text{Mod}_k(\varphi)$, agents i and j may argue about the current mental attitude $\text{Mod}_k(\varphi)$ of k.

When the mental attitude causing a divergence between two agents' ToM refers to a belief (i.e., $\text{Mod}_i(\varphi) = \text{Bel}_i(\varphi)$), ToM subdialogues will be characterised as an *information-seeking (sub)dialogue* for cases (i) and (ii) above and an *inquiry (sub)dialogue* for case (iii). When the mental attitude causing such divergence refers to a desire or intention (e.g., $\text{Mod}_i(\varphi) = \text{Des}_i(\varphi)$), those ToM subdialogues may result in more complex interactions, possibly involving persuasion in case (iii). Such subdialogues about agents' desires/intentions are supported by carefully designed argumentation schemes recently introduced by D.Walton [38].

4 MAIDS

The Multi-Agent Intentional Dialogue System (MAIDS) framework combines argumentation theory techniques, ontology, and ToM to support complex dialogues. MAIDS supports the development of multi-agent applications to assist humans in decision making, including important components to develop complex MAS applications, such as: (i) dialogues in natural language to facilitate the interaction and adaptation of human operators; (ii) argumentation-based reasoning and dialogues, which allow agents to reason about and communicate well-supported information; (iii) ontologies to help agents to organise domain knowledge and perform semantic reasoning; and (iv) ToM to make agents remember previous interactions and make smarter communications. Figure 2 shows an overview of the MAIDS framework.

As it is shown in Figure 2, our framework relies on the use of Dial4JaCa [11] as an interface to dialogue platforms such as Dialogflow[5]. The `Human user` can interact with the chatbot through voice or text. This interaction is classified into intents by Dialogflow and sent to Dial4JaCa, which makes the request available to the `Communication expert` agent assigned to that specific user. One or more `Communication expert` agents can be instantiated, each one responsible for representing a `Human user`. This allows us to customise the responses given to the user based on a previously defined (or learned) profile. With this profile, the application avoids giving too many explanatory answers to a user with a specialist background and avoids giving very superficial answers to users without a specialist background since it translates the responses of the `Assistant` (the result of the MAS reasoning) into natural language messages, using templates as described in [26], to be sent to its corresponding `Human user`. Furthermore, the ability to instantiate multiple communication expert agents, one for each system's user, also allows an `Assistant` agent to engage in multiparty conversations, helping a team or a group of users make joint decisions. The `Assistant` agent is responsible for communicating with other agents in search of information and for performing argument reasoning. Besides multiple agents specialised in communication, several `Ontology expert` agents can be instantiated in MAIDS, allowing the MAS to consult several ontologies simultaneously. These agents can also perform ontological reasoning using the Pellet reasoner [37] and its open-source continuation effort Openllet[6]. In addition, these agents can translate OWL inference rules [17] automatically to defeasible rules (representing argumentation schemes) and use them during the reasoning process. These three types of agents, together with Dial4JaCa, make up our `General approach`.

In order to address the specificity of different application domains, domain-specific agents are added to the system. For example, in the instance shown in Figure 2, we added specific agents for the bed allocation domain we use to evaluate our framework (details about the evaluation will be presented in Section 5). Among those domain-specific agents, the `Validator` agent is responsible for validating bed allocation plans using a PDDL (Planning Domain Definition Language) plan validator; the `Optimiser` agent is responsible for making suggestions for optimised allocations using the GLPSol solver of GLPK[7] (GNU Linear Programming Kit), which is a free open source software for solving linear programming problems; and the `Database` agent is responsible for querying and updating the bed allocation system database.

4.1 A Multi-Part Belief Base

In MAIDS, agents communicate using an argumentation-based approach according to Section 2. Also, agents have a belief base with at least three main components. Each of these is based on work appearing in the literature, in particular: (i) a knowledge base of argumentation schemes; (ii) the CooL-AgentSpeak language which allows for the use of ontologies and ontology alignment; and (iii) recent work on ToM for AgentSpeak agents. Below, we describe each of these separate bases forming our *multi-part belief base*.

