Specifying and Implementing a Persuasion Dialogue Game Using Commitments and Arguments

Jamal Bentahar[1], Bernard Moulin[1,2], and Brahim Chaib-draa[1]

[1] Laval University, Computer Science and Software Engineering Department,
Ste Foy, QC, G1K 7P4, Canada
jamal.bentahar.1@ulavl.ca
{bernard.moulin, brahim.chaib-draa}@ift.ulaval.ca
[2] Geomatica Research Centre,
Ste Foy, QC, G1K 7P4, Canada

Abstract. In this paper we propose a new persuasion dialogue game for agent communication. We show how this dialogue game is modeled by a framework based on social commitments and arguments. Called Commitment and Argument Network (CAN), this framework allows us to model communication dynamics in terms of actions that agents apply to commitments and in terms of argumentation relations. This dialogue game is specified by indicating its entry conditions, its dynamics and its exit conditions. In order to solve the problem of the acceptance of arguments, the protocol integrates the concept of agents' trustworthiness in its specification. The paper proposes a set of algorithms for the implementation of the persuasion protocol and discusses their termination, complexity and correctness.

1 Introduction

Research in agent communication languages and protocols has received much attention during the last years. Protocols describe the allowed communicative acts that agents can perform when conversing. These protocols specify the rules governing a dialogue between agents in multi-agent systems (MAS).

Traditionally, protocols are specified as finite state machines or Petri nets without taking into account the agents' autonomy. Therefore, these protocols are not flexible enough to be used in open MAS [13]. To solve this problem, several researchers proposed protocols using dialogue games (DGs) [9, 10, 13, 14]. DGs are interactions between players, in which each player moves by performing utterances according to a pre-defined set of roles [14].

The protocols described in the literature are often specified by pre/post conditions. These protocols often neglected the decision-making process that allows agents to accept or to refuse an utterance. The protocols based on formal dialectics [2, 15, 23] use the argumentation as a way of expressing decision-making. However, the sole argumentation does not make it possible to solve a decision-making problem. We think that other social elements such as agents' trustworthiness must be taken into account.

I. Rahwan et al. (Eds.): ArgMAS 2004, LNAI 3366, pp. 130–148, 2005.

The contribution of this paper is the proposition of a formal specification and an implementation of a new persuasion dialogue game for agent communication using a unified framework based on social commitments and on arguments. Our protocol is presented in the context of this framework called Commitment and Argument Network (CAN) [5, 6]. This protocol is characterized by the fact that it integrates the agents' trustworthiness as a component of the decision-making process.

The rest of this paper is organized as follows. In Section 2 we present our approach based on commitments and arguments. In Section 3 we introduce the CAN formalism. In Section 4 we present the specification of our dialogue game and we highlight the importance of agents' trustworthiness. In Section 5 we present our model of trustworthiness. In Section 6 we describe some issues of the implementation. In Section 7 we discuss some characteristics of our algorithms. In Sections 8 and 9 we compare our protocol to related work and we conclude the paper.

2 Approach Based on Commitments and Arguments

2.1 Social Commitment

In the domain of agent communication, it is largely recognized that social commitments are a powerful representation for modeling multi-agent interactions [4, 5, 8, 12, 13, 25]. In opposition to the BDI (beliefs, desires and intentions) approach, the commitment-based approach stresses the importance of conventions and the public and social aspects of dialogue. It is based on social commitments that are thought of as social and deontic notions. As a social notion, commitments are a base for a normative framework that makes it possible to model the agents' behavior. Indeed, considering their deontic nature, these commitments define constraints on this behavior. The agent must behave in accordance to its commitments. For example, by committing towards other agents that a certain fact is true, the agent is compelled not to contradict itself during the conversation. It must also be able to explain, argue, justify and defend itself if another participant contradicts it. A speaker is committed to a statement when he made this statement or when he agreed with this statement made by another participant. In fact, we do not speak here about the expression of a belief, but rather about a particular relationship between a participant and a statement.

A Social commitment *SC* is a commitment made by an agent (called the *debtor*), that some fact is true. This commitment is directed to a set of agents (called *creditors*) [8]. In order to model the dynamics of conversations, we interpret a speech act *SA* as an *action* performed on a commitment or on its content (we refer to this as "take position on a commitment"). A speech act is an abstract act that an agent, the speaker, performs when producing an utterance *U* and addressing it to another agent, the addressee [24]. The actions that an agent can perform on a commitment are: $Act \in \{Create, Withdraw\}$. The actions performed on the content of a commitment are: $Act\text{-}content \in \{Accept, Refuse, Challenge\}$. Thus, a speech act is defined as an action on a commitment when the speaker is the debtor, or as an action on a commitment content when the speaker is the debtor or the creditor. Formally:

Definition 1. $SA(Ag_1, Ag_2, U) =_{def}$
$Act(Ag_1, SC(Ag_1, Ag_2, p))$
$| \; Act\text{-}content(Ag_k, SC(Ag_i, Ag_j, p))$

where $i, j \in \{1, 2\}$ and $(k=i \text{ or } k=j)$, $=_{def}$ means "is interpreted by definition as", p the commitment content. This definition allows us to model agent interaction using actions that agents perform on commitments and on their contents.

2.2 Argumentation and Social Commitments

An argumentation system essentially includes a logical language L, a definition of the argument concept, a definition of the attack relation between arguments and finally a definition of acceptability. Several definitions were also proposed for the argument concept [19, 28]. In our model, we adopt the following definitions from [11]. Here Γ indicates a possibly inconsistent knowledge base with no deductive closure. $\vdash$ Stands for classical inference and $\equiv$ for logical equivalence.

Definition 2. *An argument is a pair (H, h) where h is a formula of L and H a sub-set of Γ such that : i) H is consistent, ii) $H \vdash h$ and iii) H is minimal, so no subset of H satisfying both i and ii exists. H is called the support of the argument and h its conclusion.*

Definition 3. *Let (H_1, h_1), (H_2, h_2) be two arguments. (H_1, h_1) attack (H_2, h_2) iff* $h_1 \equiv \neg h_2$.

