A Formal Model of Persuasion Dialogs for Interactions among Argumentative Software Agents

Andrés Muñoz and Juan A. Botía
Dpto. de Ingenería para la Información y las Comunicaciones
University of Murcia, Spain, C.P. 30100
E-mail: {amunoz,juanbot}@um.es

Abstract—**Multi-agent systems (MAS) offer an approach to solve complex problems where data and control are inherently distributed among several agents. Normally, agents have to cope with conflicts based on inconsistent knowledge which arise when they exchange information. A relative common alternative to solve these conflicts is Argumentation. This technique allows agents to develop a coordination process based on the exchange of justifications supporting a piece of knowledge, with the aim of proving its validness. However, most argumentation frameworks are theoretical approaches to this problem. We have developed an Argumentation System Based on Ontologies (ASBO). It follows an engineering-oriented approach to materialize a software architecture which allows working with argumentation in MAS. Moreover, ASBO has also a formal model in the background. This paper introduces such formal model, in order to identify and unambiguously define the core elements that argumentation systems should include.**

Index Terms—**Multi-agent systems engineering, Knowledge management, Argumentation, Persuasion dialogs.**

I. Introduction

AGENT-oriented paradigm [1] has been recognized as a powerful technique for modeling scenarios where a sophisticated software program autonomously acts on behalf of the users in order to fulfill their demands. As the complexity of the problems modeled in this paradigm grows, it is required that multiple agents can work together [2]–[4]. As a result, multi-agent system (MAS) engineering has been developed as a means of coordinating agents [5]. One particularity in MAS approaches is that no agent holds a complete vision of the problem faced (i.e., data and control are distributed). In these systems, the beliefs of one agent compound the personal view that such an agent has with respect to the part of the problem it is in charge of. Due to the distributed nature of multi-agent systems, these beliefs may be incomplete and overlapped with other agents' beliefs.

In this context, knowledge conflicts [6], [7] may arise from the situation where several agents have their own local knowledge base with incomplete and different information about the same domain. Moreover, each agent is responsible for maintaining the coherence of its knowledge base. Then, whenever agents exchange information, inconsistencies may arise in their local knowledge bases due to these different and/or incomplete agents' views of the state of affairs (e.g., a network administration scenario where some agents may believe that the access to a resource is granted, whereas others assert it as forbidden). As a result, agents must cope with these inconsistences as conflicts based on knowledge.

Classifications of knowledge conflicts usually divide them into two general types [8]: syntactic conflicts, so-called *contradictions*, and semantic ones. Contradictions may appear regardless how the domain is modeled (i.e., a fact and its negation, such as the network administration example above). Contrarily, semantic conflicts stick to the considered domain (e.g., classifying an object as square and rectangular in a domain where objects are restricted to be classified by a unique shape).

Although much of the work on solving knowledge conflicts in MAS has primarily been focused on syntactic conflicts [9]–[11], current proposals are also taken semantic conflicts into consideration [12], [13]. Note that this type of conflicts is more difficult to detect than syntactic ones, since they must be analyzed with respect to the meanings of a specific domain. We have proposed an approach for enabling agents to detect and solve both types of conflicts indistinctly. This approach consists in the combination of a formal knowledge model represented through ontologies and a mechanism for managing conflicts based on argumentation techniques. As a result, an argumentation system based on ontologies (ASBO henceforth) has been developed [14]. The elements of ASBO are briefly introduced here.

Firstly, the universe of discourse (i.e., the domain knowledge) is represented by means of a formal model based on Semantic Web [15] technologies. More precisely, we refer to OWL (Web Ontology Language) ontologies [16]. In this manner, the agents' beliefs are expressed as instances of these ontologies. Due to the addition of meta-information on this model, agents can support reasoning operations by means of inference engines (e.g., Pellet [17] or Jena [18]). Examples of reasoning operations are deductive processes to entail new knowledge, or consistency checking of ontologies to detect inconsistencies. Thus, detection of both semantic and syntactic knowledge conflicts are supported by these operations in a straightforward manner.

A common approach to coordinate agents to autonomously solve conflicts is by reaching agreements about the status of those conflicts. To this end, the alternative followed in ASBO resides in employing a *persuasion* dialog [19], [20]. Such kind of dialog consists in an exchange of opinions among agents that are for/against a piece of conflictive knowledge, with the aim of clarifying which opinion is the most acceptable.

Argumentation [21]–[23] is considered as a promising materialization of persuasion dialogs in multi-agent systems. In this manner, a negotiation protocol is defined via argumentation [24], which leads to a persuasion dialog where an agent tries to convince others about a specific proposal. But in this case, not only are the proposals exchanged. Furthermore, *arguments* (i.e. premises and rules used to derive those proposals) are also communicated. It allows agents to resolve conflicts more efficiently than just exchanging proposals, as proved elsewhere [25].

Different attacks (i.e., conflictive points of view on proposals, premises and rules) can be defined over arguments. These attacks could in turn be supported by other arguments, which may also be attacked. Hence, an argumentation process takes place by starting a persuasion dialog where agents exchange their arguments. Eventually, an acceptability status is determined for each argument depending on the development of the persuasion dialog. As a result, if an argument supporting a conflicting proposal is *accepted* by the agents involved in the dialog (i.e., the argument has no attacks or all its possible attacks have been invalidated by other arguments), such a proposal is also accepted. Contrarily, a *defeated* argument (i.e., an argument attacked by an accepted one) means that its proposal is not accepted.

While details on the above elements can be found elsewhere [14], [26], this paper is devoted to formalize the persuasion dialog framework used in ASBO for exchanging arguments. In this manner, we identify and formally define the elements which should be present in persuasion dialogs developed within an argumentation process. Section II presents the formal model of such a framework. Section III illustrates how this framework is used through an example. Section IV outlines most important conclusions and future works.

