A Logical Model for Commitment and Argument Network for Agent Communication (Extended Abstract)

Jamal Bentahar
Laval University, Computer Science Department. Canada
jamal.bentahar.1@ulaval.ca

Bernard Moulin
Laval University, Computer Science Department. Canada
bernard.moulin@ift.ulaval.ca

John-Jules Ch. Meyer
University Utrecht, Department of Computer Science. The Netherlands
jj@cs.uu.nl

Brahim Chaib-draa
Laval University, Computer Science Department. Canada
chaib@iad.ift.ulaval.ca

Abstract

[1]*In this paper we present a semantics for our approach based on social commitments (SCs) and arguments for conversational agents. More precisely, we propose a logical model based on CTL* and on dynamic logic (DL). Called Commitment and Argument Network, our formal framework based on this approach uses three basic elements: SCs, actions that agents apply to these SCs and arguments that agents use to support their actions. The advantage of this logical model is to bring together all these elements and the relations existing between them within the same framework. Our semantics makes it possible to represent the dynamics of agent communication. It also allows us to establish the important link between SCs as a deontic concept and arguments. CTL* enables us to express the temporal characteristics of SCs and arguments. DL enables us to capture the actions that agents are committed to achieve.*

1. Introduction

In the domain of agent communication (AgC), semantics is one of the most important aspects, particularly in the current context of open and interoperable multi-agent systems (MAS) [5]. Although a certain number of significant research works were done in this field [11, 19, 21, 22], the definition of a clear and global semantics is an objective yet to be reached. AgC pragmatics is another important aspect to be addressed. While semantics is interested in the meaning of communication acts, pragmatics deals with the way of using these acts. Pragmatics is related to the dynamics of agent interactions and to the way of relating the isolated acts to build conversations. Pragmatics was also addressed by several researchers [7, 15, 17, 18]. However, only few attempts have been made to address these two facets of AgC in the same framework.

The objective of this paper is to propose a general framework to capture pragmatic and semantic issues of an approach based on social commitments (SCs) and arguments for AgC. Indeed, this work is a continuation of our previous research in which we addressed in detail the pragmatic aspects [2, 3]. Thus, the paper highlights the semantic issues of our approach and the link with pragmatic ones. The semantics that we define here deals with all the aspects used in our approach.

In addition to proposing a unified framework for pragmatic and semantic issues, this work presents two results: 1) it semantically establishes the link between SCs and arguments; 2) it uses both a temporal logic (CTL* with some additions) and a dynamic logic (DL) to define a complete and unambiguous semantics.

Paper overview. In Section 2 we address the pragmatic aspects by introducing the main ideas of our approach. In Section 3 we present the syntax and the semantics of the main elements of our logical model. Other details will be described in an extended version of the paper. In Sections 4 and 5 we compare our approach to related work and we conclude the paper.

2. SC and Argument-based Approach

2.1. Social Commitments

A SC is a commitment made by an agent (called the *debtor*), that some fact is true [4]. This commitment is directed to a set of agents (called *creditors*). The SC content is characterized by time t_φ, which is generally different from the utterance time denoted t_u and from the time associated with the SC denoted t_{sc}. t_φ is the time described by the utterance, and thus by the content φ. Time t_{sc} refers to the time during which the SC holds. When it is an interval, this time is denoted $[t_{sc}^{\inf}, t_{sc}^{\sup}]$. If the SC is satisfied or violated we have $t_{sc}=[t_u, t_\varphi]$. However, if the SC is withdrawn, we have: $t_{sc}=[t_u, t_w]$, with t_w the withdrawal time. Time t_{sc} indicates the time during which the SC holds, i.e. the time during which the SC is *active* (we will return to this notion later). Time t_φ indicates the moment at which the SC must be satisfied.

In order to model the dynamics of conversations, we interpret a speech act SA as an action performed on a SC

[1] Student Paper

or on the content of a SC. A SA is an abstract act that an agent, *the speaker*, performs when producing an utterance *U* and addressing it to another agent, *the addressee*. The actions that an agent can perform on a SC are: *Act∈{Create, Withdraw, Reactivate, Violate, Satisfy}*. The actions performed on the content of a SC are *Act-content∈{Accept, Refuse, Challenge, Change}*. Thus, a SA leads either to an action on a SC when the speaker is the debtor, or to an action on a SC content when the speaker is the debtor or the creditor. Formally, in our framework a SA can be defined in BNF form as follows:

Definition 1: $SA(i_k, Ag_1, Ag_2, t_u, U) =_{déf}$
$Act(Ag_1, t_u, SC(id_n, Ag_1, Ag_2, t_{sc}, \varphi, t_\varphi))$
$\mid Act\text{-}content(Ag_k, t_u, SC(id_n, Ag_i, Ag_j, t_{sc}, \varphi, t_\varphi))$
where $i, j \in \{1, 2\}$ and $(k=i$ or $k=j)$, $=_{def}$ means "is interpreted by definition as", i_k is the identifier of the SA. The definiendum $SA(i_k, Ag_1, Ag_2, t_u, U)$ is defined by the definiens $Act(Ag_1, t_u, SC(id_n, Ag_1, Ag_2, t_{sc}, \varphi, t_\varphi))$ as an action performed by the debtor Ag_1 on its SC. The definiendum is defined by the definiens $Act\text{-}content(Ag_k, t_u, SC(id_n, Ag_i, Ag_j, t_{sc}, \varphi, t_\varphi))$ as an action performed by an agent Ag_k (the debtor or the creditor) on the SC content.

2.2. Taxonomy

In this section, we explain the various types of SCs we use in the logical model:

A. Absolute Commitments (ABCs): They are SCs whose fulfillment does not depend on any particular condition. Two types can be distinguished:

A1. Propositional Commitments (PCs): They are related to the state of the world and expressed by assertives.

