Verifiable Semantic Model for Agent Interactions Using Social Commitments

Mohamed El-Menshawy[1], Jamal Bentahar[2], and Rachida Dssouli[2]

[1] Concordia University, Department of Electrical and Computer Engineering, Canada
m_elme@encs.concordia.ca
[2] Concordia University, Concordia Institute for Information Systems Eng., Canada
{bentahar,dssouli}@ciise.concordia.ca

Abstract. Existing approaches about defining formal semantics of commitment usually consider operations as axioms or constrains on top of the commitment semantics, which fail to capture the meaning of interactions that are central to real-life business scenarios. Furthermore, existing semantic frameworks using different logics do not gather the full semantics of commitment operations and semantics of social commitments within the same framework. This paper develops a novel unified semantic model for social commitments and their operations. It proposes a logical model based on a new logic extending CTL^* with commitments and operations to specify agent interactions. We also propose a new definition of assignment and delegation operations by considering the relationship between the original and new commitment contents. We prove that the proposed model satisfies some properties that are desirable when modeling agent interactions in MASs and introduce a NetBill protocol as a running example to clarify the automatic verification of this model. Finally, we present an implementation and report on experimental results of this protocol using the NuSMV and MCMAS symbolic model checkers.

Keywords: Social commitments, two and three party operations, accessibility relations, the NuSMV and MCMAS model checkers.

1 Introduction

The importance of defining suitable and formal semantics of social commitments has been broadly recognized for multi-agent systems (MASs). Specifically, social commitments have been used for agent communication [22,24], artificial institutions [15] and business modeling [11,25]. In commitment protocols, the commitments capture a high-level meaning of messages that are exchanged between interacting agents as opposed to low-level operational representations (see for example [30,18,11,14,7]). These protocols are more suitable for agent communication than traditional protocols specified using for example *Finite State Machines* and *Petri Nets*, which only capture legal orderings of these messages. Particularly, expressing protocols using commitments allows these protocols to be flexible and intuitive. This is because agents will not reason about legal sequences, which are generally rigid and abstract, but about concrete commitment states and possible paths to reach them. In this context, some interesting

M. Dastani et al. (Eds.): LADS 2009, LNAI 6039, pp. 128–152, 2010.

semantic frameworks for social commitments have already been proposed using different approaches such as branching time logics (CTL^*, CTL and $CTL^{\pm}$) [1,2,17,22,29]. Recently, a model-theoretic semantics of social and dialogical commitments based on Linear-Time Logic (LTL) has been introduced in [23] and the proposed postulates are reproduced in [8] to define semantics of commitment operations in distributed systems. Current semantic frameworks for social commitments and associated operations fall into two main categories. In the former category, commitment operations are formalized based on Singh's presentation [21] as axioms or constraints on top of commitment semantics [7,11,17,18,30]. These axioms are represented either as reasoning rules, updating rules or enforcing rules to evolve the truth of commitments' states and to reason about commitment operations explicitly to accommodate exceptions that may arise at run-time. However, the real meanings of commitment operations themselves (e.g. *Create*, *Fulfill*, etc.) are not captured. This makes such frameworks not general enough to capture interoperability between heterogenous systems. In the latter category, social commitments are formalized using object-oriented paradigm to advance the idea of commitments as data structure [14]. Thus, the main objective of defining clear, practical, and verifiable semantics of commitments and associated operations within the same framework for interacting autonomous agents is yet to be reached.

Motivation. This paper addresses the above challenges by proposing a new semantics not only for social commitments, but also for the operations used to manipulate commitments. This semantics can be declaratively used to specify commitment protocols and some desirable properties so that the agents can interact successfully. For automatic verification of these protocols, the semantics of commitment operations should not be only captured by some enforced rules like in [17], but also integrated in the same framework [19]. In fact, this work is a continuation of our two previous publications [2,19]. In the former one [2], we have developed a framework unifying commitments, actions on commitments and arguments that agents use to support their actions. In the second one [19], we have proposed a new logical semantics of social commitments and associated two-party operations based on Branching Space-Time (BST) logic. BST-logic enhances this semantics with agent life cycle, space-like dimension and causal relation between interacting agents in the same (physical or virtual) space. Specifically, here (1) we refine the semantics of some operations (e.g., *Create*, *Withdraw*, *Fulfill*) to overcome the *state explosion* problem that arises in [2]; (2) reformulate the life cycle of commitment introduced in [19]; and (3) define a new semantics of multi-party operations (e.g., *Delegate* and *Assign*) using a new logic that extends CTL^*.