II. Formalization of the persuasion dialog framework in ASBO

The ASBO approach is restricted to the world of software agents. Its main goal is to provide them with an effective mechanism to solve knowledge conflicts by exchanging arguments through a persuasion dialog. ASBO agents are structured in a layered manner, from an architectural point of view. Figure 1 shows a representation of a couple of ASBO agents based on a block diagram. The top layer is related to argumentation tasks. It includes the definitions of the argumentation system (i.e., definitions of arguments and relationships of attack and defeat between them) and the persuasion dialog framework. The middle layer constitutes the formal description model of the universe of discourse. It contains a common representation (i.e., OWL ontologies) to express the agents' beliefs. Thus, this layer includes the domain knowledge upon which arguments are built. The bottom layer encapsulates a particular implementation for a specific agent platform.

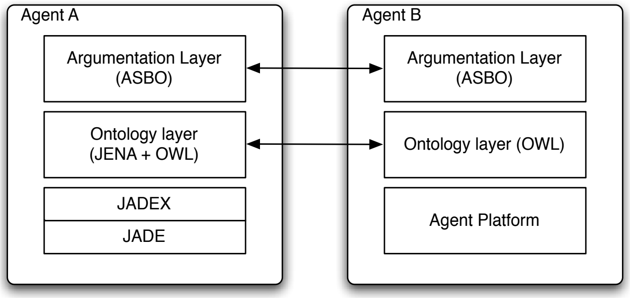
Agent A
Argumentation Layer (ASBO)
Ontology layer (JENA + OWL)
JADEX
JADE
Agent B
Argumentation Layer (ASBO)
Ontology layer (OWL)
Agent Platform

Fig. 1. Two ASBO agents showing the layered disposition of elements.

Details about the definition of the argumentation system are given in [14], whereas the middle and bottom layers are explained in [26]. The rest of the section is devoted to the persuasion dialog framework in the Argumentation layer, by giving a formal description of the communication language, the interaction protocol, the context and state of the dialog, the effect rules and the termination and outcome conditions. The selection of these elements is based on the Prakken's framework [20] and they are formalized here to define the persuasion dialogs hold in ASBO.

A. The communication language

The first element that needs to be defined in the ASBO persuasion dialog framework is a communication language for sending and receiving beliefs and the arguments supporting them. In this paper, the ASBO communication language is defined as the set of messages which can be exchanged among agents to this end. It will be denoted by $ASBO_{cm}$.

Table I summarizes all the messages defined to enable persuasion dialogs in ASBO. They are defined with (1) the form of the utterance (i.e., the performative and message body), (2) the semantic of the utterance (i.e., the *locutionary* force), (3) the intention of the emitter (i.e., the *illocutionary* act) and (4) the effect on the receiver (i.e., the *perlocutionary* effect).

Each performative ($Claim$, Why, $Since$, $\ldots$) defines the type of communicative act of the $ASBO_{cm}$ messages, explained below. On the other hand, the message body is formed by either assertions (i.e., agents' beliefs) and arguments. Assertions are denoted by φ, whereas arguments are denoted by S. An argument is defined as

$$S = (\varphi, \Phi),$$

where φ is the assertion representing the conclusion of the argument, and Φ is the support set which justifies that conclusion. In turn, Φ is composed of assertions called *premises*, denoted here by σ. The language for expressing both assertions and arguments is given by the Ontology layer in figure 1, namely OWL.

Let us see now the intended communicative act of each message. $[Claim\ \varphi]$ allows an agent to communicate the assertion φ to other agents, with the aim of convincing them about its validness. $[Why\ \varphi]$ enables an agent to ask for arguments about φ. Each argument is communicated through $[\varphi\ Since\ S]$. The message $[Concede\ \varphi]$ is used to inform that φ is accepted by the emitter. On the other hand, $[Retract\ \varphi]$ allows an agent to withdraw from supporting φ when such an agent does not have any argument for it or its arguments

TABLE I
AVAILABLE MESSAGES IN $ASBO_{cm}$

(1) Utterance	**(2) Literal meaning**	**(3) Intention of the emitter**	**(4) Effect in the receiver**
Claim φ	Statement of assertion φ	To impose φ to other agents	The speaker is associated to φ
Why φ	Challenge of assertion φ, seeking for arguments supporting it	To obtain arguments for φ	The assertion φ must be justified with arguments
φ **Since** S $(S = (\varphi, \Phi))$	Disclosure of an argument S which supports φ	To prove φ as a justified assertion	The speaker is associated to premises σ_i in $\Phi, i = 1..n$
Concede φ	Assumption of assertion φ	To announce that the speaker agrees to φ	The speaker is associated to φ
Retract φ	Rejection of assertion φ	To withdraw from φ because no valid argument can be found for it	The speaker withdraws from the assertion φ
Accept S $(S = (\varphi, \Phi))$	Acceptation of argument S	To update the speaker's knowledge according to the argument S	The speaker accepts the conclusion φ supported by S
No-Response m	Withdrawal from answering a message m	To announce that the speaker has no valid responses to m	The speaker can not respond a message m

are not valid anymore (i.e., they have been defeated by other arguments during the persuasion dialog). The message $[Accept\ S]$ is used when the emitter agrees to the argument S. Finally, the message $[No\text{-}Response\ m]$ is useful when an agent can not answer a message m and the dialog is not finished yet (see Section II-F). Observe that the performatives in Table I are not FIPA standard [27], since ASBO agents do not necessarily follow it. As a result, we have extended the FIPA performative set to include them. This subject requires a deeper analysis and it is beyond the scope of this paper.