A2. Action Commitments (ACs): They are always directed towards the future and are related to actions that the debtor is committed to carrying out. This type of SCs is typically conveyed by promises.

B. Conditional Commitments (CCs): In several cases, agents need to make SCs not in absolute terms but under given conditions. CCs allow us to express that if a condition β is true, then the creditor will be committed towards the debtor to making γ or that γ is true.

C. Commitment Attempts (CTs): The SCs described so far directly concern the debtor who commits either that a certain fact is true or that a certain action will be carried out. These SCs do not allow us to explain the fact that an agent asks another one to be committed to carrying out an action. To solve this problem, we propose the concept of CT. We consider a CT as a request made by a debtor to push a creditor to be committed.

We notice that there is no explicit relation between PCs and ACs. When the current state of the world does not satisfy a PC we speak about a violation of this SC. There is no rule indicating that the agent develops an AC to make the content of a PC true when this PC becomes violated.

2.3. 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 [1]. In our model, we adopt the following definition from [8]. 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 that no subset of H satisfying both i and ii exists. H is called the support of the argument and h its conclusion.*

The link between SCs and arguments enables us to capture both the public and reasoning aspects of AgC. This link is explained as follows. Before committing to some fact *h* being true (i.e. before creating a SC 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. For example, an agent Ag_1 accepts the SC content *h* proposed by another agent Ag_2 if it is able to build an argument which supports this content from its knowledge base. If Ag_1 has an argument neither for *h*, nor for $\neg h$, then it must ask for an explanation.

The argumentation relations that we use in our model are thought of as actions applied to SC contents. The set of these relations is: *{Justify, Defend, Attack, Contradict}*.

We used this approach in [3] to propose a formal framework called Commitment and Argument Network (CAN). The idea is to reflect the dynamics of AgC by a network in which agents manipulate SCs and arguments. In the following section we propose a formal semantics of this formalism in the form of a logical model.

3. The Logical Model

3. 1. Syntax

In this section we specify the syntax of the main elements we use in our framework. The details of the other elements are described in an extended version of the paper. Our formal language *£* is based on an extended version of CTL* [9] and on DL [12]. We use a branching time for the future and we suppose that the past is linear (Figure 1). We also suppose that time is discrete. Let Φp be the set of atomic propositions and Φa the set of action symbols. The set of the agents is denoted *A* and the set of time units is denoted *TU*. The various types of SCs, the actions of the agents on SCs and on their contents and the argumentation relations are introduced as modal operators. We denote *£sc* a sub-language of *£* for SCs. To simplify the notation, a SC, independently of its type, is denoted:

$SC(Id_0, Ag_1, Ag_2, \varphi)$. $Id_0 \in N$ is the SC identifier, Ag_1 and Ag_2 are two agents and φ the SC content. The language £ can be defined by the following syntactic rules.

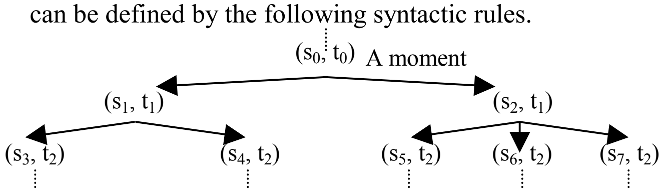

Figure 1. The branching time model

3.1.1. Propositional Elements.

R1. $\forall \phi \in \Phi p, \phi \in \pounds$: *Atomic formula*

R2. $p, q \in \pounds \Rightarrow p \wedge q \in \pounds$: *Conjunction*

R3. $p \in \pounds \Rightarrow \neg p \in \pounds$: *Negation*

R4. $p, q \in \pounds \Rightarrow p \therefore q \in \pounds$: *Argumentation*

This means that p is an argument for q. We can read this formula: *p, so q*. At this level, our definition of the argument does not take into account the defeasible aspect. This aspect will be introduced into our model by the argumentation relations (Section 3.1.6).

R5. $p \in \pounds \Rightarrow ?p \in \pounds$: *Is p true?*

R6. $p \in \pounds \Rightarrow Ap \in \pounds$: *Universal path-quantifier*

R7. $p \in \pounds \Rightarrow Ep \in \pounds$: *Existential path-quantifier*

R8. $p, q \in \pounds \Rightarrow p\, U^+ q \in \pounds$: ***Until (in the future)***

Informally, $p\ U^+\ q$ (*p until q*) means that on a given path from the given moment, there is some future moment in which q will eventually hold and p holds at all moments until that future moment.

R9. $p \in \pounds \Rightarrow X^+ p \in \pounds$: ***Next moment (in the future)***

X^+p holds at the current moment, if p holds at the next moment.

R10. $p, q \in \pounds \Rightarrow p\, U^- q \in \pounds$: ***Since (in the past)***

The intuitive interpretation of $p\ U^-\ q$ (*p since q*) is that on a given path from the given moment, there is some past moment in which q eventually held and p holds at all moments since that past moment.

R11. $p \in \pounds \Rightarrow X^- p \in \pounds$: ***Previous moment (in the past)***

X^-p holds at the current moment, if p held at the previous moment.

3.1.2. Actions.

R12. $p \in \pounds/\pounds sc, \alpha \in \Phi a \Rightarrow Perform(\alpha)p \in \pounds$: ***Action performance (about propositions)***

Perform(α)p is an operator from DL. It indicates that the achievement of action α makes the proposition p true.

R13. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc, \alpha \in \Phi a \Rightarrow$
$Perform(\alpha)SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds$: ***Action performance (about SCs)***

This indicates that the achievement of action α makes the social commitment $SC(Id_0, Ag_1, Ag_2, \varphi)$ true in our model.