Contributions. The contributions of this paper are manifold: (1) a novel unified logical model for social commitments and associated operations; (2) a new semantics of creation, withdrawal, fulfilment, violation and release operations using the notions of accessible and non-accessible paths; and (3) new definitions of assignment and delegation operations by taking into account the fact that

the assigned and delegated commitment's deadline could be different from the deadline of the original commitment. The proposed logical model can be used to flexibly and rigorously specify multi-agent commitment protocols and give semantics to protocol messages in terms of creation and manipulation of the commitments among interacting agents. Furthermore, compared to existing semantic frameworks for commitments, this logical model has high expressiveness because the contents of commitments are CTL^*-like path formulae [12] rather than only state formulae, and their semantics is expressed not in terms of rigid deadlines, but in terms of accessible paths. In addition to providing flexibility and expressiveness, the proposed model is based on a Kripke-like structure and accessibility relations for commitments and associated operations, which makes our semantics computationally grounded [27]. By doing this, the paper addresses the automatic verification of the proposed model with respect to the NetBill protocol against some given properties, which we capture in our semantics as they are desirable when modeling interaction protocols in MASs. This verification is done using symbolic model checking along with implementations with the MCMAS [16] and NuSMV [5] model checkers. In fact, the checked properties are compatible with the protocol properties introduced in [31] to model commitment protocols at design time.

Paper Overview. The remainder of this paper is organized as follows. Section 2 describes the notion of social commitment and its formal notation extended from [19]. Given this context, Section 3 presents the syntax and semantics of the main elements of our logical model. In Section 4, we proceed to discuss some temporal properties based on the defined semantics and Sections 5 introduces an implementation of NetBill protocol using the NuSMV and MCMAS model checkers for verifying agent interactions. The paper ends in Section 6 with a discussion of relevant literature and future work directions in Section 7.

2 Social Commitments

A commitment is an engagement in the form of *contract* made by one agent, the *debtor*, and directed towards another agent, the *creditor*, so that some fact, which is the content of the commitment, is true. The debtor must respect and behave in accordance with his commitments. These commitments are contextual, manipulable and possibly conditional [21]. Furthermore, commitments are social and observable by all the participants. Consequently, social commitments (SC) are different from the concepts of agent's private mental states such as *beliefs*, *desires* and *intentions*. Several approaches assume that agents will respect their commitments. However, this assumption is not always guaranteed, especially in real-life business scenarios where a violation can occur if agents are malicious, deceptive, malfunctioning or unreliable. It is crucial to introduce *violation* operation of social commitments along with their satisfaction and to use model checking to automatically verify agents' behaviors regarding to these operations. In the following, we distinguish between two types of social commitments: *unconditional* and *conditional*.

Definition 1. *Unconditional social commitments are related to the state of the world and denoted by* $SC^p(Ag_1, Ag_2, \phi)$ *where* Ag_1 *is the debtor,* Ag_2 *is the creditor and* ϕ *is a well-formed formula (expressed in some logics) representing the commitment content.*

The basic idea is that Ag_1 is committed towards Ag_2 that the propositional formula ϕ is true. In some cases, for example in business processes, we need to introduce an *identifier* for commitment to distinguish it from other commitments. As in [8], we use time-stamps as identifiers, when they are needed, within the propositional content of commitments. For example, to ensure that a different copy of some good items is delivered to each customer so that each customer will pay for her purchase to the merchant, this will be formally represented as follows: $SC^p(customer, merchant, pay(id))$ where id is the identifer for recognizing the payment. Such identifiers may be used to reason and track dependencies for commitments when their circumstances might be changed as in [13]. In other situations, agents can only commit about some facts when some conditions are satisfied. Conditional commitments are introduced to capture this issue.

Definition 2. *Conditional social commitments are denoted by* $SC^c(Ag_1, Ag_2, \tau, \phi)$ *where* Ag_1, Ag_2, *and* ϕ *have the same meanings as in Definition 1 and* τ *is a well-formed formula representing the condition.*

As for unconditional commitment, we can use identifiers to capture changing in the condition of commitments e.g., $SC^c(customer, merchant, pay(id), deliver\text{-}item(id))$ means that if the customer pays the merchant for a given item this item will be deliver to the customer.

2.1 Social Commitment Life Cycle

Having explained the formal definitions of commitments, in this section we present their life cycle to specify the relationship between commitment's states. The UML state diagram of this life cycle proceeds as follows: the commitment could be *Conditional* or *Unconditional*. This is represented by the selection operator (see Figure 1). The first operation an agent can perform on a commitment is *creation*. When created, a conditional commitment can move either to *Negotiation* state to negotiate the condition of commitment among the participants or to *Condition* state to check the satisfaction of the condition. The conditional commitment could be negotiated many times until reaching either a mutual agreement about the condition, meaning that the negotiation's outcome is a conditional commitment where the condition is negotiated and agreed upon among the participants, or no agrement can be reached about the condition where the commitment moves to the *Final* state. If the condition is not satisfied, the commitment also moves to the *Final* state. When the unconditional commitment is created, then it may either move to the one of the following states: *Fulfilled*, *Violated*, *Withdrawn*, *Released*, *Delegated*, *Assigned* or move to *Negotiation* state again but here to negotiate the commitment content. When the participant agents reach an agreement, then the commitment moves to the one