B. The interaction protocol

After defining the messages in $ASBO_{cm}$, it is necessary a protocol for controlling the exchange of such messages between agents. The goal of this protocol is to delimit how to start, follow and end a persuasion dialog in ASBO. These dialogs are held between two agents, namely a *proponent* and an *opponent*. The proponent puts forward its arguments to persuade the opponent to accept an initial assertion. On the other hand, the opponent tries to prove that such an assertion is not valid through its own arguments, if possible (i.e., the opponent has beliefs which are inconsistent with the proponent's assertions).

The interaction protocol is defined in ASBO by using an AUML diagram [28] (see figure 2). The symbol $\diamond$ represents a decision node, in which only one message is selected. Let us denote this protocol with P_{ASBO}. P_{ASBO} starts when the proponent *claims* an assertion φ. The opponent may respond *conceding* that assertion if it agrees to such a claim, or *asking for* an argument supporting φ through the performative *Why*. In the latter case, the proponent *retracts* φ if no arguments can be built for it or all them are defeated. If the proponent has a valid argument S supporting φ, it is sent through the performative *Since*.

When receiving an argument S, the opponent has three options: (1) To *accept* S if the opponent agrees to it; (2) to *ask* for arguments supporting a premise σ_i in the support set of S; or (3) to attack the conclusion or premises in S by giving another argument T. The concept of *attack* to an assertion (i.e., conflict between pieces of knowledge) is represented by the negation symbol $\neg$ in figure 2. When the opponent answers as in the case (2), the proponent must now give an argument for the inquired premise or retract it. In the case (3), the proponent has now the same three options listed above with respect to the argument T (i.e., to accept T, to challenge a premise in T, or to attack T). The rest of the protocol can easily be followed from figure 2. Moreover, the message $[No\text{-}Response\ m]$ could be used as response for the rest of messages in $ASBO_{cm}$, although it has not been included in the figure for simplicity. This message has no responses and it will only be used when an agent can not attack the message body of m neither it can be accepted.

Fig. 2. AUML diagram of the ASBO persuasion protocol.

Observe that the protocol in figure 2 offers several options to answer a message. These options are called the *answer result set* (ARS) of a message. Thus, P_{ASBO} could be expressed as a function which takes a message and returns the correspondent ARS for that message according to the protocol in figure 2. In the rest of the paper, we will denote its use as $P_{ASBO}(m)$, where m is a message. For example, suppose that an agent receives the message $[Claim\ \varphi]$[1]. By applying P_{ASBO} to it, the agent obtains the following answer result set

$$\begin{aligned} P_{ASBO}([Claim\ \varphi]) = \quad & \{[Why\ \varphi], [Concede\ \varphi], \\ & [No-Response(Claim\ \varphi)]\} \end{aligned}$$

Notice that the ARS for a message is not sufficient to generate a suitable response. Therefore, an additional piece of information is needed for that: the context of the dialog.

C. The context and the state of a dialog

The context of a dialog is the set of information that enables an agent to take a decision about which message in the ARS is suitable to be used as an answer. Each agent maintains its own context, which is updated as the dialog progresses (see Section II-D). Hence, the context of a dialog for one agent in a moment t includes the sequence of messages emitted by the agent until that moment, the beliefs that such an agent holds, and the assertions emitted by the other agent. Formally, a context in ASBO is defined as follows:

Definition 1 (Context): Let us consider P, O as two agents which are using P_{ASBO} to exchange messages. Let Δ_P^t be the set of beliefs that P holds in a particular moment t. Let $H_{P,O}^t$ be the set of messages emitted by P to O in t, and let $\Pi_{P,O}^t$ be the set of assertions that O has communicated to P in t. Then, a context for the agent P in a moment t with respect to a dialog with the agent O is defined as the 3-tuple (Δ_P^t, $H_{P,O}^t$, $\Pi_{P,O}^t$), denoted as $C_{P,O}^t$. Analogously, the context for the agent O is $C_{O,P}^t$ = (Δ_O^t, $H_{O,P}^t$, $\Pi_{O,P}^t$).

Notice that $H_{P,O}^t$ contains the history of all messages sent by P to O until t. This history is used together with P_{ASBO} as follows: when the agent P needs to know which utterances may be answered, it uses P_{ASBO} to obtain the corresponding ARS. After this, only those messages compound by a performative and a message body which do not cause cycles can be emitted. Precisely, messages m causing cycles are those which for a particular message $m' \in H_{P,O}^t$, the performative and message body are the same in m and m' (i.e., either assertions φ in m and φ' in m' or arguments S in m and S' in m' are equivalent). Then, the remaining answers in the ARS are evaluated using the agent's beliefs, Δ_P^t, and the assertions the other agent is committed to, $\Pi_{P,O}^t$, so as to decide which answer is the optimal. Details about this evaluation are beyond the scope of this paper, and we suppose it is available in the form of the function 1 given below.

Once the context of a dialog for an agent has been given, it is possible to define a strategy function which, starting from the ARS of a message, it generates a single response:

Function 1 (Strategy function $\mathcal{P}$): We define the strategy function $\mathcal{P}$ in ASBO as

$$\mathcal{P} : ARS \times C \longrightarrow ASBO_{cm},$$

where ARS is the set of all possible answer result sets and C is the set of all possible contexts. Then, for two particular agents P, O involved in an exchange of messages in an instant of time t, $\mathcal{P}$ is such that $\forall ars \in ARS$, $\mathcal{P}(ars, C_{P,O}^t) \notin H_{P,O}^t$. Analogously, $\mathcal{P}(ars, C_{O,P}^t) \notin H_{O,P}^t$.