3.1.3. Social Commitments.

R14. $p \in \pounds/\pounds sc \wedge Id_0 \in N \wedge \{Ag_1, Ag_2\} \subseteq A \Rightarrow$
$PC(Id_0, Ag_1, Ag_2, p) \in \pounds sc$: *Propositional commitment*

R15. $\alpha \in \Phi a \wedge p \in \pounds/\pounds sc \wedge Id_0 \in N \wedge \{Ag_1, Ag_2\} \subseteq A \Rightarrow$
$AC(Id_0, Ag_1, Ag_2, \alpha)p \in \pounds sc$: *Action commitment*

R16. $\beta \in \pounds/\pounds sc \wedge \gamma \in \pounds/\pounds sc \cup \Phi a \wedge Id_0 \in N \wedge \{Ag_1, Ag_2\} \subseteq A \Rightarrow CC(Id_0, Ag_1, Ag_2, \beta \Rightarrow \gamma) \in \pounds sc$: *Conditional commitment*

In order to formally introduce the notion of CT we need some definitions from first order logic.

Definitions:

TerC: a set of constant terms. A constant term can be a number, a name, etc.

Var: a set of variables.

Val: Var $\mapsto$ *TerC*: a valuation function associating a variable to a constant term.

Let Ξ_{Val} be a substitution that makes it possible to substitute each free variable x that appears in a formula φ by a constant term, i.e. by $Val(x)$. We denote a formula φ in which appears a sequence of free variables X by $?X\varphi$. The expression $?X\varphi \bullet \Xi_{Val}$ indicates the formula φ in which each variable x of the sequence of free variables X is substituted by a corresponding value (i.e. by $Val(x)$). Thus, we can define the syntax of a CT as follows:

R17. $?X\varphi \in \pounds/\pounds sc \wedge Id_0 \in N \wedge \{Ag_1, Ag_2\} \subseteq A \Rightarrow$
$CT(Id_0, Ag_1, Ag_2, ?X\varphi) \in \pounds sc$: *Commitment attempt*

3.1.4. Actions applied to Commitments.

R18. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Create(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Creation of a SC*

R19. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Withdraw(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Withdrawal*

R20. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Satisfy(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Satisfaction*

R21. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Violate(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Violation*

R22. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Active(SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *An active SC*

3.1.5. Actions applied to Commitment Contents.

R23. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Accept\text{-}content(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Acceptation of a SC content*

R24. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Challenge\text{-}content(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Challenge*

3.1.6. Argumentation Relations.

R25. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \wedge \varphi' \in \pounds/\pounds sc \Rightarrow$
$Justify\text{-}content(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi') \in \pounds sc$: *Justification*

R26. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \Rightarrow$
$Contradict\text{-}content(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi)) \in \pounds sc$: *Contradiction.* This relation means that an agent contradicts the content of its SC.

R27. $SC(Id_0, Ag_1, Ag_2, \varphi) \in \pounds sc \wedge \varphi' \in \pounds/\pounds sc \Rightarrow$
$Attack\text{-}content(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi') \in \pounds sc$: *Attack of a SC content*

R28. SC(Id_0, Ag_1, Ag_2, φ) ∈ £sc ∧ φ' ∈ £/£sc ⇒ Defend-content(Ag_2, SC(Id_0, Ag_1, Ag_2, φ), φ') ∈ £sc: *Defense of a SC content against an attacker*

Abbreviations: We use in our model the following abbreviations:

A1. p ∨ q (disjunction) is the abbreviation of ¬(¬p ∧ ¬q)

A2. p ⇒ q (implication) is the abbreviation of ¬p ∨ q

A3. true is the abbreviation of p ∨ ¬p

A4. false is the abbreviation of ¬true

A5. F^+p (sometimes in the future) is the abbreviation of true U^+ p

A6. G^+p (globally in the future) is the abbreviation of $\neg F^+\neg p$

A7. F^-p (sometimes in the past) is the abbreviation of true U^- p

A8. G^-p (globally in the past) is the abbreviation of $\neg F^-\neg p$

3.2. Semantics

In this section, we define the formal model in which we evaluate the well-formed formulas of our framework. Thereafter, we give the semantics of the different elements that we specified syntactically in the previous section.

3.2.1. The Formal Model. Let *S* be a set of states. A path *Pa* is an infinite sequence of states $<s_0, s_1,...>$ where $T(s_0) < T(s_1) <$.... The function *T* gives us for each state s_i the corresponding moment *t* (this function will be specified later). Generally, for all *i* and *j* of *N*, if $i < j$ and s_i and s_j belong to the same path *Pa*, then $T(s_i) < T(s_j)$. We denote the set of all paths by σ. The set of all paths starting from the state s_i are denoted: σ^{si}. In our vision of branching future, we can have several states at the same moment. Thus, in Figure 1 we have two different states: s_1 and s_2 at the same time t_1. At moment t_2 we have the states s_3, s_4, s_5, s_6, s_7. Only along a given path (for example the real path) there is one and only one state at one moment. Indeed, in our framework, s_i does not indicate (necessarily) the state at moment *i*. Therefore, it is necessary to specify the state *s* and the moment *t* i.e. a couple $(s, t) \in S \times TU$. According to this formalization, we can use the notation: $M, s_i, T(s_i) \models \psi$ to indicate that ψ is satisfied in the Kripke model *M* at the state s_i at the moment $T(s_i)$. To simplify this notation, we will use in the rest of the paper the following notation: $M, s_i \models \psi$. In this notation: $M, s_i \models \psi$ there is a "hidden" time. Following this simplification we can write:

$M, s_i, T(s_i) \models \psi$ *iff* $M, s_i \models \psi$.

The formal model for £ is defined as follows:

M(S, A, Np, Np?, Fap, Rpc, Rac, T) where

S : a nonempty set of states.