The *defense* relation is defined as a dual relation of *attack*.

Argumentation is based on the construction of arguments and counter-arguments, the comparison of these various arguments and finally the selection of the arguments that are considered to be acceptable. In our approach, agents must reason on their own mental states in order to build arguments in favor of their future commitments, as well as on other agents' commitment in order to be able to take position with respect to the contents of these commitments.

In fact, before committing to some fact h being true (i.e. before creating a commitment whose content is h), the speaker agent must use its argumentation system to build an argument (H, h). On the other side, the addressee agent must use its own argumentation system to select the answer it will give (i.e. to decide about the appropriate manipulation of the content of an existing commitment). For example, an agent Ag_1 accepts the commitment content h proposed by another agent Ag_2 if Ag_1 has an argument for h. If Ag_1 has an argument neither for h, nor for $\neg h$, then it must ask for an explanation. Thus, we claim that an agent's argument must support an action performed by this agent on a given commitment or on its content. The semantics of our commitment and argument approach is described in [6]. Surely, an argumentation system is essential to help agents to act on commitments and on their contents. However, reasoning on other social attitudes should be taken into account in order to explain the agents' decisions. In our persuasion protocol we highlight the importance of agents' trustworthiness to decide, in some cases, about the acceptance of arguments.

3 The CAN Formalism

So far, we presented our framework of commitments and arguments. Thus, agents can participate in conversations by manipulating commitments and by producing arguments. In this section, we show how a conversation can be modeled using the CAN formalism on the basis of this framework. In this paper we use a simplified version of the CAN which is sufficient to specify our persuasion DG. The complete version is described in [5]. A CAN is a mathematical structure which we define formally as follows:

Definition 4: *A CAN is a 7-uple:* $<A, E, SC(Ag_1, Ag_2, p), \Omega, \Sigma, \Delta, \alpha>$ *where:*

- A: *a finite set of agents.* $A=\{Ag_1, \ldots, Ag_n\}$.
- E: *a finite set of commitments.* $E=\{SC(Ag_1, Ag_2, p), SC(Ag_2, Ag_1, q),\ldots\}$.
- $SC(Ag_1, Ag_2, p)$: *a distinguished element of E: the initial commitment.*
- Ω: *a finite set of creation and positioning actions.* $\Omega=\{Create, Accept, Refuse, Challenge, Withdraw\}$.
- Σ: *a finite set of argumentation relations.*
  $\Sigma=\{Defend, Attack, Justify\}$.
- Δ: *a partial function relating a commitment to another commitment using one argumentation relation.*
  $\Delta: E\times E\rightarrow\Sigma$
- α: *a partial function relating an agent to a commitment using a creation and a positioning action.*
  $\alpha: A\times E\rightarrow\Omega$

The function Δ allows us to define the argumentation relation that can exist between two commitments, i.e. a defense, an attack or a justification relation. For example:

$\Delta(SC(Ag_1, Ag_2, p), SC(Ag_1, Ag_2, q))=Defend$.

This means that the commitment $SC(Ag_1, Ag_2, p)$ (called *source* of the defense relation) defends the commitment $SC(Ag_1, Ag_2, q)$ (called *target* of the defense relation).

The function α allows us to define creation and positioning actions (acceptance, refusal, etc.) performed by an agent on a commitment content. For example:

$\alpha(Ag_1, SC(Ag_2, Ag_1, p))=Accept$

This reflects the acceptance of the content of $SC(Ag_1, Ag_2, p)$. Ag_1 belongs to the debtors set associated with this commitment.

4 Specification of a Persuasion Dialogue Game Based on the CAN Formalism

In this section, we propose a new protocol for persuasion dialogues modeled as *actions* that agents apply to commitments. In this protocol, the persuasion is captured by the argument agents use to support their actions. The semantics of these actions is

defined in a dynamic logic and that of the argumentation relations is defined in an extension of CTL* [6]. Our purpose is to show that the CAN framework can be successfully used to represent a persuasion dialogue game. At a theoretical level, this framework can represent all the elements that constitute the persuasion dynamics. This framework offers a language to represent the dynamics more expressive than the simple pre/post conditions traditionally used as a specification of dialogue games. The differences between our protocol and other protocols proposed in the agent literature are discussed in Section 8.

4.1 General Form

According to the classification of Walton and Krabbe [29], each type of dialogue has an initial situation and the goal of the dialogue is to change this situation in a particular way. Fig. 1 illustrates the initial situation as well as the goal of the persuasion dialogue.

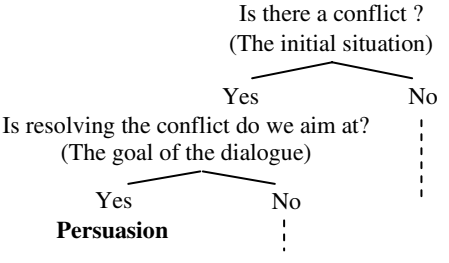

Fig. 1. Goal and initial situation of the persuasion dialogue

In the same context, Vanderveken [27] proposed a *logic of discourse* in which there are only four possible discursive goals speakers can attempt to achieve by conversing. These goals are: descriptive, deliberative, declaratory and expressive goals. Persuasion dialogue is a sub-type of the dialogue types having a descriptive goal. In his typology, Vanderveken argued that each dialogue type with a discursive goal has a mode of achievement of the discursive goal and preparatory conditions. The mode of achievement imposes a certain sequence of speech acts. For a persuasion dialogue, a certain sequence of defense utterances, questions and answers is needed for the successful implementation of such a dialogue. Preparatory conditions determine a structured set of presuppositions related to the discursive goal. The persuasion dialogue has the preparatory conditions that there is a conflict between the agents' points of view and that each agent has the capacity to defend its point of view.

In addition, in the domain of artificial intelligence and law, many computational and logical models of argument and debate, and of reasoning with conflicting information have been proposed [3, 17, 18]. In [18], Prakken and Sartor introduced a dialectical proof theory for an argumentation framework. A proof of a formula takes the form of a dialogue tree, in which each branch of the tree is a dialogue and the root of the tree is an argument for the formula. The idea is that every move in a dialogue consists of an argument based on the input theory, where each stated argument attacks the last move of the opponent in a way that meets the player's burden of proof.