Apart from the context, agents also need to maintain the *state* of the dialog. This state contains the set of all messages exchanged between agents until the moment t, which agent has the turn of speaking in t, and which message is currently being processed. In ASBO, it is defined as follows:

Definition 2 (State): Let P, O be two agents using P_{ASBO} to exchange messages in a moment t. Let $\mathcal{T} \in \{P, O\}$ be the agent responsible for sending a message in $t+1$, and let m_t be the current message being processed in t by $\mathcal{T}$. Then, the state of that exchange of message in t is the 3-tuple $(\mathcal{T}, m_t, H_{P,O}^t \cup H_{O,P}^t)$, denoted as $\mathcal{S}^t$.

Observe that $\mathcal{S}^t$ is the same for both agents P and O. The state of a dialog is updated by means of the effects associated to the $ASBO_{cm}$ messages. Such effects are defined in the form of *effect rules*, explained in the next section.

D. Effect rules

$ASBO_{cm}$ messages are communicative acts. Therefore, the agents' contexts and the state $\mathcal{S}$ of a dialog must be updated according to effects of these acts. The update of $\mathcal{S}$ may be performed through two different directions. Suppose $\mathcal{S}^t = (P, m_t, H_{P,O}^t \cup H_{O,P}^t)$. Then, some messages will update $\mathcal{S}$ in $t+1$ as

$$\mathcal{S}^{t+1} = (O, m_{t+1}, H_{P,O}^{t+1} \cup H_{O,P}^{t+1}),$$

i.e. the agent P emitting a message in $\mathcal{S}^t$ is now the receiver of the next one in $\mathcal{S}^{t+1}$. Some other messages, namely *surrendered* (those using the performatives $Concede$, $Retract$, $Accept$ and $No\text{-}Response$), update $\mathcal{S}$ in $t+1$ as

$$\mathcal{S}^{t+1} = (P, m_{t'}, H_{P,O}^{t+1} \cup H_{O,P}^{t+1}),$$

where $t' < t$ and $m_{t'} \in H_{O,P}^t$, i.e. P must again utter a new message in $\mathcal{S}^{t+1}$ as response to a previous message sent by O. This situation occurs due to the emitter of a surrendered message has reached the end of an argumentation line in $\mathcal{S}^t$, and therefore it is still the responsible for continuing the exchange of messages by exploring another argumentation line (if possible). In order to define these exploration mechanisms, ASBO agents need a *backward* function:

Function 2 (Backward function $\mathcal{B}$): We define a backward function in ASBO as

$$\mathcal{B} : ASBO_{cm} \times S \longrightarrow ASBO_{cm},$$

[1] And the dialog is in the appropriate state for considering such message as correct.

where S is the set of all possible states. Then,

$$\mathcal{B}(m_{t+n}, \mathcal{S}^{t+n}) = m_t$$

for a particular message m_{t+n} and the state of the exchange of messages $\mathcal{S}^{t+n} = (E, m_{t+n}, H^{t+n}_{E,R} \cup H^{t+n}_{R,E})$ between two agents E and R in $t+n$, $n > 1$, if the following conditions hold:

1) Let m_t, m_{t+1} and m_{t+n} be messages exchanged among E and R until the instant $t+n$, where $m_t, m_{t+n} \in H^{t+n}_{R,E}$ and $m_{t+1} \in H^{t+n}_{E,R}$, and
2) $m_{t+1} = \mathcal{P}(P_{ASBO}(m_t), C^t_{E,R})$, and
3) $m_{t+n} = \mathcal{P}(P_{ASBO}(m_{t+1}), C^{t+n-1}_{R,E})$.

Then, if E emits a surrendered message in $\mathcal{S}^{t+n}$ as an answer to m_{t+n}, it still has the turn of speaking. Now, to discover which previous message emitted by R must be responded, E applies $\mathcal{B}(m_{t+n}, \mathcal{S}^{t+n})$, and then

$$\mathcal{S}^{t+n+1} = (E, \mathcal{B}(m_{t+n}, \mathcal{S}^{t+n}), H^{t+n+1}_{E,R} \cup H^{t+n+1}_{R,E}).$$

A case of use of $\mathcal{B}$ is given in Section III. Let us see now the effect rules of $ASBO_{cm}$ messages.

Definition 3 (Effect rules): Let E, R be the respective emitter and receiver agents of an $ASBO_{cm}$ message m_{t+1}. Let $\mathcal{S}^t = (E, m_t, H^t_{E,R} \cup H^t_{R,E})$ be the state of the exchange of messages between these agents in the instant t. Let $C^t_{E,R}$ and $C^t_{R,E}$ be the contexts of agents E and R in t, respectively, such as $m_{t+1} = \mathcal{P}(P_{ASBO}(m_t), C^t_{E,R})$. Then, the set of effect rules for m_{t+1} is defined as follows:

ER1. If m_{t+1} = $[Claim\ \varphi]$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $(\Delta^t_R$, $H^t_{R,E}$, $\Pi^t_{R,E}$ $\cup\{\varphi\})$, and
- $\mathcal{S}^{t+1} = (R, m_{t+1}, H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

ER2. If m_{t+1} = $[Why\ \varphi]$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $C^t_{R,E}$, and
- $\mathcal{S}^{t+1} = (R, m_{t+1}, H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

ER3. If m_{t+1} = $[\varphi\ Since\ S], S = \{\varphi, \Phi\}$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $(\Delta^t_R, H^t_{R,E}, \Pi^t_{R,E} \cup \Phi)$, and
- $\mathcal{S}^{t+1}$ = $(R, m_{t+1},$ $H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

ER4. If m_{t+1} = $[Concede\ \varphi]$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $(\Delta^t_R, H^t_{R,E}$, $\Pi^t_{R,E} \cup \{\varphi\})$, and
- $\mathcal{S}^{t+1} = (E, \mathcal{B}(m_t, \mathcal{S}^t), H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