A : a nonempty set of agents.

$\boldsymbol{Np : S \mapsto 2^{\Phi ap}}$: function relating each state $s \in S$ to the set of the atomic propositions that are true in this state.

$\boldsymbol{Np? : S \mapsto 2^{\Phi ap}}$: function relating each state $s \in S$ to the set of the atomic propositions that are neither true nor false in this state (i.e. we do not know if they are true or false).

$\boldsymbol{Fap : S \times \Phi a \mapsto 2^S}$: function that gives us the state transitions caused by the achievement of an action.

$\boldsymbol{Rpc : A \times A \times S \mapsto 2^S}$: function producing the accessibility modal relations for PCs.

$\boldsymbol{Rac : A \times A \times S \mapsto 2^S}$: function producing the accessibility modal relations for AC.

$\boldsymbol{T : S \mapsto TU}$: function associating to any state s_i the corresponding time. For instance, in Figure 1 we have: $T(s_5) = t_2$.

The functions ***Rpc*** and ***Rac*** give us the states that correspond to the time t_φ, i.e. the states in which the SC created by an agent Ag_1 towards another agent Ag_2 ***must be*** satisfied. These functions allow us to define a deadline for determining whether a violation or a satisfaction occurs. They give us all the states corresponding to the time t_φ on all paths starting from the state at moment t_u. The fact that these two functions give us a set of states means that the SC must be satisfied whatever the future. Since there is only one real path, the SC is satisfied or is violated only in one state of the set given by *Rpc* and *Rac*. Indeed, the outputs of the functions *Rpc* and *Rac* are known only after the creation of the SC. Thus, this depends on the state in which the SC is created. For example, if we have: $s_j \in Rpc(Ag_1, Ag_2, s_i)$, then this means that at moment $T(s_i)$ agent Ag_1 is committed towards agent Ag_2 to satisfy a certain SC at moment $T(s_j)$. We can see that *Rpc* depends on the current moment $T(s_i)$.

As in CTL*, we have in our model path formulas and state formulas. We propose to evaluate the static formulas (the different types of SCs) as state formulas. These formulas can also be interpreted on paths in which case one considers satisfaction in the first state of a path. On the other hand, we propose to evaluate dynamic formulas (the actions on SCs) on paths. These path formulas can become state formulas if they are true on all the paths starting from a given state. $M, s_i \models \psi$ indicates that the formula ψ is evaluated in the state s_i of the model *M*. $M, Pa, s_i \models \psi$ indicates that the formula ψ is evaluated on the path *Pa* starting from the state s_i of the model *M*. We can now define the semantics of the elements of *£*.

3.2.2. Propositional Elements.

S1. $M, s_i \models \psi$ iff $\psi \in Np(s_i)$ with $\psi \in \Phi p$.

S2. $M, s_i \models p \wedge q$ iff $M, s_i \models p$ & $M, s_i \models q$

S3. $M, s_i \models \neg p$ iff $\neg(M, s_i \models p)$

S4. $M, s_i \models p \therefore q$ iff $M, s_i \models p$ & $(\forall j : M, s_j \models p \Rightarrow M, s_j \models q)$.

In S4 we add the first clause ($M, s_i \models p$) to capture the following aspect: when an agent presents an argument *p* for *q* (i.e. $p \therefore q$) for this agent *p* is true and if *p* is true then *q* is true. Indeed, *p so q* is stronger than just stating that both p and q are true. The implication is much stronger since it holds in all the states of the model M. The idea is to express that p is the support of the conclusion q.

S5. $M, s_i \models ?p$ iff $p \in Np?(s_i)$.

S6. $M, s_i \models Ap$ iff $(\forall Pa : Pa \in \sigma^{si} \Rightarrow M, Pa, s_i \models p)$

S7. $M, s_i \models Ep$ iff $(\exists Pa \in \sigma^{si}$ & $M, Pa, s_i \models p)$

S8. $M, Pa, s_i \models p$ iff $M, s_i \models p$: *Propositional path formulas*

S9. $M, Pa, s_i \models p \wedge q$ iff $M, Pa, s_i \models p$ & $M, Pa, s_i \models q$

S10. $M, Pa, s_i \models \neg p$ iff $\neg(M, Pa, s_i \models p)$

S11. $M, Pa, s_i \models p\ U^+\ q$ iff $(\exists j : i \leq j$ & $M, Pa, s_j \models q$ & $(\forall k : i \leq k < j \Rightarrow M, Pa, s_k \models p))$

S12. $M, Pa, s_i \models X^+p$ iff $M, Pa, s_{i+1} \models p))$

S13. $M, Pa, s_i \models p\ U^-\ q$ iff $(\exists j : j \leq i$ & $M, Pa, s_j \models q$ & $(\forall k : j < k \leq i \Rightarrow M, Pa, s_k \models p))$

S14. $M, Pa, s_i \models X^-p$ iff $M, Pa, s_{i-1} \models p))$

3.2.3. Actions.

S15. $M, Pa, s_i \models Perform(\alpha)p$ iff $\forall s_j : s_j \in Fap(s_i, \alpha) \wedge s_j \subset Pa \Rightarrow M, Pa, s_j \models p$.

where $s_j \subset Pa$ indicates that Pa, s_j is a *suffix* of Pa, s_i.

S16. $M, s_i \models Perform(\alpha)p$ iff $\forall Pa : Pa \in \sigma^{si} \Rightarrow$ $M, Pa, s_i \models Perform(\alpha)p$.

Action performance (related to SCs)

S17. $M, Pa, s_i \models Perform(\alpha)SC(Id_0, Ag_1, Ag_2, \varphi)$ iff $\forall s_j : s_j \in Fap(s_i, \alpha) \wedge s_j \subset Pa \Rightarrow$ $M, Pa, s_j \models SC(Id_0, Ag_1, Ag_2, \varphi)$.