Our persuasion protocol is defined by specifying its entry conditions, its exit conditions and its dynamics. Entry conditions correspond to the initial situation of the dialogue and to the preparatory conditions. Exit conditions correspond to the final situation that makes it possible to determine if the dialogue goal is achieved or not. Dynamics results in the different types of actions that can be performed by agents so that each agent can achieve its goal. The dynamics correspond to the mode of achievement of the discursive goal. It also corresponds to the dialectical proof theory where the root is the persuasion subject. Dynamics is reflected by a set of initiative/reactive DGs. An initiative game is captured by creating a new commitment. A reactive game is captured by taking position on an existing commitment (acceptance, refusal, challenge, defense, etc.).

4.1.1 Entry Conditions

As illustrated by Fig. 1, the entry condition of the persuasion protocol is a conflict of point of view. This is translated in the CAN formalism by the creation of a commitment $SC(p)$ by an agent Ag_1 and the refusal of this commitment by an agent Ag_2. Formally, the initial situation is reflected as follows:

$\alpha(Ag_1, SC(Ag_1, Ag_2, p))=Create$, $\alpha(Ag_2\ SC(Ag_1, Ag_2, p))=Refuse$
$\alpha(Ag_2, SC(Ag_2, Ag_1, \neg p))=Create$.

4.1.2 Dynamics

Generally, the persuasion dialogue takes the form of a sequence of attacks and defenses where each agent tries to defend its point of view or attack the point of view of its partner. This dialogue can also contain questions and answers (dialogue game of information seeking). In the CAN formalism, this results in the creation of a commitment that defends or attacks the initial commitment and other commitments and argumentation relations. The dialogue games of information seeking can be represented by challenge actions and argumentation relations. Formally, the dialogue dynamics can be expressed by a combination of the following functions:

$\alpha(Ag_1, SC(Ag_1, Ag_2, q))=Create$, $\Delta(SC(Ag_1, Ag_2, q), SC(Ag_1, Ag_2, p))=Defend$,
$\alpha(Ag_2, PC(Ag_2, Ag_1, r))=Create$, $\Delta(SC(Ag_2, Ag_1, r), SC(Ag_1, Ag_2, p))=Attack$,
where p, q, r are propositional formulas.

Information seeking can be, for example, represented by:

$\alpha(Ag_2, SC(Ag_1, Ag_2, q))=Challenge$
$\alpha(Ag_1, SC(Ag_1, Ag_2, r))=Create$
$\Delta(SC(Ag_1, Ag_2, r), SC(Ag_1, Ag_2, q))=Justify$.

4.1.3 Exit Conditions

The persuasion dialogue terminates either if the conflict is solved, or with a situation in which each agent does not accept the argument of the other. In this case the protocol terminates with an unsolved conflict. The conflict is solved when one of the two agents adopts the point of view of its partner. In the CAN formalism, this results in the acceptance of the initial commitment $SC(Ag_1, Ag_2, p)$ (respectively $SC(Ag_2, Ag_1, \neg p)$) by Ag_2 (respectively Ag_1). This implies the cancellation of all commitments

attacked $SC(Ag_1, Ag_2, p)$ (respectively $SC(Ag_2, Ag_1, \neg p)$). Formally, if Ag_2 accepts $SC(p)$, the final situation is described as follows:

$\alpha(Ag_2, SC(Ag_1, Ag_2, p))=Accept \Rightarrow$
$[\forall q: \Delta(SC(Ag_2, Ag_1, q), SC(Ag_1, Ag_2, p))=Attack \Rightarrow$
$\alpha(Ag_2, SC(Ag_2, Ag_1, q)):=Withdraw]$

When the two agents mutually refuse the argument of the other, the protocol stops because the conflict cannot be solved.

4.2 Algorithms of the Persuasion Dialogue Game

The general algorithm representing our persuasion dialogue game is given by Algorithm 1. Part *A* of Algorithm 1 specifies the entry conditions. Part *B* indicates the exit conditions. The persuasion dynamics (i.e. the sequence of utterances) is given by the function *Dynamics*. The specification of this function is given by Algorithms 2, 3, 4, 5 and 6. In these algorithms S_{Ag1} indicates the set of arguments of agent Ag_1 (i.e. its knowledge base). S'_{Ag1} indicates the set of arguments that Ag_1 used in the current dialogue. The set S'_{Ag1} allows us to avoid the use of same arguments several times. These algorithms specify the different DGs of our dialogue as *if then* roles. These DGs are: *acceptance*, *refusal*, *challenge*, *justification*, *attack* and *defense*.

```
{ If α(Ag1, SC(Ag1, Ag2, p)) = Create
  And α(Ag2, SC(Ag1, Ag2, p)) = Refuse        } Part A
    Then
      { Conflict := 0 ;
        Dynamics;
        If Conflict = 1 Then
           « The conflict is solved »          } Part B
        Else « The conflict is not solved »
      } }
```

Algorithm 1

Algorithm 2 deals with the acceptance and the refusal cases. The acceptance of $SC(Ag_1, Ag_2, p)$ makes it possible to solve the conflict and to stop the algorithm. In the refusal case, if Ag_1 finds an argument (r, q) not yet used for its commitment $SC(Ag_1, Ag_2, q)$, then this agent creates a new commitment $SC(Ag_1, Ag_2, r)$ to defend $SC(Ag_1, Ag_2, q)$. Ag_1 updates the set S'_{Ag1} by adding the argument (r, q). Ag_1 informs Ag_2 about its action using the *Send* primitive. The *Send* primitive has the form *Send(Destination, Action)*. If Ag_1 does not have arguments to defend its commitment, then the conflict cannot be solved because each agent refuses the arguments of the other and the algorithm stops.