ER5. If m_{t+1} = $[Retract\ \varphi]$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E - \{\varphi\}, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $(\Delta^t_R, H^t_{R,E}, \Pi^t_{R,E} - \{\varphi\})$, and
- $\mathcal{S}^{t+1}$ = $(E, \mathcal{B}(m_t, \mathcal{S}^t), H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

ER6. If m_{t+1} = $[Accept\ S], S = \{\varphi, \Phi\}$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E \cup \{\varphi\}, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $(\Delta^t_R, H^t_{R,E}, \Pi^t_{R,E} \cup \{\varphi\})$, and
- $\mathcal{S}^{t+1}$ = $(E, \mathcal{B}(m_t, \mathcal{S}^t), H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

ER7. If m_{t+1} = $[No\text{-}Response\ m_t]$, then

- $C^{t+1}_{E,R}$ = $(\Delta^t_E, H^t_{E,R} \cup \{m_{t+1}\}, \Pi^t_{E,R})$,
- $C^{t+1}_{R,E}$ = $C^t_{R,E}$, and
- $\mathcal{S}^{t+1} = (E, \mathcal{B}(m_t, \mathcal{S}^t),$ $H^t_{E,R} \cup H^t_{R,E} \cup \{m_{t+1}\})$.

Observe that when E utters m_{t+1} as a surrendered message ($Concede$, $Retract$, $Accept$ and $No\text{-}Response$), it still keeps the turn of speaking in $\mathcal{S}^{t+1}$, and the current message to be processed in that instant is $\mathcal{B}(m_t, \mathcal{S}^t)$. In other words, when E accepts an R's premise or argument, or it withdraws from an assertion previously claimed by itself, then E must give a new answer to a previous message sent by R (if possible).

E. ASBO persuasion dialog

After defining the previous elements (i.e., communication language, interaction protocol, agents' contexts, state of an exchange of messages and effect rules), persuasion dialogs can now be built as a sequence of messages according to such definitions. Thus, the formal definition of an ASBO persuasion dialog follows.

Definition 4 (ASBO persuasion dialog): Let agents P, O be the proponent and opponent of an assertion φ, respectively. Then, a sequence $d = [m_1, m_2, \ldots, m_n]$ is an ASBO persuasion dialog about φ if and only if:

1) m_1 = $[Claim\ \varphi]$, such as $m_1 \in H^n_{P,O}$, and
2) $\forall m_i \in d, i > 1$,
$$m_i = \mathcal{P}(P_{ASBO}(m_{i'}), C^{i-1}_{\mathcal{T},X}),$$
such as $1 \leq i' < i$, and
$$\mathcal{S}^{i-1} = (\mathcal{T}, m_{i'}, H^{i-1}_{P,O} \cup H^{i-1}_{O,P}),$$
where $X \equiv O$ if $\mathcal{T} \equiv P$ and $X \equiv P$ if $\mathcal{T} \equiv O$, and
$$m_{i'} = \mathcal{B}(m_{i-2}, \mathcal{S}^{i-2})$$
iff m_{i-1} is a surrendered message, or $m_{i'}$ = m_{i-1} otherwise, and
3) $\forall m_i, m_{i'} \in d$, if $i \neq i'$, $m_i \not\equiv m_{i'}$.

The first condition states that the dialog is always commenced by P claiming the assertion φ. The second condition defines each message m_i in the dialog (except the first one) as a result of applying $\mathcal{P}$ to a previous message $m_{i'}$ with respect to the context $C^{i-1}_{\mathcal{T},X}$ of the agent $\mathcal{T}$ holding the turn of speaking. Finally, the last condition avoids cycles in the dialog since the repetition of message is not allowed. Although the function $\mathcal{P}$ forbids an agent to repeat an utterance, this last condition is necessary to prevent an agent from emitting the same message previously sent by the other agent.

F. Termination and outcome conditions

There are three circumstances in the ASBO persuasion framework that produce the end of a dialog discussing the assertion φ. In the first one, the opponent emits $[Concede\ \varphi]$ if it agrees to that claim or after accepting an argument supporting φ, since it can not find any valid opposition to such an argument. In the second one, the proponent utters $[Retract\ \varphi]$ when all the arguments supporting φ have been invalidated (i.e., defeated). Finally, the third condition occurs when the proponent emits $[No\text{-}Response\ [Why\ \varphi]]$ or the opponent utters $[No\text{-}Response\ [Claim\ \varphi]]$. This last condition means that one of the agents withdraws from the dialog without accepting or rejecting the initial claim φ. This situation may occur when some arguments for and against φ are incomparable, and neither are they accepted nor defeated.

Moreover, following these termination conditions, the outcome of the dialog can also be defined. If the first condition holds, the proponent wins and the opponent accepts φ. If the second condition holds, then the opponent wins and the proponent must reject φ. In the third case, none of the agents wins and φ is undecided with respect to them. The termination and outcome conditions are formally expressed next.

Definition 5 (Termination and outcome conditions): Let agents P, O be the proponent and opponent of an assertion φ, respectively. Let d be an ASBO persuasion dialog between P and O about φ, where $m_1 \in d$ is such that $m_1 = [Claim\ \varphi]$. Moreover, let S^t be the state of d in the instant t. Then, the dialog d terminates when:

1) $\mathcal{S}^t$ = $(O, m_t, H^t_{P,O} \cup H^t_{O,P})$, and m_{t+1} = $[Concede\ \varphi]$, or
2) $\mathcal{S}^t$ = $(P, m_t, H^t_{P,O} \cup H^t_{O,P})$, and m_{t+1} = $[Retract\ \varphi]$, or
3) $\mathcal{S}^t$ = $(\mathcal{T}, m_t, H^t_{P,O} \cup H^t_{O,P})$, and
   - $\mathcal{T} \equiv P$ and m_{t+1} = $[No\text{-}Response\ [Why\ \varphi]]$, or
   - $\mathcal{T} \equiv O$ and m_{t+1} = $[No\text{-}Response\ [Claim\ \varphi]]$.