4.1.1 Argumentation-based Reasoning in Agent Programming. Our agents have an internal rule-based argumentation mechanism capable of generating (evolving) arguments. In this work, we use specifically the approach and implementation by Panisson et al. [30], which has been extended to consider the representation of argumentation schemes (i.e., reasoning patterns) for various applications

---

[5] https://cloud.google.com/dialogflow/es/docs

[6] https://github.com/Galigator/openllet

[7] http://winglpk.sourceforge.net/

domains [25, 28], which offers promising direction also for this work. Agents only accept propositions/claims which they do not have an acceptable argument against (i.e., they have a *cautious* attitude [33, 34]), and agents only assert propositions/claims for which they have an acceptable argument (i.e., a *thoughtful* attitude [33, 34]). In our dialogue approach, we will need to determine the acceptability of an argument from the agent's perspective (i.e., whether the agent does or does not have an argument for a given claim). That implementation referred to above and upon which we have built this component of our belief base provides that for us.

4.1.2 The CooL-AgentSpeak Language. CooL-AgentSpeak stands for "Cooperative description-Logic AgentSpeak" [19]. It resulted from various strands of past work on combining AgentSpeak with ontological reasoning [3, 18, 23], and has the following features: (i) it extends the AgentSpeak programming language with *ontological knowledge*, formally by means of a description logic, and in a practical implementation through the use of OWL ontologies; (ii) it has an explicit *cooperation strategy* to be used when agents exchange plans; and (iii) it takes advantage of *ontology matching* functions so that agents using different ontologies can communicate, in practice using available ontology matching services. Because it has all these features that are, in practice, important in multi-agent settings, we take that programming language as the basis for this component of the belief base that we require for our structured dialogue approach.

4.1.3 Theory of Mind in Agent Programming. The term Theory of Mind (ToM) is used to refer to the ability to model and reason about other agents' minds [14]. In this work, we take advantage of existing approaches to ToM in agent programming in order to model and reason about other agents' mental attitudes. Similar to ontological inquiries, in our approach, agents' ToM may also be the target of subdialogues, in which agents will argue about their own or other agents' mental attitudes. In fact, ToM subdialogues may be more often required than ontological ones, given how susceptible ToM is to being incorrect or incomplete. Even with probabilistic models, such as in [36], when an agent builds a model of other's minds, this model is often different from reality, given that there are many factors that can mislead the perception of the mental attitudes of others, and given that agents change their mental attitudes constantly, particularly in highly-dynamic multi-agent systems.

4.2 Expressivity of the Framework

Some desiderata for task-oriented dialogue systems have been recently formulated [10]. We summarise those desiderata below and give in parenthesis the lines of an example dialogue using our framework (shown in Section 4.3) where each of the features of the desiderata is demonstrated. The example also illustrates the ontological subdialogues supported by our framework.

(1) The system should allow the explicit representation of the user's desires that are implicit in requests such as in (1).
(2) The system should be able to represent the meaning of users' utterances in logical forms, including constraints having two superlative expressions, one embedded within the other as exemplified in (8 and 27).
(3) In the case of multiparty dialogues, it should keep track of the mental attitudes of all the involved participants as in (9 and 19).
(4) It is important to reason about plans and intentions, as it allows the system to be helpful by reasoning about what the user is trying to do, as in (18–25).
(5) It should reason about the meaning of mental attitudes as in (1 and 22).
(6) It should also represent beliefs of other agents without having precise information about what those beliefs are (9).

The idea behind such desiderata is to have a system that is fully explainable because everything it says has an explicitly represented plan being referred to by the system.