Algorithm 3 deals with the challenge case. Ag_1 justifies its commitment if it finds an argument not yet used. As for the refusal case, Ag_1 updates S'_{Ag1} and informs Ag_2 about its action. If Ag_1 does not find such an argument, then it indicates to Ag_2 that the content of the challenged commitment is a knowledge that Ag_1 believes true by

justifying it by itself. The formal definition of the *justification* relation is the same as the *defense* relation.

```
If α(Ag2, SC(Ag1, Ag2, p)) = Accept Then {
   Conflict := 1; Return Conflict; }
If α(Ag2, SC(Ag1, Ag2, q)) = Refuse Then {
   If (r, q) ∈ SAg1 / S'Ag1 Then {
      α(Ag1, SC(Ag1, Ag2, r)) := Create;
      Δ(SC(Ag1, Ag2, r), SC(Ag1, Ag2, q)):=Defend;
      S'Ag1 := S'Ag1 ∪{(r, q)};
      Send(Ag2, Δ(SC(Ag1, Ag2, r), SC(Ag1, Ag2, q))); }
   Else { Conflict : = -1; Return Conflict; }}
```

Algorithm 2

```
If α(Ag2, PC(Ag1, Ag2, q)) = Challenge Then {
   If (r, q) ∈ SAg1 / S'Ag1 Then {
      α(Ag1, SC(Ag1, Ag2, r)) : = Create;
      Δ(SC(Ag1, Ag2, r), SC(Ag1, Ag2, q)) := Justify;
      S'Ag1 = S'Ag1 ∪{(r, q)};
      Send(Ag2, Δ(SC(Ag1, Ag2, r), SC(Ag1, Ag2, q))); }
   Else { Δ(SC(Ag1, Ag2, q), SC(Ag1, Ag2, q)) := Justify;
      Send(Ag2, Δ(SC(Ag1, Ag2, q), SC(Ag1, Ag2, q)));}}
```

Algorithm 3

Algorithm 4 deals with the case of Ag_1 reaction if Ag_2 justifies the content of its commitment by itself. *Trustworthy*(Ag_2) is a Boolean function that enables Ag_1 to determine if Ag_2 is trustworthy or not. If according to Ag_1, Ag_2 is trustworthy, then Ag_1 accepts Ag_2's commitment. If not, Ag_1 refuses Ag_2's commitment. In the following section we propose a probabilistic model of trustworthiness to determine the value of *Trustworthy*(Ag_2) function.

```
If Δ(SC(Ag2, Ag1, q), SC(Ag2, Ag1, q)) = Justify Then {
   If Trustworthy(Ag2)
      Then α(Ag1, SC(Ag2, Ag1, q)):=Accept
      Else α(Ag1, SC(Ag2, Ag1, q)):= Refuse
   Send(Ag2, α(Ag1, SC(Ag2, Ag1, q)));
}
```

Algorithm 4

Algorithm 5 deals with the case where Ag_2 attacks the support of Ag_1's argument. Ag_1 attacks Ag_2's argument if it has an against-argument or it defends its argument if it has an argument or it accepts Ag_2's argument if it has an argument. If Ag_1 has no arguments nor against-arguments, then it challenges Ag_2's argument.

If $\Delta(SC(Ag_2, Ag_1, q), SC(Ag_1, Ag_2, r)) = Attack$ *Then {*
  If $(s, \neg q) \in S_{Ag1} / S'_{Ag1}$ *Then {*
    $\alpha(Ag_1, SC(Ag_1, Ag_2, s)) := Create;$
    $\Delta(SC(Ag_1, Ag_2, s), SC(Ag_2, Ag_1, q)) := Attack;$
    $S'_{Ag1} := S'_{Ag1} \cup \{(s, \neg q)\};$
    $Send(Ag_2, \Delta(SC(Ag_1, Ag_2, s), SC(Ag_2, Ag_1, q)));$ *}*
  Else If $(s, r) \in S_{Ag1} / S'_{Ag1}$ *Then {*
      $\alpha(Ag_1, SC(Ag_1, Ag_2, s)) := Create;$
      $\Delta(SC(Ag_1, Ag_2, s), SC(Ag_1, Ag_2, r)) := Defend;$
      $S'_{Ag1} = S'_{Ag1} \cup \{(s, r)\};$
      $Send(Ag_2, \Delta(SC(Ag_1, Ag_2, s), SC(Ag_1, Ag_2, r)));$*}*
    Else {
      If $(s, q) \in S_{Ag1} / S'_{Ag1}$ *Then*
        $\alpha(Ag_1, SC(Ag_2, Ag_1, q)) := Accept;$
      Else $\alpha(Ag_1, SC(Ag_2, Ag_1, q)) := Challenge;$
      $Send(Ag_2, \alpha(Ag_1, SC(Ag_2, Ag_1, q)));$*}}*

Algorithm 5

Algorithm 6 deals with the case in which the reactive game of Ag_2 is a defense of its argument. Thus, Ag_1 can attack the support of the Ag_2's argument or its conclusion according to Ag_1's arguments. As in Algorithm 5, Ag_1 accepts or challenges the support of Ag_2's argument in the opposite case.

5 Trustworthiness Model

Several models of trustworthiness have been developed in the context of MAS [20, 22, 31]. However, their formulations do not take into account the elements we use in our approach. For this reason, we propose a model that is more appropriate for our protocol. This model has the advantage of being simple and rigorous.