This formula indicates that the achievement of action α makes the social commitment true in all the accessible states from the state s_i. As for S15, the accessible states are defined by the function *Fap*. The evaluation of this operator in a state is given by the following formula:

S18. $M, s_i \models Perform(\alpha)SC(Id_0, Ag_1, Ag_2, \varphi)$ iff $\forall Pa: Pa \in \sigma^{si} \Rightarrow M, Pa, s_i \models Perform(\alpha)SC(Id_0, Ag_1, Ag_2, \varphi)$.

3.2.4. Social Commitments.

Social commitment as a path formula

S19. $M, Pa, s_i \models SC(Id_0, Ag_1, Ag_2, \varphi)$ iff $M, s_i \models SC(Id_0, Ag_1, Ag_2, \varphi)$

S20. $M, s_i \models PC(Id_0, Ag_1, Ag_2, p)$ iff $(\forall s_j : s_j \in Rpc(Ag_1, Ag_2, s_i) \Rightarrow M, s_j \models p)$

S21. $M, s_i \models AC(Id_0, Ag_1, Ag_2, \alpha)p$ iff $\forall s_j : s_j \in Rac(Ag_1, Ag_2, s_i) \Rightarrow M, s_j \models Perform(\alpha)p))$.

The formula S21 indicates that agent Ag_1 is committed towards agent Ag_2 to do α and that in all accessible states s_j performing α makes p true. According to formulas S20 and S21, the semantics we give to the SCs requires their fulfillment. Thus, if it is created, a SC must be held. However, it is always possible to violate or withdraw such a SC. For this reason, these two operations (violation and withdrawal) are explicitly included in our framework. Thus, it is possible to have wrong SCs in the model. The reason is that *Rpc* and *Rac* give us only the states that correspond to the states in which the SC *must* be satisfied. These states are not conceived as merely "possible", but as states when the content of a SC *must* be true.

We notice that although *Rpc* and *Rac* are dynamic functions, we do not need to change the Kripke model M to capture this dynamics. This way of modeling is different from that used for example in KARO framework [16]. In our model which fits in naturally with CTL* the whole dynamics is represented in one unique model.

S22. $M, s_i \models CC(Id_0, Ag_1, Ag_2, \beta \Rightarrow \gamma)$ iff $(M, s_i \models EF^+\beta \Rightarrow M, s_i \models ABC(Id_0, Ag_1, Ag_2, \gamma))$

This formula indicates that agent Ag_1 commits to perform γ (or that γ is true) only if the condition β is true (or is satisfied).

In order to define the semantics of CTs, we define the binary relation $\models^{\Xi Val}$ between a pair (M, s_i) and a formula $CT(Id_0, Ag_1, Ag_2, ?X\varphi)$ as follows:

S23. $M, s_i \models^{\Xi Val} CT(Id_0, Ag_1, Ag_2, ?X\varphi)$ iff $(M, s_i \models EX^+F^+ABC(Id_0, Ag_2, Ag_1, ?X\varphi \bullet \Xi_{Val})$ $\vee (\exists \beta \in$ £/£sc : $M, s_i \models EX^+F^+CC(Id_0, Ag_2, Ag_1, \beta \Rightarrow ?X\varphi \bullet \Xi_{Val}))$

This formula indicates that a CT whose content is $?X\varphi$ is satisfied in the model M according to a substitution Ξ_{Val} iff the creditor (i.e. Ag_2) will commit that a content $?X\varphi \bullet \Xi_{Val}$ is true. In other words, the CT is satisfied iff the interlocutor will commit that the substitution Ξ_{Val} for the sequence X of free variables appearing in the formulae φ is true. The SC of the interlocutor can be absolute (ABC) or conditional (CC). We suppose here that agents are "dialogically" co-operative in so far as an agent accepts to offer a substitution Ξ_{Val} for the sequence X.

3.2.5. Actions applied to Commitments.

S24. $M, Pa, s_i \models Create(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi))$ iff $\exists \alpha \in \Phi a$ & $M, Pa, s_i \models Perform(\alpha)SC(Id_0, Ag_1, Ag_2, \varphi)$ $\wedge\ G^-\neg SC(Id_0, Ag_1, Ag_2, \varphi)$

This formula indicates that the creation of a SC is satisfied in the model M along a path Pa iff there is an action α whose performance makes true the SC (i.e. the SC holds after the performance of the action α) and if in the past (before the creation moment of the SC), the SC was never satisfied in this model. This formula highlights the fact that the creation of a SC is an action in itself. Indeed, the action α corresponds to the agent's utterance which creates the SC.

S25. $M, Pa, s_i \models Withdraw(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi))$ iff $\exists \alpha \in \Phi a, M, Pa, s_i \models$ $X^-F^-Create(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi))$ $\wedge\ Perform(\alpha)\neg SC(Id_0, Ag_1, Ag_2, \varphi)$.

This formula indicates that an agent withdraws its SC for φ iff: (1) The agent has already created this SC. (2) The agent performs an action α so that this SC does not hold at the current moment.

The semantics of the satisfaction operation depends on the type of the SC. In this paper we give only the semantics of the satisfaction of a PC as follows:

S26. M, Pa, $s_i \models$ Satisfy(Ag_1, PC(Id_0, Ag_1, Ag_2, p)) iff
$\exists j : j \leq i$ & M, Pa, $s_j \models$ Create(Ag_1, PC(Id_0, Ag_1, Ag_2, p))
$\wedge$ M, Pa, $s_i \models p \wedge s_i \in$ Rpc(Ag_1, Ag_2, s_j).