4.3 Example

We now reproduce some excerpts from a dialogue involving both humans and agents, including a version of the dialogue system that supports natural language interaction through the use of Dialogflow and has been developed and evaluated with the support of medical staff from a Brazilian Hospital. These excerpts exemplify the type of dialogues that can take place in systems developed with the approach put forward in this paper. They demonstrate the ontological discussions (in lines 17–17p) and the desired features discussed above. For simplicity, and due to the lack of space, we only explicitly show a few messages communicated by the agents in our case study, the ones that relate to the desiderata by Cohen [10], and we only describe the remaining dialogue parts succinctly. However, the complete dialogue is available online[8].

This case study includes the following agents: **assistant (a)**: the internal representation in MAS for a chatbot that assists hospital staff in carrying out bed allocation in a hospital; **operator (o)**: the internal representation in MAS for the hospital staff member who operates the system for allocating beds; **nurse (n)**: the internal representation in MAS for a nurse who in that hospital serves as domain expert for bed allocation and whom the operator needs to consult in case of doubt; **database (d)**: an agent that has access to the hospital's general information system for checking details of past and current patients, bed allocations, etc. **ontology (on)**: an agent expert in ontologies, responsible for semantic reasoning using argumentation schemes as defeasible rules generated automatically from the semantic rules contained in the ontology. **optimiser (op)**: an agent responsible for making suggestions for optimised allocations using the GLPSol solver of GLPK.

The dialogue starts with the operator trying to allocate a bed to a particular patient and proceeds as follows. We show each (numbered) dialogue game move, but *before* it we provide an English equivalent for readability. We enclose in curly brackets the changing beliefs of some of the agents which underlie the dialogue move. Note that our approach only allows for atomic formulæ in argument conclusions, but it allows for constraints on a particular conclusion to be specified using Jason annotations, so if a dialogue move contains a formula `p(X)[q(X)]`, it means that in Jason we will find an instantiation for `X` such that `p(X) & q(X)` holds.

operator to assistant: check if any female surgical bed is free;
`1. question(o,a,free(B)[female(B),surgical(B)])`
{assistant: `des(o,allocate(P,B)[female(B), surgical(B)])`,
`bel(d,free(B))`} (desiderata (1 and 5))

---

[8] https://github.com/smart-pucrs/MAIDS-bed-allocation/blob/main/AAMAS2023_MAIDS.pdf

... the assistant checks with the database agent if any female surgical bed is free. The database agent responds that bed 203b is available. The assistant provides that information to the operator ...

operator to nurse: I'm allocating Patient8 to 203b;

`5. assert(o,n,allocate(patient8,203b))`

... the nurse refuses justifying their position ...

nurse to operator: this bed is in a room that has many beds, for Patient8 we need the smallest room with the fewest occupied beds;

`8. justify(n,*,[defeasible_rule(~allocate(patient8,203b), [large(203),in_room(203b,203)])[as(nurse_statement)], defeasible_rule(allocate(patient8,B),[in_room(B,R),smallest(R) [fewest_occupants(R)]])[as(nurse_restriction)]])`(desiderata (2))

{assistant: `des(n,allocate(patient8,B)[female(B), surgical(B), in_room(B,R), smallest(R) [fewest_occupants(R)]]), defeasible_rule(allocate(patient8,B), [in_room(B,R), smallest(R) [fewest_occupants(R)]]) [as(nurse_restriction)]`

operator to assistant: how about allocating Patient8 to the bed that was freed yesterday by Patient6;

`9. assert(o,a,allocate(patient8,B)[allocated(patient6,B,TI,TF), within_time(yesterday,TI,TF)])`

{assistant: `des(n,allocate(patient8,B)[female(B), surgical(B), in_room(B,R), smallest(R) [fewest_occupants(R)]]), des(o,allocate(patient8,B) [allocated(patient6,B,TI,TF), within_time(yesterday,TI,TF)]), bel(d,allocated(P,B,TI,TF))`} (desiderata (3 and 6))