In our model, an agent's trustworthiness is a probability function defined as follows: $TRUST : A \times A \times D \rightarrow [0, 1]$. This function associates to each agent a probability measure representing its trustworthiness in the domain D according to another agent. Let X be a random variable representing an agent's trustworthiness. To evaluate the trustworthiness of an agent Ag_b, an agent Ag_a uses the records of its interactions with Ag_b. Formula 1 indicates how to calculate this trustworthiness as a probability measure (number of successful outcomes / total number of possible outcomes).

$$TRUST(Ag_b)_{Aga} = \frac{Nb\_\arg(Ag_b)_{Aga} + Nb\_SC(Ag_b)_{Aga}}{T\_Nb\_\arg(Ag_b)_{Aga} + T\_Nb\_SC(Ag_b)_{Aga}} . \quad (1)$$

If $\Delta(PC(Ag_2, Ag_1, q), PC(Ag_2, Ag_1, r)) = Defend$ *Then {*
  If $(s, \neg q) \in S_{Ag1} / S'_{Ag1}$ *Then {*
    $\alpha(Ag_1, SC(Ag_1, Ag_2, s)) := Create;$
    $\Delta(SC(Ag_1, Ag_2, s), SC(Ag_2, Ag_1, q)) := Attack;$
    $S'_{Ag1} := S'_{Ag1} \cup \{(s, \neg q)\};$
    $Send(Ag_2, \Delta(SC(Ag_1, Ag_2, s), SC(Ag_2, Ag_1, q)));\}$
  Else If $(s, \neg r) \in S_{Ag1} / S'_{Ag1}$ *Then {*
      $\alpha(Ag_1, SC(Ag_1, Ag_2, s)) := Create;$
      $\Delta(SC(Ag_1, Ag_2, s), SC(Ag_2, Ag_1, r)) := Attack;$
      $S'_{Ag1} := S'_{Ag1} \cup \{(s, \neg r)\};$
      $Send(Ag_2, \Delta(SC(Ag_1, Ag_2, s), SC(Ag_2, Ag_1, r)));\}$
    Else {
      If $(s, q) \in S_{Ag1} / S'_{Ag1}$ *Then*
        $\alpha(Ag_1, SC(Ag_2, Ag_1, q)) := Accept;$
      Else $\alpha(Ag_1, SC(Ag_2, Ag_1, q)) := Challenge;$
      $Send(Ag_2, \alpha(Ag_1, SC(Ag_2, Ag_1, q)));\}\}$

Algorithm 6

$Nb\_arg(Ag_b)_{Aga}$ is the number of Ag_bs' arguments that are accepted by Ag_a.
$Nb\_SC(Ag_b)_{Aga}$ is the number of satisfied commitments whose Ag_b is the debtor and Ag_a is the creditor.
$T\_Nb\_arg(Ag_b)_{Aga}$ is the total number of Ag_bs' arguments towards Ag_a.
$T\_Nb\_SC(Ag_b)_{Aga}$ is the total number of commitments whose Ag_b is the debtor and Ag_a is the creditor.

All these commitments and arguments are related to the domain D. The basic idea is that the trust degree of an agent can be induced according to how much information acquired from it has been accepted as belief in the past. Because all the factors of equation 1 are related to the past, this information number is finite.

$TRUST(Ag_b)_{Aga}$ is the trustworthiness of Ag_b according to Ag_a's point of view. This trustworthiness is a dynamic value that changes according to the interactions taking place between Ag_a and Ag_b. This supposes that Ag_a knows Ag_b. If not, or if the number of interactions is not sufficient to determine this trustworthiness, the consultation of other agents becomes necessary.

As proposed in [1, 31], each agent has two kinds of beliefs when evaluating the trustworthiness of another agent: local beliefs and total beliefs. Local beliefs are based on the direct interactions between agents. Total beliefs are based on the combination of the different testimonies of other agents called *witnesses*. In our model, local

beliefs are given by Formula 1. Total beliefs require studying how different probability measures offered by witnesses can be combined. We deal with this aspect in the following section.

5.1 Estimating Agent's Trustworthiness

Let us suppose that an agent Ag_a wants to evaluate the trustworthiness of an agent Ag_b with which it never (or not enough) interacted before. This agent must consult agents that it knows to be trustworthy (*confidence agents*). A trustworthiness threshold w must be fixed. Thus, Ag_b will be considered trustworthy for Ag_a iff $TRUST(Ag_b)_{Aga}$ is higher or equal to w. Ag_a attributes a trustworthiness measure to each confidence agent Ag_i. When it is consulted by Ag_a, each confidence agent Ag_i provides a trustworthiness value for Ag_b if Ag_i knows Ag_b. Confidence agents use their local beliefs to calculate this value (Formula 1). Thus, the problem consists in evaluating Ag_b's trustworthiness using the trustworthiness values transmitted by confidence agents.

We notice that this problem cannot be formulated as a problem of conditional probability. Consequently, it is not possible to use *Bayes' theorem* or *total probability theorem*. The reason is that events in our problem are not mutually exclusive, whereas this condition is necessary for these two theorems. Probability values offered by confidence agents are not mutually exclusive since they are provided simultaneously.

To solve this problem we must study the distribution of the random variable X. Since X takes only two values: 0 (the agent is not trustworthy) or 1 (the agent is trustworthy), variable X follows a Bernoulli distribution ß(1, p). According to this distribution, we have:

$$E(X) = p\ . \tag{2}$$

where $E(X)$ is the expectation of the random variable X *and* p is the probability that the agent is trustworthy. Thus, p is the probability that we seek. Therefore, *it is enough to calculate the expectation* $E(X)$ *to find* $TRUST(Ag_b)_{Aga}$. However, this expectation is a theoretical mean that we must estimate. To this end, we can use the *Central Limit Theorem* (CLT) and the *law of large numbers*. The CLT states that whenever a random sample of size n $(X_1,\ldots X_n)$ is taken from any distribution with mean μ, then the sample mean $(X_1 + \ldots + X_n) / n$ will be approximately normally distributed with mean μ. As an application of this theorem, the arithmetic mean (average) $(X_1+\ldots+ X_n)/n$ approaches a normal distribution of mean μ, the expectation Generally, and according to the law of large numbers, the expectation can be estimated by the weighted arithmetic mean.