In the first case, P wins the dialog and φ is accepted. In the second case, O is the winner and φ is not accepted. Otherwise, φ is undecided.

III. An example of persuasion dialog in ASBO

In order to illustrate all the ideas explained so far, an example of a complete ASBO persuasion dialog is given next. Let us consider an agent A_{Ex} supporting an assertion φ by means of two different arguments, S=(φ, Φ_S) and R=(φ, Φ_R). Now, let B_{Ex} be an agent with an argument T = (ψ, Φ_T), where ψ defeats a premise in Φ_S (i.e., T undercuts S). Moreover, B_{Ex} does not have any opinion related to φ. A_{Ex} tries to convince B_{Ex} about the validness of φ. Therefore, both agents start an ASBO persuasion dialog, being A_{Ex} the proponent by claiming φ, and B_{Ex} the opponent by requiring a valid support to that claim while trying to invalidate it. At the initial moment, the local knowledge bases of each agents are

$$\Delta^0_{A_{Ex}} = \{\{\varphi\} \cup \Phi_S \cup \Phi_R\},$$
$$\Delta^0_{B_{Ex}} = \{\{\psi\} \cup \Phi_T\}.$$

Then, the dialog is developed as shown next:

- At t=0,
  - $\mathcal{S}^0$=$(A_{Ex}, -, \emptyset)$,
  - $C^0_{A_{Ex}}$=$(\Delta^0_{A_{Ex}}, \emptyset, \emptyset)$, and
  - $C^0_{B_{Ex}}$=$(\Delta^0_{B_{Ex}}, \emptyset, \emptyset)$.

Agent A_{Ex} seeks to persuade B_{Ex} to accept φ. Then, m_1=$[Claim\ \varphi]$.

- At t=1, m_1=$[Claim\ \varphi]$, and
  - $\mathcal{S}^1$=$(B_{Ex}, m_1, \{m_1\})$,
  - $C^1_{A_{Ex}}$=$(\Delta^0_{A_{Ex}}, \{m_1\}, \emptyset)$, and
  - $C^1_{B_{Ex}}$=$(\Delta^0_{B_{Ex}}, \emptyset, \{\varphi\})$.

Agent B_{Ex} does not have any initial information to agree to φ, so it asks for an argument supporting it. Then, $\mathcal{P}(P_{ASBO}(m_1), C^1_{B_{Ex}})$ = m_2 = $[Why\ \varphi]$.

- At t=2, m_2 = $[Why\ \varphi]$, and
  - $\mathcal{S}^2$=$(A_{Ex}, m_2, \{m_1, m_2\})$,
  - $C^2_{A_{Ex}}$=$C^1_{A_{Ex}}$, and
  - $C^2_{B_{Ex}}$=$(\Delta^1_{B_{Ex}}, \{m_2\}, \{\varphi\})$.

Agent A_{Ex} must communicate an argument justifying φ. As stated above, it owns arguments S and R for this. For the purpose of this example, suppose that argument S is the most appropriate alternative according to A_{Ex}'s strategy function. Then, $\mathcal{P}(P_{ASBO}(m_2), C^2_{A_{Ex}})$ = m_3 = $[\varphi\ Since\ S]$.

- At t=3, m_3 = $[\varphi\ Since\ S]$, and
  - $\mathcal{S}^3$ = $(B_{Ex}, m_3, \{m_1, m_2, m_3\})$,
  - $C^3_{A_{Ex}}$ = $(\Delta^2_{A_{Ex}}, \{m_1, m_3\}, \emptyset)$, and
  - $C^3_{B_{Ex}}$ = $(\Delta^2_{B_{Ex}}, \{m_2\}, \{\varphi \cup \Phi_S\})$.

Agent B_{Ex} has the belief ψ which is in conflict with a premise in argument S. Moreover, ψ is supported by the argument T. Then, $\mathcal{P}(P_{ASBO}(m_3), C^3_{B_{Ex}})$ = m_4 = $[\psi\ Since\ T]$. Note that argument T defeats argument S since ψ undercuts a premise in Φ_S.

- At t=4, m_4 = $[\psi\ Since\ T]$, and
  - $\mathcal{S}^4$ = $(A_{Ex}, m_4, \{m_1, m_2, m_3, m_4\})$,
  - $C^4_{A_{Ex}}$ = $(\Delta^3_{A_{Ex}}, \{m_1, m_3\}, \{\psi \cup \Phi_T\})$, and
  - $C^4_{B_{Ex}}$ = $(\Delta^3_{B_{Ex}}, \{m_2, m_4\}, \{\varphi \cup \Phi_S\})$.

Fig. 3. Tree generated from the persuasion dialog between A_{Ex} and B_{Ex}.

Agent A_{Ex} does not have any attacks against argument T. Then, $\mathcal{P}(P_{ASBO}(m_4), C^4_{A_{Ex}})$ = m_5 = [*Accept* T]. Notice that m_5 is a surrendered message answering to m_4. In this manner, the argumentation line supporting φ through S is ended since such an argument has been defeated. Now, by applying $\mathcal{B}(m_4, \mathcal{S}^4)$, the message to process in $\mathcal{S}^5$ by A_{Ex} is m_2 again. This result is due to a new argumentation line can be opened from answering m_2 = $[Why\ \varphi]$ with another argument supporting φ.