... the assistant, with other agents' help, found out that Patient6 was allocated to bed 202b yesterday, and this bed is unsuitable for Patient8. After informing the operator agent, the operator requests an explanation (executing a challenge move) about why bed 202b is unsuitable for Patient8. Considering that this information comes from the ontology inference rules, they enter an ontological subdialogue ...

entering an ontological subdialogue using OAsdlg2

`17. ontoargsubdlg(a,*,~suitable(202b,patient8))`

... the assistant asks for an explanation from the ontology agent, receives the answer and sends it to all. The operator agent questions why bed 202a is in the adolescent age group (i.e., a challenge move). After asking the ontology agent and receiving the answer, the assistant informs everyone ...

nurse to all: we can make an exception in this case, they can stay in the same room provided they are of the same gender and same type of care;

`17h. assert(n,*,defeasible_rule(suitable(B,patient8), [patient(patient8),bed(B),is_of_the_gender(patient8,G), bed_is_of_gender(B,G),is_of_care(patient8,C), bed_is_of_care(B,C)])[as(nurse_exception)])`

{assistant: `des(n,defeasible_rule(suitable(B, patient8), [patient(patient8), bed(B), is_of_the_gender(patient8,G), bed_is_of_gender(B,G), is_of_care(patient8,C), bed_is_of_care(B,C)]) [as(nurse_exception)])`}

... based on a response from the ontology agent, the assistant informs everyone that bed 202b is not yet suitable for the patient Patient8, justifying its position ...

assistant to all: Patient Patient8 is of Intensive care and bed 202b is of Minimal care that is different from Intensive care. so bed 202b is unsuitable for patient Patient8

`17o. justify(a,*,defeasible_rule( is_unsuitable_for(202b, patient8),[patient(patient8),hospital_bed(202b), is_care(patient8,intensive),bed_is_care(202b,minimal), differentFrom(intensive,minimal)])[as(nSbyCare)]`

... all agents agree bed 202b is unsuitable for patient Patient8 and the ontological subdialogue is closed ...

{assistant:`des(o,allocate(patient8,B) [female(B),surgical(B)])`}

assistant to operator: Would you like me to try to suggest another bed for Patient8?

`18. question(a,o,des(o,suggestion(B,patient8)[suitable(B, patient8)]))`(desiderata (4))

operator to assistant: yes, please!

`19. assert(o,a,des(o,suggestion(B,patient8)[suitable(B, patient8)]))`

{assistant:`des(n,defeasible_rule(allocate(patient8,B), [in_room(B,R) ,smallest(R) [fewest_occupants(R)]]]) [as(nurse_restriction)]), des(o,suggestion(B,patient8) [suitable(B,patient8)])` (desiderata (3))

... the assistant asks for a suggestion from the optimiser agent considering the restriction imposed by the nurse, but it answers that it has no suggestion considering this restriction ...

{assistant:`des(n,defeasible_rule(suitable(B, patient8), [patient(patient8),bed(B), is_of_the_gender(patient8,G), bed_is_of_gender(B,G), is_of_care(patient8,C), bed_is_of_care(B,C)]) [as(nurse_exception)])`}

assistant to all: can I use the exception made by nurse?

`22. question(a,*,des(o,suggestion(B,patient8)[suitable(B, patient8),defeasible_rule(Ce,Re)[as(nurse_exception)]]))` (desiderata (5))

nurse to all: yes, you can.

`23. assert(n,*,des(o,suggestion(B,patient8)[suitable(B, patient8),defeasible_rule(Ce,Re)[as(nurse_exception)]]))`

... the assistant sends the exception made by the nurse to the optimiser agent and asks for an allocation considering this exception. The optimiser suggests bed 201a, and the assistant passes the suggestion on to everyone ...