Our random variable X is the weighted average of n independent random variables X_i that correspond to Ag_b's trustworthiness according to the point of view of confidence agents Ag_i. These random variables follow the same distribution: the Bernoulli distribution. They are also independent because the probability that Ag_b is trustworthy according to an agent Ag_t is independent of the probability that this agent (Ag_b) is trustworthy according to another agent Ag_r. Consequently, the random variable X follows a normal distribution whose average is the weighted average of the expectations of the independent random variables X_i. The estimation of expectation $E(X)$ is given by Formula 3.

$$M = \frac{\sum_{i=1}^{n} TRUST(Ag_i)_{Aga} N(Ag_i)_{Agb} TRUST(Ag_b)_{Agi}}{\sum_{i=1}^{n} TRUST(Ag_i)_{Aga} N(Ag_i)_{Agb}} \quad . \tag{3}$$

The value M represents an estimation of $TRUST(Ag_b)_{Aga}$ where $N(Ag_i)_{Agb}$ indicates the number of interactions between a confidence agent Ag_i and Ag_b. This number can be identified by the total number of Ag_b's commitments and arguments. This formula shows how trust can be obtained by merging the trustworthiness values transmitted by some mediators. This merging method takes into account the proportional relevance of each trustworthiness value, rather than treating them equally. This formula gives us a good estimation of $TRUST(Ag_b)_{Aga}$ that takes into account the three most important factors: (1) the trustworthiness of confidence agents according to the point of view of Ag_a (2) the Ag_b's trustworthiness according to the point of view of confidence agents (3) the number of interactions between confidence agents and Ag_b. This number is an important factor because it makes it possible to favor information coming from agents knowing more Ag_b. The function $Trustworthy(Ag_y)$ can be specified as follows:

If M > w Then Return true Else return false.

According to (3), we have :

$$\forall i, TRUST(Ag_b)_{Agi} < w \Rightarrow M < w. \frac{\sum_{i=1}^{n} TRUST(Ag_i)_{Aga} N(Ag_i)_{Agb}}{\sum_{i=1}^{n} TRUST(Ag_i)_{Aga} N(Ag_i)_{Agb}}$$

$$\Rightarrow M < w$$

Consequently, the well-known *lottery paradox* of Kyburg can never happen. If all trustworthiness values transmitted by the mediators are below the threshold w, then Ag_a will not trust Ag_b.

To calculate M, we need the trustworthiness of other agents. A practical solution consists in building a *trust graph* like the *TrustNet* proposed by [31].

6 Implementation

The algorithms and the trustworthiness model presented in this paper are implemented using *Jack*^{TM} technology. *Jack*^{TM} is an agent-oriented language offering a framework for MAS development. It is built on top of and fully integrated with Java programming language. The implemented prototype enabled us to verify the correctness of our algorithms and that the persuasion dynamics terminates because it converges to an acceptance or a refusal of the conversation subject. An agent accepts the conversation subject presented by $SC(p)$ or $SC(\neg p)$ if it accepts the last argument presented by its interlocutor using its argumentation system or because this interlocutor is trustworthy.

Agents' knowledge are implemented using *Jack*^{TM} data structures called *beliefsets*. The argumentation systems are implemented as Java modules using a logical programming paradigm. These modules use agents' beliefsets to build arguments for or against certain propositional formulas. The actions that agents perform on commitments or on their contents (presented by the functions α *and* Δ) are programmed as *events*. When an agent receives such an event, it seeks a *plan* to handle it. These plans are the algorithms presented in the paper.

The trustworthiness model is implemented using the same principle (events + plans). The requests sent by an agent about the trustworthiness of another agent are events and the calculations are programmed in plans. The trust graph is implemented as a Java data structure (oriented graph). Fig. 2 illustrates an example generated by our prototype of the process allowing an agent Ag_1 to measure the trustworthiness of another agent Ag_7 in a given domain.

Fig. 2. Example of process of trustworthiness measure

Fig. 3 illustrates an abstract example of the persuasion dynamics. In this figure an argument is denoted (*[Support],Conclusion*).

7 Discussion

In this section we discuss three fundamental characteristics of our algorithms: termination, complexity and correctness.

1. Termination. To prove the termination of Algorithm 1, it is enough to prove that the protocol dynamics always converges to a final acceptance or a final refusal.

According to the Algorithms 2, 3, 4 ,5 and 6, the protocol chaining can have one of the following possibilities:

Fig. 3. Example of persuasion dynamics

1- Agent Ag_2 accepts all the supports of the initial commitment $SC(Ag_1, Ag_2\ p)$. Therefore, we have: $\alpha(Ag_2, SC(Ag_1, Ag_2, p)) = Accept$.

2- Agent Ag_2 refuses one of the supports of $SC(Ag_1, Ag_2, p)$, *and* Ag_1 *does not find an argument to defend this support*. Thus, we have: $\alpha(Ag_2, SC(Ag_1, Ag_2, p)) = Refuse$.

3- The two agents attack each other about a part of the last arguments.

4- Agent Ag_2 challenges a part of the arguments presented by Ag_1.

Possibilities 1 and 2 converge to a final acceptance and a final refusal. Possibility 3 converges to a situation where an agent finds an argument *(H, h)* to attack the support of the interlocutor's argument, but this argument was already used ($(H, h) \in S'_{Ag}$). The reason is that the agents' knowledge bases are finite. In this case, this agent refuses the interlocutor's argument (Algorithm 2). Thus, possibility 3 converges to a final refusal. For the same reason, possibility 4 converges to the situation in which Ag_1 justifies a support by itself. In this situation, Ag_2 can play only an acceptance move if Ag_1 is trustworthy or a refusal move if not (Algorithm 4). Thus, possibility 4 converges to a final acceptance or a final refusal.

2. Complexity. The purpose of Algorithm 1 is to resolve the initial conflict or to decide after a finite number of moves that the conflict can not be resolved. Every move is based on the state of S_{Ag} and S'_{Ag} because agents must seek arguments or counter-arguments in S_{Ag} and S'_{Ag}. If we do not take into account the trustworthiness part of the algorithm, and since $|S_{Ag}| < |S'_{Ag}|$, the time complexity of algorithm 1 is $O(max(|S_{Ag1}|, |S_{Ag2}|))$. Thus the complexity is linear in the size of the knowledge bases of the agents. Before dealing with the complexity of the trustworthiness part, we introduce the following definition of the trust graph.