- At t=5, m_5 = [*Accept* T], and
  - $\mathcal{S}^5$ = $(A_{Ex}, m_2, \{m_1, m_2, m_3, m_4, m_5\})$,
  - $C^5_{A_{Ex}}$ = $(\Delta^4_{A_{Ex}} \cup \{\psi\}, \{m_1, m_3, m_5\}, \{\psi \cup \Phi_T\})$, and
  - $C^5_{B_{Ex}}$ = $(\Delta^4_{B_{Ex}}, \{m_2, m_4\}, \{\varphi \cup \Phi_S \cup \psi\})$.

Agent A_{Ex} still has the argument R supporting φ. This argument has not previously been uttered neither defeated. Then, $\mathcal{P}(P_{ASBO}(m_2), C^5_{A_{Ex}})$ = m_6 = $[\varphi\ Since\ R]$. Hence, it gives a new argument supporting the assertion φ, by answering to the message $[Why\ \varphi]$ again.

- At t=6, m_6 = $[\varphi\ Since\ R]$, and
  - $\mathcal{S}^6$ = $(B_{Ex}, m_6, \{m_1, m_2, m_3, m_4, m_5, m_6\})$,
  - $C^6_{A_{Ex}}$ = $(\Delta^5_{A_{Ex}}, \{m_1, m_3, m_5, m_6\}, \{\psi \cup \Phi_T\})$, and
  - $C^6_{B_{Ex}}$ = $(\Delta^5_{B_{Ex}}, \{m_2, m_4\}, \{\varphi \cup \Phi_S \cup \psi \cup \Phi_R\})$.

Agent B_{Ex} does not have any attacks against argument R. Then, $\mathcal{P}(P_{ASBO}(m_6), C^6_{B_{Ex}})$ = m_7 = [*Accept* R]. Observe that B_{Ex} emits a surrendered message in m_7 as a response to message m_6. In turn, m_6 is an answer to m_2. Finally, m_2 is answering to m_1. As a result, $\mathcal{B}(m_6, \mathcal{S}^6)$ = m_1 is the message to be processed by B_{Ex} in $\mathcal{S}^7$.

- At t=7, m_7 = [*Accept* R], and
  - $\mathcal{S}^7$ = $(B_{Ex}, m_1, \{m_1, m_2, m_3, m_4, m_5, m_6, m_7\})$,
  - $C^7_{A_{Ex}}$ = $(\Delta^6_{A_{Ex}}, \{m_1, m_3, m_5, m_6\}, \{\psi \cup \Phi_T\ \cup \varphi)$, and
  - $C^7_{B_{Ex}}$ = $(\Delta^6_{B_{Ex}} \cup \varphi, \{m_2, m_4, m_7\}, \{\varphi, \Phi_S, \psi, \Phi_R\})$.

Agent B_{Ex} has previously accepted the argument R supporting φ. As a result, it now agrees to such an assertion. Then, $\mathcal{P}(P_{ASBO}(m_1), C^7_{B_{Ex}})$ = m_8 = $[Concede\ \varphi]$.

Eventually, the dialog has finished since B_{Ex} has conceded the initial proposal φ. The dialog sequence is $[\lfloor Claim\ \varphi \rfloor, \lceil Why\ \varphi \rceil, \lfloor \varphi\ Since\ S \rfloor, \lceil \psi\ Since\ T \rceil, \lfloor Accept\ T \rfloor, \lfloor \varphi\ Since\ R \rfloor, \lceil Accept\ R \rceil, \lceil Concede\ \varphi \rceil]$, where $\lfloor p \rfloor$ is a message uttered by A_{Ex} and $\lceil p \rceil$ is a message uttered by B_{Ex}. Consequently, A_{Ex} has won the dialog, and φ is finally accepted by B_{Ex}. Figure 3 shows the dialog using a tree-based representation.

IV. Conclusions

Argumentation has demonstrated to be a useful approach when resolving conflicts during the exchange of proposals in a MAS. This is possible due to agents can use their rationality to build and attack arguments that justify those proposals. However, most argumentative approaches in MAS are theoretical. We have developed an Argumentation System Based on Ontologies, ASBO, with an engineering oriented approach. In this manner, ASBO is offered as a software architecture ready to be used by agents performing argumentation in conventional MAS. This paper has presented the formal background for the persuasion dialog framework included in such architecture. This formal model can then be used to identify the necessary elements to be defined for implementing persuasion dialogs within argumentation processes.

One future step in ASBO is to adopt a BDI agency paradigm. In this manner, an agent could select the optimal argument during the persuasion dialog according to its intentions and desires.

Acknowledgments

This work has been supported by the Fundación Séneca grant "Programa de Ayuda a los grupos de excelencia 04552/GERM/06 " and thanks to the Spanish Ministerio de Ciencia e Innovación (MICINN) under the grant AP2006-4154 in frames of the FPU Program, the TIN2005-08501-C03-01 Program and the CICYT TIN2008-06441-C02-02 project.

References

[1] M. Luck and J. J. Gómez-Sanz, Eds., *Agent-Oriented Software Engineering IX, 9th International Workshop, AOSE 2008, Estoril, Portugal, May 12-13, 2008, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 5386. Springer, 2009.

[2] S. Ossowski, J.-L. P. de-la Cruz, J. Z. Hernández, J.-M. Maseda, A. Fernández, M.-V. Belmonte, A. García-Serrano, J. M. Serrano, R. León, and F. Carbone, "Towards a generic multiagent model for decision support: Two case studies," in *CAEPIA*, ser. Lecture Notes in Computer Science, R. Conejo, M. Urretavizcaya, and J.-L. P. de-la Cruz, Eds., vol. 3040. Springer, 2003, pp. 597–607.