{assistant: `suitable(201a,patient8)`}

assistant to all: Considering the exception made by nurse I suggest allocating Patient8 to bed 201a

`26. assert(a,*,suggestion(201a,patient8)[suitable(201a, patient8),[defeasible_rule(Ce,Re)[as(nurse_exception)]])`

operator to assistant: ok, please book bed 201a for Patient8 who will leave the operation room not before 19:00 nor after 20:30;

`27. assert(o,a,booked(201a,patient8,19:00h,20:30h))` (desiderata (2))

{assistant: `des(o,booked(201a, patient8, 19:00h, 20:30h))`}

... the dialogue ends booking bed 201a to patient Patient8.

5 EVALUATION USING A BED ALLOCATION SCENARIO

A hospital in Brazil has kindly agreed to support us in evaluating our system. We started an evaluation process with the help of some professionals responsible for bed management in that hospital, seeking to assess whether changes would be necessary to adapt the dialogue system instance created from the MAIDS framework to be used with real data from that hospital. For the first phase of the evaluation, we fed the web interface with fictitious data about beds and patients. Then, we asked that professionals use the simulator to check out the fictitious hospital situation and ask the chatbot to validate the bed allocation they created, give suggestions, evaluate the availability of a bed related to a specific patient, and explain the statements put forward. After, we asked the professionals to evaluate the answers that the chatbot gave, also performing a questionnaire to collect their opinion about the use of the system. All professionals signed a consent form for participation.

Two hospital staff responded to our questionnaire. The first one has been a bed management administrator for nine years. Moreover, the second one has been the medical coordinator in this hospital for one year and is one of the doctors who assisted in the construction of a manual for the implantation and implementation of the internal regulation committee (including bed-allocation rules) for general and specialised hospitals used by many hospitals in the country.

Among the questions asked in the questionnaire, some sought to understand whether the rules for allocating beds used by our agents followed the rules currently used in the hospital. We concluded that some rules would need to be added, for example, related to patients with infection, information about health insurance plans or health plans, and information sent by the bed requesting unit. Due to inconsistencies between the rules used by the agents and those used in the hospital, the interaction with the chatbot was also compromised since the explanations it gave sometimes did not match the reasons used in real life. On the other hand, both professionals agree that the answers given by the chatbot are easily understandable. In addition, they also agree that when asked if a bed is suitable for a patient, the chatbot can answer and also explain how it reached that conclusion in an easily understandable way.

As a consequence of this evaluation, the managers of the local (university) hospital have asked us to help deploy our multi-agent system to be used in their daily bed management activities as soon as we can interface it with the information systems currently used in the hospital. After proceeding with the adjustments recommended by the professionals, adapting the rules used by our agents to those practised in the hospital, and adjusting the tasks that the chatbots can perform according to the requests made by the evaluators, we intend to carry out a new evaluation, this time using real historical bed and patient data. After this validation, we will proceed with the integration with the system currently used by the hospital so that operators can use a pilot of our system in their daily activities.

6 RELATED AND FUTURE WORK

The only work that supports agents arguing about OWL ontologies specifically, to the best of our knowledge, appeared in [24]. However, that work was not formalised in the context of an agent programming language, and did not support ToM, nor the structured dialogue approach we introduced in this paper. Furthermore, that framework does not seem to have been further developed and does not seem to be available for download, so it does not support the development of practical dialogue systems like ours. In fact, we are not aware of any practical agent framework that supports all the features of dialogue systems supported by our framework.

There is much work on allowing for defeasibility in description logic and OWL [5, 13], but this is also distant from our work in that it does not provide practical support for agent programming with argumentation-based dialogues.

Much related work in the area of argumentation was already cited throughout the paper, but it is worth mentioning at least that although there is work on nested dialogues [7], the possibility to digress about ontological and ToM issues in subdialogues as put forward in this paper is completely original.

Another strand of work in argumentation to mention here, because it points to one of our main future works, is on using automated planning techniques to support an agent's strategy in taking part in dialogue games [6]. We aim to apply this to decide when to move to subdialogues (currently, for the case study, we used a simple strategy, one that moves to subdialogues as soon as possible). Future work also includes allowing only subsets of the agents entering into one of the subdialogues, further developing the applications so they also use the ToM subdialogues, and experimenting with our framework to develop dialogue systems in other hospital management domains besides bed allocation.