Definition 5. *A trust graph is a directed and weighted graph. The nodes are agents and an edge (Ag_i, Ag_j) means that agent Ag_i knows agent Ag_j. The weight of the edge (Ag_i, Ag_j) is a pair (x, y) where x is the Ag_j' trust according to the point of view of Ag_i and y is the interaction number between Ag_i and Ag_j. The weight of a node is the agent trust according to the point of view of the source agent.*

According to this definition, in order to determine the trustworthiness of the target agent Ag_b, it is necessary to find the weight of the node representing this agent in the graph. The algorithm is based on the construction of the graph and on a recursive call to assess the weight of all the nodes. Since each node is visited exactly once, there are *n* recursive calls, where *n* is the number of nodes in the graph. To assess the weight of a node we need the weights of its neighboring nodes and the weights of the input edges. Thus, the algorithm takes a time in $O(n)$ for the recursive calls and a time in $O(a)$ to assess the agents' trust where *a* is the number of edges. The run time of the trustworthiness algorithm is therefore in $O(max(a, n))$ i.e. linear in the size of the graph. In total, Algorithm 1 takes a time in $O(max(|S_{Ag1}|, |S_{Ag2}|) + max(a, n)) = O(max(|S_{Ag1}|, |S_{Ag2}|, a, n))$.

3. Correctness. We formalize the correctness problem of our algorithms as follows: Algorithm 1 is correct iff the protocol description based on this algorithm satisfies the protocol specification (i.e. what the protocol must do). The specification can be formalized as a set of claims or properties. The idea is to describe the protocol by a formal model *M* using a Kripke structure, and to express the specification as a logical formula ψ using our DCTL*$_{CAN}$ logic [6]. This formalization enables us to deal with the correctness problem as a model-checking problem, i.e. whether $M \models \psi$ or not. For this reason, it is possible to use the well-known model-checking technique for the CTL* fragment of our logic. However, resolving this problem for the all DCTL*$_{CAN}$ logic needs to develop a new model-checking technique for dynamic and temporal properties. The solution we are investigating as a future work is to use a combination of an automata-theoretic approach and a tableau-based approach [7].

8 Related Work

Smith et al [26] developed protocols having the advantage of being based on a logical theory (the theory of joint intention) that suggests how protocols can be linked together to form more complex interactions. However, these protocols do not take into account how different strategies can be chosen. Because our protocol uses DGs, it is possible to combine it with other protocols (information seeking, negotiation, …). Semantically, the protocols proposed by Smith et al. are based on private attitudes whereas we use a public and argumentative semantics.

Yolum and Singh [30] developed an approach for specifying protocols in which actions' content is captured through agents' commitments [25]. They provide operations and reasoning rules to capture the evolution of commitments. Using these rules, agents can reason about their actions. In a similar way, Fornara and Colombetti [12] proposed a method to define interaction protocols. This method is based on the specification of an interaction diagram (ID) specifying which actions can be performed under given conditions. The advantage of these approaches is to be verifiable because they are based on public notions. They also allow us to represent the interaction dynamics through the allowed operations. Our protocol is comparable to these protocols because it is also based on commitments. However, it is different in the following respects. The choice of the various operations is explicitly dealt with in our protocol by using argumentation and trustworthiness. The CAN formalism used to represent the protocol enables us to distinguish the various operations applied to commitments and to their contents as well as the argumentation relations. In addition, our protocol uses a specification based on philosophical foundations that allow us to specify the interaction dynamics.

To tackle the problem of the lack of flexibility in protocols, Reed [21], Dastani et al. [9], Maudet and Chaib-draa [13], and Dignum et al. [10] proposed protocols based on DGs. These protocols can be composed of various operations: sequencing, chaining, etc. Our protocol belongs to this family of protocols. However, our approach based on commitments and arguments makes our protocol different in terms of the allowed actions and in terms of the specification that our protocol has. In addition, our protocol clearly indicates how agents can choose a strategy using argumentative and social notions.

Parsons et al. [16], Amgoud et al. [2], McBurney [15], Sadri et al. [23] proposed protocols based on an argumentative approach. These protocols are based on Walton and Krabbe's classification and on formal dialectics. In these protocols, agents can argue about the truth of propositions. Agents can communicate both propositional statements and arguments about these statements. These protocols have the advantage of taking into account the capacity of agents to reason as well as their attitudes (confident, careful,…). Semantically, these protocols are specified by defining pre/post conditions for each locution. The difference between these protocols and ours is that our protocol deals with the social aspects of the interaction in its specification by integrating the notion of trustworthiness. In addition, we use a hybrid approach based on commitments and arguments. Our protocol is specified not by pre/post conditions, but by algorithms specifying the entry conditions, the exit conditions and

the dynamics. Particularly, there are other differences between our protocol and that of Parsons et al.: 1. From the theoretical point of view, Parsons et al.'s protocol uses moves from formal dialectics, whereas our protocol uses actions that agents apply on commitments. These actions capture the speech acts that agents perform when conversing (see Definition 1). The advantage of using these actions is that they enable us to better represent the persuasion dynamics considering that their semantics can be defined in an unambiguous way in a dynamic logic. 2. Parsons et al.'s protocol uses only three moves: assertion, acceptance and challenge, whereas our protocol uses, over and above creation, acceptance, refusal and challenge actions, attack and defense actions in an explicit way. These argumentation relations allow us to directly illustrate the concept of dispute in this type of protocols. 3. Parsons et al. use an acceptance criterion directly related to the argumentation system, whereas we use an acceptance criteria for the agents (supports of arguments and trustworthiness). This makes it possible to decrease the computational complexity of the protocol for agent communication.

9 Conclusion and Future Work

In this paper we proposed a new persuasion protocol based on DGs. This protocol is presented within a social and argumentative approach. Using our CAN formalism, this protocol is specified by indicating its entry conditions, exit conditions and dynamics. This protocol is characterized by the fact that it integrates trustworthiness as a component of the decision-making process. We described the implementation of this protocol using an agent platform.

As future work, we intend to specify other protocols according to Walton and Krabbe's classification and Vanderveken's typology. Another objective of this research is to verify some formal properties of these protocols (termination, soundness, …) using model-checking techniques. The idea we are investigating is to use a tableau method and an automata theoretic approach to branching time model checking. Thus, to prove that our protocol M verifies some properties ψ, we have to verify that $M \models \psi$ which is a model-checking problem.