[3] N. dos Santos, F. M. Varejáo, and O. de Lira Tavares, "Multi-agent systems and network management - A positive experience on Unix environments," in *IBERAMIA 2002: Proceedings of the 8th Ibero-American Conference on AI*. London, UK: Springer-Verlag, 2002, pp. 616–624.

[4] D. Sharma and F. Shadabi, "An intelligent multi-agent design in healthcare management system," in *KES-AMSTA*, ser. Lecture Notes in Computer Science, N. T. Nguyen, G. Jo, R. J. Howlett, and L. C. Jain, Eds., vol. 4953. Springer, 2008, pp. 674–682.

[5] R. Choren, A. F. Garcia, H. Giese, H. fung Leung, C. J. P. de Lucena, and A. B. Romanovsky, Eds., *Software Engineering for Multi-Agent Systems V, Research Issues and Practical Applications*, ser. Lecture Notes in Computer Science, vol. 4408. Springer, 2007.

[6] A. Muñoz, J. A. Botía, F. J. García, G. Martínez, and A. F. G. Skarmeta, "Solving conflicts in agent-based ubiquitous computing systems: A proposal based on argumentation," in *Agent-Based Ubiquitous Computing*, ser. Atlantis Ambient And Pervasive Intelligence, E. Mangina, J. Carbó, and J. M. Molina, Eds., vol. 1, no. 1. Atlantis Press, May 2009, pp. 1–12.

[7] C. Tessier, L. Chaudron, and H.-J. Müller, Eds., *Conflicting agents: conflict management in multi-agent systems*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[8] N. T. Nguyen, *Advanced Methods for Inconsistent Knowledge Management (Advanced Information and Knowledge Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[9] J. Fox, P. Krause, and S. Ambler, "Arguments, contradictions and practical reasoning," in *ECAI '92: Proceedings of the 10th European conference on Artificial intelligence*. New York, NY, USA: John Wiley & Sons, Inc., 1992, pp. 623–627.

[10] M. N. Huhns and D. M. Bridgeland, "Multiagent truth maintenance," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 6, pp. 1437–1445, 1991.

[11] G. K. Kraetzschmar, *Distributed Reason Maintenance for Multiagent Systems*, J. Siekmann and J. G. Carbonell, Eds. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[12] S. Ram and J. Park, "Semantic conflict resolution ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts," *Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 189–202, 2004.

[13] K. Lee and M. Kim, "A novel approach for conflict resolution in context-awareness using semantic unification of multi-cognition," in *PRIMA '08: Proceedings of the 11th Pacific Rim International Conference on Multi-Agents*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 267–274.

[14] A. Muñoz and J. A. Botía, "ASBO: Argumentation system based on ontologies," in *Cooperative Information Agents XII*, ser. Lecture Notes in Artificial Intelligence, M. Klusch, M. Pechoucek, and A. Polleres, Eds., vol. 5180. Springer, 2008, pp. 191–205.

[15] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[16] M. Dean, D. Connoll, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "Web Ontology Language (OWL)," W3C, Tech. Rep., 2004. [Online]. Available: http://www.w3.org/TR/owl-ref/

[17] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007.

[18] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: implementing the Semantic Web recommendations," in *Proceedings of the 13th international World Wide Web conference*. ACM Press, 2004, pp. 74–83.

[19] G. Brewka, "Dynamic argument systems: A formal model of argumentation processes based on situation calculus," *JLC: Journal of Logic and Computation*, vol. 11, 2001.

[20] H. Prakken, "Coherence and flexibility in dialogue games for argumentation," *Journal of Logic and Computation*, vol. 15, no. 6, pp. 1009–1040, 2005.

[21] D. V. Carbogim, D. Robertson, and J. Lee, "Argument-based applications to knowledge engineering," *Knowledge Engineering Review*, vol. 15, no. 2, pp. 119–149, 2000.

[22] N. Maudet, S. Parsons, and I. Rahwan, Eds., *Argumentation in Multi-Agent Systems, Third International Workshop, ArgMAS 2006, Hakodate, Japan, May 8, 2006, Revised Selected and Invited Papers*, ser. Lecture Notes in Computer Science, vol. 4766. Springer, 2007.

[23] I. Rahwan and B. Banihashemi, "Arguments in OWL: A Progress Report," in *Computational Models of Argument (COMMA)*, ser. Frontiers in Artificial Intelligence and Applications, P. Besnard, S. Doutre, and A. Hunter, Eds., vol. 172. IOS Press, 2008, pp. 297–310.

[24] S. Parsons, C. Sierra, and N. R. Jennings, "Agents that reason and negotiate by arguing," *Journal of Logic and Computation*, vol. 1998, no. 3, pp. 261–292, 1998.

[25] N. Karunatillake and N. R.Jennings, "Is it worth arguing?" in *Proc. of Argumentation in Multi-Agent Systems (ArgMAS)*, Springer-Verlag, Ed., vol. LNAI 3366, New York, USA, 2004, pp. 134–250.

[26] A. Muñoz, A. Sánchez, and J. A. Botía, "A software architecture for an argumentation-oriented multi-agent system," in *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'09)*, ser. Advances in Intelligent and Soft Computing, Y. Demazeau, J. Pavn, J. Corchado, and J. Bajo, Eds., vol. 55. Springer, 2009, pp. 197–207.

[27] FIPA, "Fipa communicative act library specification," Tech. Rep., December 2002.

[28] B. Bauer, J. P. Müller, and J. Odell, "Agent UML: A formalism for specifying multiagent software systems," in *AOSE*, ser. Lecture Notes in Computer Science, P. Ciancarini and M. Wooldridge, Eds., vol. 1957. Springer, 2000, pp. 91–104.