However, it is worth mentioning that such a sophisticated combination of components used to achieve the dialogue presented in this paper also provides the means for the development of sophisticated methods for human-agent interaction in the context of Hybrid Intelligence [1] (where the need for such interactions are very evident) and eXplainable Artificial Intelligence (XAI) [15, 16].

In the context of Hybrid Intelligence, as described in [1], it requires humans and intelligent systems working together, and one of the key challenges to achieving this partnership is the capability of agents to understand human actors (which also requires a ToM about them). Our framework supports such an understanding of the users by combining the ToM component described in Section 4.1.3 plus the ToM subdialogues, with which agents are able to argue about the users' mental attitudes. In the context of XAI, as described in [4], there is little work addressing the issues of multi-agent explainability, personalisation of explanation, and context awareness. Our framework allows agents to engage in argumentation-based dialogues to support bed allocation, which makes them aware of other agents' reasons/justifications/opinions about a particular bed allocation, so interface agents are able to provide argumentation-based explanations to users, resulting from the collective construction of such arguments. In the line of the work on XAI, thanks to the ToM component and the understanding of the users supported by it, agents would be able to personalise argumentation-based explanations, for example, omitting information that agents know the user already knows, making the communication more concise.

7 CONCLUSIONS

In this paper, we have proposed a multi-part belief base for a BDI agent programming language and a structured approach to dialogues where agents argue about the main belief base component but can move on to subdialogues to discuss specific issues related to the ontological component or the ToM component of the multi-part belief base. With an example dialogue, we have shown that our current implementation[9] covers the features recently put forward as desiderata for future dialogue systems (i.e., that current popular dialogue platforms do not address), and the ontological and ToM "digressions" give even further expressivity on top of that. Although much work remains to be done, as discussed in the previous section, in its current state our framework already indicates a concrete way towards a high level of sophistication in explainable AI, hybrid intelligence, and human-agent dialogue systems.

---

[9] MAIDS implementation has been supported by several open-source technologies such as Jason platform [9], interfaces with ontologies [12], argumentation-based reasoning mechanism [28], ToM reasoning mechanism [32]. However, putting together such pieces of code and implementing the basic multi-agent dialogue game, as well as the dialogue structure formalised in this paper on top of them, was by no means a straightforward engineering task. Due to lack of space, we do not give further details of the implementation here but refer the interested reader to https://github.com/smart-pucrs/MAIDS-bed-allocation.git where all the source code for the programming framework on top of Jason as well as the domain rules supporting the dialogue shown in 4.3 can be downloaded.

ACKNOWLEDGMENTS

This research was partially funded by CNPq, CAPES, FCT CEECIND /01997/2017 and UIDB/00057/2020.

REFERENCES

[1] Zeynep Akata, Dan Balliet, Maarten de Rijke, Frank Dignum, Virginia Dignum, Guszti Eiben, Antske Fokkens, Davide Grossi, Koen Hindriks, Holger Hoos, et al. 2020. A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect With Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence. *Computer* 53, 8 (2020), 18–28.

[2] Leila Amgoud, Nicolas Maudet, and Simon Parsons. 2000. Modeling Dialogues Using Argumentation.. In *ICMAS*. IEEE Computer Society, 31–38.

[3] Davide Ancona, Viviana Mascardi, Jomi Fred Hübner, and Rafael H. Bordini. 2004. Coo-AgentSpeak: Cooperation in AgentSpeak through Plan Exchange. In *3rd International Joint Conference on Autonomous Agents and Multiagent Systems, New York, USA*. IEEE Computer Society, 696–705.

[4] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*. International Foundation for Autonomous Agents and Multiagent Systems, 1078–1088.