A PC is satisfied iff it was already created and the propositional content is true in the moment that corresponds to the moment where the SC must be satisfied. This moment is denoted by s_i that defines the deadline. For example, if an agent commits at 14PM that it will rain at 16PM, we say that the SC is satisfied if it really rains at 16PM, if not, the SC is violated.

We can think of satisfaction and violation as two dual relations. Hence, we can express the relation between satisfaction and violation for any SC type. For example, for a PC this relation is specified by the formula:

S27. M, Pa, $s_i \models$ Violate(Ag_1, PC(Id_0, Ag_1, Ag_2, φ)) iff
$\exists j : j \leq i$ & M, Pa, $s_j \models$ Create(Ag_1, PC(Id_0, Ag_1, Ag_2, φ))
$\wedge s_i \in$ Rpc(Ag_1, Ag_2, s_j)
$\wedge$ M, Pa, $s_i \models \neg$Satisfy(Ag_1, PC(Id_0, Ag_1, Ag_2, φ)).

This formula expresses the following property: If an agent violates its SC in the state s_i (which represents the deadline) along the path *Pa*, then this agent does not satisfy this SC in this state along this path and vice versa.

After introducing the different actions that the debtor can apply to its SCs, we can define the semantics of an active SC as follows:

S28. M, Pa, $s_i \models$ Active(SC(Id_0, Ag_1, Ag_2, φ)) iff
M, Pa, $s_i \models$ (($\neg$Violate(Ag_1, SC(Id_0, Ag_1, Ag_2, φ))
$\wedge \neg$Satisfy(Ag_1, SC(Id_0, Ag_1, Ag_2, φ))
$\wedge \neg$Withdraw(Ag_1, SC(Id_0, Ag_1, Ag_2, φ)))
U^- Create(Ag_1, SC(Id_0, Ag_1, Ag_2, φ)))

This property indicates that a SC is active iff : (1) This SC was already created. (2) Until the current moment, the SC was neither violated, withdrawn nor satisfied. Therefore, once the SC is satisfied, violated or withdrawn, it becomes inactive.

3.2.6. Actions applied to Commitment Contents.

S29. M, Pa, $s_i \models$
Accept-content(Ag_2, SC(Id_0, Ag_1, Ag_2, φ)) iff
M, Pa, $s_i \models$ Active(SC(Id_0, Ag_1, Ag_2, φ))
$\wedge$ Create(Ag_2, SC(Id_1, Ag_2, Ag_1, φ))

This formula indicates that the acceptance of the SC content φ by agent Ag_2 is satisfied in the model *M* along a path *Pa* iff: (1) The SC is active on this path because we cannot act on a SC content if the SC is not active. (2) Agent Ag_2 creates a SC whose content is φ. Therefore, Ag_2 becomes committed towards the content φ.

S30. M, Pa, $s_i \models$
Challenge-content(Ag_2, SC(Id_0, Ag_1, Ag_2, φ)) iff
$\exists \alpha \in \Phi a, \exists \varphi' \in$ £/£sc
& M, Pa, $s_i \models$ Perform(α)PC(Id_1, Ag_2, Ag_1, ?φ)
$\wedge$ Active(SC(Id_0, Ag_1, Ag_2, φ))
$\wedge$ EX^+F^+Justify-content(Ag_1, SC(Id_0, Ag_1, Ag_2, φ), φ')

This formula indicates that the challenge of the SC content φ by an agent Ag_2 is satisfied in the model *M* along a path *Pa* iff: (1) Agent Ag_2 commits that $?\varphi$. Indeed, $PC(Id_1, Ag_2, Ag_1, ?\varphi)$ states that "Ag_2 does not know φ but it would like to know it". (2) The challenged commitment is active on this path. (3) Agent Ag_1 justifies in the future its SC for φ. Indeed, when we challenge a statement, we expect an answer from the speaker. Thus, in our semantics the fact that there is a possibility of having an answer is included in the meaning of the challenge. The operator *E* in (EX^+F^+*Justify-content*(Ag_1, SC(Id_0, Ag_1, Ag_2, φ), φ')) allows us to capture the concept of possibility i.e. that there is a path along which Ag_1 will justify its SC. This formula highlights the fact that the challenge of a SC content is an action in itself. As for the creation operation, the action α corresponds to the production of the utterance that challenges the SC content.

3.2.7. Argumentation Relations.

S31. M, Pa, $s_i \models$
Justify-cont(Ag_1, SC(Id_0, Ag_1, Ag_2, φ), φ') iff
M, Pa, $s_i \models$ Active(SC(Id_0, Ag_1, Ag_2, φ))
$\wedge$ Create(Ag_1, SC(Id_1, Ag_1, Ag_2, $\varphi' \therefore \varphi$)).

This formula indicates that the justification of the SC content φ by an agent Ag_1 is satisfied in the model *M* on a path *Pa* iff: (1) This SC is active on this path. (2) This agent creates on this path a SC whose content is φ' that supports the conclusion φ. In other words, an agent's SC towards another agent to make a content φ true is justified (by means of φ') iff the SC exists (has been created) and moreover a SC is created to establish an argument (φ', φ), where φ' is committed to be true because according to the definition of the connector ($\therefore$), φ' is true for Ag_1. The fact that this operator is included in the SC indicates that the agent is committed that φ' is true and then φ is true, i.e. φ is true because φ' is true. Indeed, agents have knowledge bases and the propositions that are not challenged can be used for justification (i.e. as supports of arguments). Hence, to end the chain of argumentation, agents use PCs that are not challenged any further. The justification operation is the basis of other argumentation operations. As shown by the following properties (S33 and S34), this is due to the fact that all the other operations are defined using this operation.