Acknowledgments

We would like to thank John-Jules Ch. Meyer for his valuable suggestions. We would also like to thank the two anonymous referees. Their detailed and very interesting comments allowed us to improve the quality of this paper.

References

1. Abdul-Rahman, A. and Hailes. S. Supporting Trust in Virtual Communities. In Proc. Of the 33rd Hawaii Int. Conf. On Systems Science (2000) .
2. Amgoud, L., Maudet, N., and Parsons, N. Modelling dialogues using argumentation. In Proc. of the 4th Int. Conf. on MAS (2000) 31-38.

3. Bench-Capon, T.J.M., Freeman J.B., Hohmann, H., and Prakken, H. Computational models, argumentation theories and legal practice. In Reed, C. and Norman, T.J. (eds.). Argumentation Machines. New Frontiers in Argument and Computation. Kluwer Argumentation Library (2003) 85-120.
4. Bentahar, J., Moulin, B., and Chaib-draa, B. Vers une approche à base d'engagements et d'arguments pour la modélisation du dialogue. In Modèles Formels de l'Interaction, Cépaduès (2003) 19-28.
5. Bentahar, J., Moulin, B., and Chaib-draa, B. Commitment and argument network: a new formalism for agent communication. Dignum, F. (ed.). Advances in Agent Communication. Lecture Notes in Artificial Intelligence, vol. 2922. Springer-Verlag, (2003) 146-165.
6. Bentahar, J., Moulin, B., Meyer, J-J, Ch., and Chaib-draa, B. A logical model for commitment and argument network (extended abstract). In Proc. Of the 3rd Int. J. Conf. On AAMAS (2004) 792-799.
7. Bhat, G., Cleaveland, R., and Groce, A. Efficient model checking via Büchi tableau automata. In Berry, G., Comon, H., and Finkel, A. (eds). Computer-Aided Verification, Lecture Notes in Computer Scuence, vol. 2102. Springer-Verlag, (2001) 38-52.
8. Castelfranchi, C. Commitments: from individual intentions to groups and organizations. In Proc. of the Int. Conf. on Multi-Agent Systems (1995) 41-48.
9. Dastani, M., Hulstijn, J. and der Torre, L. V. Negotiation protocols and dialogue games. In Proc. of the Belgium/Dutch AI Conf. (2000) 13-20.
10. Dignum, F., Dunin-Keplicz, and Verbugge, R., Creation collective intention through dialogue. Logic Journal of the IGPL, 9(2) (2001) 305-319.
11. Elvang-Goransson, M., Fox, J., and Krause, P. Dialectic reasoning with inconsistent information. In Proc. of the 9th Conf. on Uncertainty in AI. (1993) 114-121.
12. Fornara, N. and Colombetti, M. Defining protocols using a commitment-based agent communication language. In Proc. Of the 2nd Int. J. Conf. On AAMAS (2003) 520-527.
13. Maudet, N. and Chaib-draa, B. Commitment-based and dialogue-game based protocols, new trends in agent communication languages. Knowledge Engineering Review, 17(2), Cambridge Univ. Press (2002) 157-179.
14. McBurney, P. and Parsons, S. Games that agents play: A formal framework for dialogues between autonomous agents. Journal of Logic, Language, and Information, 11(3) (2002) 1-22.
15. McBurney, P. Rational Interaction. Thesis of Univ. of Liverpool (2002).
16. Parsons, S., Wooldridge, M., and Amgoud, L. On the outcomes of formal inter-agent Dialogues. In Proc. Of the 2nd Int. J. Conf. On AAMAS (2003) 616-623.
17. Prakken, H. Logical Tools for Modelling legal argument. A study of defeasible reasoning in law. Kluwer Law and Philosophy Library (1997).
18. Prakken, H. and Sartor, G. Modelling reasoning with precedents in a formal dialogue game. Artificial Intelligence and Law, vol.6 (1998) 231-287.
19. Prakken, H. and Vreeswijk, G. Logics for defeasible argumentation. Gabbay, D., (ed.), Handbook of Philosophical Logic, Kluwer (2000) 218-319.
20. Ramchurn, S.D., Sierra, C., Jennings, N.R., and Godo L. A computational trust model for multi-agent interactions based on confidence and reputation. In Proc. of 6th Int. Workshop of Deception, Fraud and Trust in Agent Societies (2003) 69-75.
21. Reed, C. Dialogue frames in agent communication. In Proc. of the 3rd Int. Conf. on MAS. (1998) 246-253.
22. Sabater, J. and Sierra, C. Reputation and social network analysis in multi-agent systems. In Proc. Of the 1st Int. J. Conf. On AAMAS (2002) 475-482.
23. Sadri, F., Toni, F., Torroni, P. Logic agents, dialogues and negotiation: an abductive approach. In Proc. of Symposium on Information Agents for E-Commerce (2001).

24. Searle, J.R. Speech acts: an essay in the philosophy of language. Cambridge University Press, England (1969).
25. Singh, M.P. Agent communication languages: rethinking the principles, IEEE Computer (1998) 40-47.
26. Smith, I.A., Cohen, P.R., Bradshaw, J.M., Greaves, M., and Holmback, H. Designing conversation policies using joint intention theory. In Proc. of the 3rd Int. Conf on MAS (1998) 269-276.
27. Vanderveken, D., Illocutionary logic and discourse typology. Searle with his Replies of Revue Internat. de Philosophie. 2001.
28. Vreeswijk, G.A.W. Abstract argumentation systems. Artificial Intelligence, 90 (1-2), 1997, 225-279.
29. Walton, D.N. and Krabbe, E.C.W. Commitment in dialogue: basic concepts of interpersonal reasoning. State Univ. of New York Press, Albany, NY (1995).
30. Yolum, P. and Singh, M.P. Flexible protocol specification and execution: applying event calculus planning using commitments. In Proc. of Int. J. Conf. On AAMAS (2002) 527-534.
31. Yu, B. and Singh, M. An evidential model of distributed reputation management. In Proc. Of the 1st Int. J. Conf. On AAMAS (2002) 294-301.