[5] Grigoris Antoniou and Gerd Wagner. 2003. Rules and Defeasible Reasoning on the Semantic Web. In *Rules and Rule Markup Languages for the Semantic Web, Second International Workshop, RuleML 2003, Sanibel Island, FL, USA, October 20, 2003, Proceedings*. 111–120.

[6] Elizabeth Black, Amanda Jane Coles, and Christopher Hampson. 2017. Planning for Persuasion. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*. 933–942.

[7] Elizabeth Black and Anthony Hunter. 2007. A generative inquiry dialogue system. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 241.

[8] Rafael H. Bordini, Michael Fisher, Willem Visser, and Michael J. Wooldridge. 2006. Verifying Multi-agent Programs by Model Checking. *Autonomous Agents and Multi-Agent Systems* 12, 2 (2006), 239–256.

[9] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons.

[10] Philip Cohen. 2019. Foundations of Collaborative Task-Oriented Dialogue: What's in a Slot?. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, Stockholm, Sweden, 198–209.

[11] Débora Engelmann, Juliana Damasio, Tabajara Krausburg, Olimar Borges, Mateus Colissi, Alison R Panisson, and Rafael H Bordini. 2021. Dial4JaCa–a communication interface between multi-agent systems and chatbots. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 77–88.

[12] Artur Freitas, Alison R Panisson, Lucas Hilgert, Felipe Meneguzzi, Renata Vieira, and Rafael H Bordini. 2017. Applying ontologies to the development and execution of multi-agent systems. In *Web Intelligence*, Vol. 15(4). IOS Press, 291–302.

[13] Laura Giordano and Valentina Gliozzi. 2019. Strengthening the Rational Closure for Description Logics: An Overview. In *Proceedings of the 34th Italian Conference on Computational Logic, Trieste, Italy, June 19-21, 2019 (CEUR Workshop Proceedings, Vol. 2396)*, Alberto Casagrande and Eugenio G. Omodeo (Eds.). CEUR-WS.org, 68–81.

[14] Alvin I Goldman et al. 2012. Theory of mind. *The Oxford handbook of philosophy of cognitive science* 1 (2012), 1–25.

[15] David Gunning. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web* 2 (2017), 2.

[16] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. 2019. XAI—Explainable artificial intelligence. *Science Robotics* 4, 37 (2019).

[17] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. 2004. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission* 21, 79 (2004), 1–31.

[18] Thomas Klapiscak and Rafael H. Bordini. 2008. JASDL: A Practical Programming Approach Combining Agent and Semantic Web Technologies. In *Declarative Agent Languages and Technologies VI, 6th International Workshop, DALT 2008, Estoril, Portugal, May 12, 2008, Revised Selected and Invited Papers*. 91–110.

[19] Viviana Mascardi, Davide Ancona, Matteo Barbieri, Rafael H Bordini, and Alessandro Ricci. 2014. CooL-AgentSpeak: Endowing AgentSpeak-DL agents with plan exchange and ontology services. *Web Intelligence and Agent Systems: An International Journal* 12, 1 (2014), 83–107.

[20] Peter McBurney and Simon Parsons. 2001. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information* 11 (2001), 2002.

[21] Peter McBurney and Simon Parsons. 2002. Dialogue Games in Multi-Agent Systems. *Informal Logic* 22 (2002), 2002.

[22] Peter McBurney and Simon Parsons. 2005. Locutions for argumentation in agent interaction protocols. In *Agent Communication*. Springer, 209–225.

[23] Álvaro F. Moreira, Renata Vieira, Rafael H. Bordini, and Jomi Fred Hübner. 2005. Agent-Oriented Programming with Underlying Ontological Reasoning. In *Declarative Agent Languages and Technologies III, Third International Workshop, DALT 2005, Utrecht, The Netherlands, July 25, 2005, Selected and Revised Papers*. 155–170.