S32. M, Pa, $s_i \models$
Contradict-content(Ag_1, SC(Id_0, Ag_1, Ag_2, φ)) iff
($\exists \varphi' \in$ £/£sc: (M, Pa, $s_i \models$ Active(SC(Id_0, Ag_2, Ag_1, φ))
$\wedge$ Create(Ag_1, SC(Id_1, Ag_1, Ag_2, φ'))) $\wedge$ ($\varphi' \therefore \neg\varphi$))

This formula indicates that an agent contradicts its previous SC whose content is φ if it creates another SC whose content is a logical conclusion of $\neg\varphi$, whereas its SC for φ is still active.

Properties:

S33. $M, Pa, s_i \models$
$\text{Attack-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi')$ iff
$M, Pa, s_i \models \text{Active}(SC(Id_0, Ag_1, Ag_2, \varphi))$
$\wedge \text{Justify-content}(Ag_2, SC(Id_1, Ag_2, Ag_1, \neg\varphi), \varphi')$

This formula indicates that the attack of the SC content φ by an agent Ag_2 is satisfied in the model M along a path Pa iff: (1) This SC is active on this path. (2) This agent justifies along this path its SC whose content is $\neg\varphi$.

S34. $M, Pa, s_i \models$
$\text{Defend-content}(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi')$ iff
$\exists\varphi'' \in £/£sc$ & $M, Pa, s_i \models \text{Active}(SC(Id_0, Ag_1, Ag_2, \varphi))$
$\wedge X^{-}F^{-}\text{Attack-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi''))$
$\wedge \text{Attack-content}(Ag_1, SC(Id_1, Ag_2, Ag_1, \varphi''), \varphi'))$

This formula indicates that the defense of the SC content φ by an agent Ag_1 is satisfied in the model M along a path Pa iff: (1) This SC is active on this path. (2) This agent attacks the attacker of the content of its SC.

3.2.8. Link between Commitments and Arguments. Until now we gave the semantics of the main elements of our formalism. We can now formally establish the link between SCs and arguments. This link is shown by the two following formulas:

S35. $A(\text{Create}(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi)) \Rightarrow$
$((\neg(F^{+}\text{Contradict-content}(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi))))$
$\wedge(F^{+}(\text{Challenge-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi)) \Rightarrow \exists\varphi':$
$AX^{+}F^{+}\text{Justify-content}(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi')))$
$\wedge(F^{+}\text{Attack-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi') \Rightarrow \exists\varphi'':$
$AX^{+}F^{+}\text{Defend-content}(Ag_1, SC(Id_0, Ag_1, Ag_2, \varphi), \varphi''))))$

This formula provides the conditions generated by the creation of a SC on all paths. The agent must be in a position to check these conditions before creating a SC. Indeed, if an agent creates a SC, then it should not contradict itself during the conversation. It must also be able to justify its SC if it is challenged and to defend it if it is attacked. By establishing the link between SCs and arguments, this formula reflects the deontic aspect of SCs. These conditions are also valid for withdrawal, acceptance and refusal because their semantics is expressed in terms of the creation operation. On the other hand, an agent challenges a SC content if it has no argument for or against this content. Therefore, An agent challenges a SC content if it cannot accept or refuse it. Formally:

S36. $A((\text{Active}(SC(Id_0, Ag_1, Ag_2, \varphi))$
$\wedge \neg\text{Accept-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi))$
$\wedge \neg\text{Refuse-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi)))$
$\Rightarrow \text{Challenge-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi)))$

4. Related Work

Semantical considerations for AgC have recently begun to find a significant audience in the MAS community. We can distinguish three kinds of semantics:

1- Mentalistic semantics: This subjective semantics is based on so-called agent's mental states. The best-known formalisms describing it are [6, 13, 16, 19]. KQML [10] and FIPA-ACL use this semantics to define a pre/post conditions semantic of communication acts. The advantage of this semantics is its compatibility with the formalisms used for reasoning about rational agents. However, the verification of such a semantics is not possible if we cannot have access to the agents' programs. In addition, this pre/post condition semantics offers no dynamic or operational description of AgC. Because our approach is based on public and argumentative concepts, the compliance verification can be made without having access to the agents' programs. The satisfaction and the violation of agents' SCs make it possible to determine if the agent respects our semantics. In addition, the agents' ability to justify their SCs facilitates this verification. In addition, our semantics treats more explicitly the dynamics aspect of AgC using the agents' actions on SCs and on their contents.

2- Social semantics: This objective semantics was proposed by Singh [20, 21] as an alternative to the mentalistic one. Singh used CTL to propose a formal language and a model in which the notion of SC is described. Verdicchio and Colombetti [22] proposed an interesting logical model of SCs by extending CTL*. This model is based on the fact that AgC should be analyzed in terms of communicative acts. Mallya et al. [14] used the temporal commitment structure specified by [11] to define some constraints in order to capture some operations on SCs. Our logical model uses some ideas of [22] and it belongs to this kind of semantics, but it differs from these propositions in the following respects: 1) In our approach the SC semantics is not defined as an abstract accessibility relation, but as an accessibility relation that takes into account the satisfaction of the SC. The semantics is defined in terms of the deadline at which the SC must be satisfied. This way is more intuitive than the semantics defined by Singh. 2) We differentiate SCs as static structures evaluated in states from the operations applied to SCs as dynamic structures evaluated on paths. This enables us to describe more naturally the evolution of the AgC as a system of states / transitions which reflects the interaction dynamics. 3) In our model, the strength of SCs as a basic principle of AgC does not result only from the fact that they are observable, but also from the fact that they are supported by arguments. The SC notion we formalize is not only a public notion but also a deontic one. The deontic aspect is captured by the fact that SCs are considered as obligations. The agent is obliged to satisfy its SCs, to behave in accordance with these SCs and to justify them. It is also obliged not to contradict its SC contents during the conversation. 4) We capture in our semantics not only PCs, but the various types of SCs. This enables us to have a greater expressivity and to capture the different types of SAs.

3- Argumentation-based semantics: This semantics is defined in [1] to capture the meaning of certain communication acts. It is based upon an argumentation system and on the formal dialectics. This semantics has the advantages of being simple and of taking into account the argumentation aspect of AgC. In addition to the fact that this semantics does not take into account temporal and dynamic aspects in its formalization, it is different from our approach on several points: 1) It is based on an informal logic. 2) It is described in terms of pre/post conditions and it does not offer the meaning of the different communication acts. 3) The commitment notion used in this semantics captures only the propositions stated by the agents. 4) Contrary to our approach, the satisfaction, violation, cancellation, attack and defense notions do not appear.

5. Conclusion and Future Work

In this paper we developed a formal semantics for our approach based on SCs and arguments to model agents' interactions. We proposed a logical model based on a combination of CTL* and DL. The model captures different SC types, different actions applied to these SCs and various argumentation relations. Our CAN formalism includes both pragmatic and semantic issues of AgC.

We plan as future work to define flexible protocols based on dialogue games and specified by our semantics. These protocols must respect some properties we are formalizing, such as "a SC remains withdrawn until a new reactivation". We also intend to automatically verify whether these protocols satisfy the semantic properties. The idea we are investigating is to use model checking techniques. Model checking allows us to verify whether a formula φ is satisfied in a model M (i.e. if $M \models \varphi$?). In our case the model M is a protocol and the formula φ is a property that the protocol must satisfies.

References

[1] Amgoud, L., Maudet, N., and Parsons, N., "An argumentation-based semantics for agent communication languages", *15th Euro. Conf. on AI*, France, 2002, pp. 38-42.

[2] 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*, Springer Verlag, 2003, pp. 146-165.

[3] Bentahar, J., Moulin, B., and Chaib-draa, B., "Commitment and Argument Network: a formal framework for representing conversation dynamics", *Logic and Dialogue*, Netherlands, Kluwer. 2003 (to appear).

[4] Castelfranchi, C., "Commitments: from individual intentions to groups and organizations", *The Internat. Conf. ICMAS*, USA, 1995, pp. 41-48.

[5] Chaib-draa, B., and Dignum, F., "Trends in agent communication language", *Computational Intelligence,* 18(2), Oxford, USA, 2002, pp. 89-101.

[6] Cohen, P.R., and Levesque, H.J., "Persistence, intentions and commitment", Cohen, P.R., Morgan. J., and Pollack, M.E., *Intentions in Communication*, Cambridge, 1990, pp. 33-70.

[7] Dastani, M., Hulstijn, J., and der Torre, L.V., "Negotiation protocols and dialogue games", *The Belgium/Dutch AI Conf.*, Kaatsheuvel, 2000, pp. 13-20.

[8] Elvang-Goransson, M., Fox, J., and Krause, P., "Dialectic reasoning with inconsistent information", *The 9th Conf. on Uncertainty in AI*, USA, 1993, pp. 114-121.

[9] Emerson, E.A., "Temporal and Modal logic", *Handbook of Theoretical Computer Science*, van Leeuwen, J. (eds), B., Elsevier Science Publishers, 1990, PP. 995-1072.

[10] Finin, T., Labrou, Y., and Mayfield, J., "KQML as an agent communication language", Bradshaw, J.M., *Software Agent*, MIT Press, Cambridge, 1997, pp. 291-316.

[11] Fornara, N., and Colombetti, M., "Operational specification of a commitment-based agent communication language", *The First Internat. Conf. on AAMAS*, Italy, 2002, pp. 536- 542.

[12] Harel, D. "Dynamic logic", *Handbook of Philosophical Logic*, Gabbay, D.M., and Guenther, F. (eds), (2), the Netherlands, 1984, pp. 497-604.

[13] van der Hoek, W., and Wooldridge, M., "Towards a logic of rational agency", *Logic Journal of the IGPL*, 11(2), 2003, pp. 133-157.

[14] Mallya, A.U., Yolum, P., and Singh, M., "Resolving Commitments Among Autonomous Agents", Dignum, F. (ed.), *Advances in Agent Communication*, Springer Verlag, 2003, pp. 166-182.

[15] McBurney, P., Parsons, S., and Wooldridge, M., "Desiderata for agent argumentation protocols", *The First Internat. Conf. on AA MAS*, Italy, 2002, pp. 402-409.

[16] Meyer, J.-J. Ch., van der Hoek, W., and van Linder, B., "A logical Approach to the dynamics of commitments", *AI Journal*, 113 (1-2), 1999, pp. 1-40.

[17] Parsons, S., Sierra, C., and Jennings, N., "Agents that reason and negotiate by arguing", *Journal of Logic and Computation*, 8(3), 1998, pp. 261-292.

[18] Pitt, J., and Mamdani, A., "Communication protocols in multi-agent systems: a development method and reference architecture", Dignum, F., and Greaves, M., *Issues in Agent Communication*, Springer Verlag, 2000, pp. 160-177.

[19] Rao, A.S., and Georgeff, M.P., "BDI agents: from theory to practice", *The Internat. Conf. ICMAS*, USA, 1995, pp.312-319.

[20] Singh, M.P., "Agent communication languages: rethinking the principles", *IEEE Computer*, 31(12), 1998, pp. 40-47.

[21] Singh, M.P., "A social semantics for agent communication language", Dignum, F., and Greaves. M. (eds.), *Issues in Agent Communication*, Springer Verlag, 2000, pp. 31-45.

[22] Verdicchio, M., and Colombetti, M., "A logical model of social commitment for agent communication", *The Second Internat. Conf. on AA MAS*, Australia, 2003, pp. 528-535.