[24] Andrés Muñoz and Juan A. Botía. 2008. ASBO: Argumentation System Based on Ontologies. In *Cooperative Information Agents XII, 12th International Workshop, CIA 2008, Prague, Czech Republic, September 10-12, 2008. Proceedings*. 191–205.

[25] Alison R Panisson and Rafael H Bordini. 2017. Argumentation schemes in multi-agent systems: A social perspective. In *International Workshop on Engineering Multi-Agent Systems*. Springer, 92–108.

[26] Alison R Panisson, Débora C Engelmann, and Rafael H Bordini. 2022. Engineering explainable agents: an argumentation-based approach. In *Engineering Multi-Agent Systems: 9th International Workshop, EMAS 2021, Virtual Event, May 3–4, 2021, Revised Selected Papers*. Springer, 273–291.

[27] Alison R Panisson, Giovani Farias, Artur Freitas, Felipe Meneguzzi, Renata Vieira, and Rafael H Bordini. 2014. Planning interactions for agents in argumentation-based negotiation. In *Proc. of 11th Int. Workshop on Argumentation in Multi-Agent Systems*. 1–15.

[28] Alison R Panisson, Peter McBurney, and Rafael H Bordini. 2021. A computational model of argumentation schemes for multi-agent systems. *Argument & Computation* Preprint (2021), 1–39.

[29] Alison R. Panisson, Felipe Meneguzzi, Moser Silva Fagundes, Renata Vieira, and Rafael H. Bordini. 2014. Formal semantics of speech acts for argumentative dialogues. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*. 1437–1438.

[30] Alison R Panisson, Felipe Meneguzzi, Renata Vieira, and Rafael H Bordini. 2014. An approach for argumentation-based reasoning using defeasible logic in multi-agent programming languages. In *11th International Workshop on Argumentation in Multiagent Systems*. 1–15.

[31] Alison R. Panisson, Felipe Meneguzzi, Renata Vieira, and Rafael H. Bordini. 2015. Towards Practical Argumentation in Multi-agent Systems. In *2015 Brazilian Conference on Intelligent Systems, BRACIS 2015, Natal, Brazil, November 4-7, 2015*. 98–103.

[32] Alison R. Panisson, Stefan Sarkadi, Peter McBurney, Simon Parsons, and Rafael H. Bordini. 2018. On the Formal Semantics of Theory of Mind in Agent Communication. In *Agreement Technologies - 6th International Conference, AT 2018, Bergen, Norway, December 6-7, 2018, Revised Selected Papers*. 18–32.

[33] Simon Parsons and Peter McBurney. 2003. Argumentation-based dialogues for agent co-ordination. *Group Decision and Negotiation* 12, 5 (2003), 415–439.

[34] Simon Parsons, Michael Wooldridge, and Leila Amgoud. 2002. An Analysis of Formal Inter-Agent Dialogues. In *In 1st International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press, 394–401.

[35] Fariba Sadri, Francesca Toni, and Paolo Torroni. 2001. Logic Agents, Dialogues and Negotiation: An Abductive Approach. In *Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce*. AISB, 1–8.

[36] Ştefan Sarkadi, Alison R Panisson, Rafael H Bordini, Peter McBurney, and Simon Parsons. 2018. Towards an approach for modelling uncertain theory of mind in multi-agent systems. In *International Conference on Agreement Technologies*. Springer, 3–17.

[37] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. 2007. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics* 5, 2 (2007), 51 – 53. Software Engineering and the Semantic Web.

[38] Douglas Walton. 2019. Using argumentation schemes to find motives and intentions of a rational agent. *Argument & Computation* Preprint (2019), 1–43.

[39] Douglas N Walton. 1998. *The new dialectic: Conversational contexts of argument*. University of Toronto Press.

[40] Michael J. Wooldridge. 2000. Computationally Grounded Theories of Agency. In *4th International Conference on Multi-Agent Systems, ICMAS 2000, Boston, MA, USA, July 10-12, 2000*. IEEE Computer Society, 13–22.