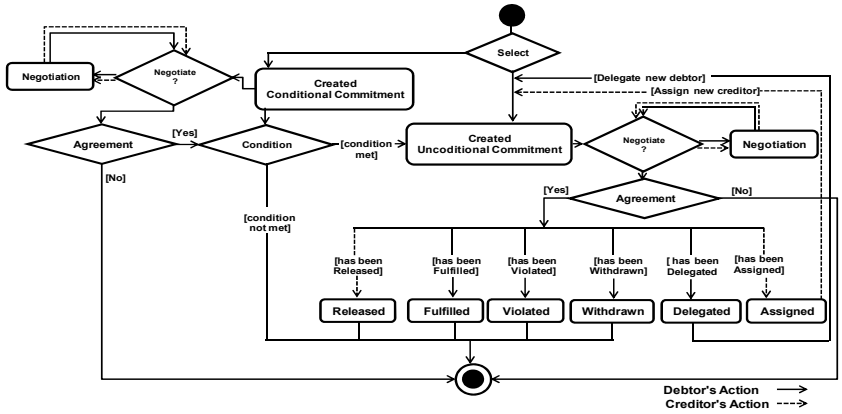

Fig. 1. Life cycle of social commitment

of the aforementioned states or to the *Final* state if no agreement is reached. In this paper, we consider all the operations except negotiation, which needs other techniques such as game theory or dialogue games that are beyond the scope of this paper. When the debtor performs the withdrawal action (within a specified deadline), the commitment is withdrawn. Specifically, only the debtor is able to perform this action without any intervention from the creditor. The commitment is fulfilled if its content is satisfied by the debtor within the deadline. The social commitment is violated if its content is violated by the debtor when the deadline is over. The social commitment can be released by the creditor so that the debtor is no longer obliged to carry out his commitment. The social commitment can be assigned by the creditor, which results in releasing this creditor from the commitment and having a new unconditional commitment with a new creditor. The social commitment can be delegated by the debtor, which results in withdrawing this debtor from the commitment and delegating his role to another debtor within a new commitment. In our approach, some operations such as delegation and assignment can be applied on a commitment multiple times e.g., the delegated commitment can be delegated again or move to another state such as fulfillment and so on. Only release, fulfillment, violation, and withdrawal can be applied one time.

3 The Logical Model of Social Commitments

This section introduces the syntax and semantics of the different elements of our formal language $\mathcal{L}$. This propositional language uses extended Computation Tree Logic (CTL^*) [12] with past operators extended by two new modalities SC^p for unconditional and SC^c for conditional commitments and actions applied to commitments. We refer to the resulted branching time logic as CTL^{*sc}. Conventionally, a model of CTL^* is a tree whose nodes correspond to the states of the system being considered. In our logic, the branches or paths of the tree represent

all choices in the future that agents have when they participate in conversations, while the past is linear. The time is discrete and the dynamic behavior of agents is captured by actions these agents perform on their commitments.

3.1 The Syntax of CTL^{*sc}

Let Φ_p be a set of atomic propositions, AGT a set of agent names and ACT a set of actions used to manipulate commitments. We also use the following conventions: Ag, Ag_1, Ag_2, Ag_3, etc. are agent names in AGT, p, p_1, p_2, etc. are atomic propositions in Φ_p, $\alpha, \alpha_1, \alpha_2$, etc. are actions performed by agents in ACT and ϕ, ψ, etc. are formulae in $\mathcal{L}$. Table 1 gives the formal syntax of $\mathcal{L}$ expressed in Backus-Naur Form (BNF) grammar where "::=" and "|" are meta-symbols of this grammar. The intuitive meanings of the most constructs are straightforward (from CTL^* with next (X^+), previous (X^-), until (U^+), and since (U^-) operators). A and E are the universal and existential path-quantifiers over the set of all paths starting from the current moment. $A\phi$ (respectively $E\phi$) means that ϕ holds along all (some) paths starting at the current moment. Notice that, the content and condition of commitment are path formulae, which are more expressive than state formulae. Furthermore, there are some useful abbreviations based on temporal operators (X^+, X^-, U^+, U^-): (sometimes in the future) $F^+\phi \triangleq true\ U^+\phi$; (sometimes in the past) $F^-\phi \triangleq true\ U^-\phi$; (globally in the future) $G^+\phi \triangleq \neg F^+\neg\phi$ and (globally in the past) $G^-\phi \triangleq \neg F^-\neg\phi$. We also introduce $\mathcal{L}^- \subset \mathcal{L}$ as the subset of all formulae without temporal operators.

Table 1. The Syntax of CTL^{*sc} Logic

$$
\begin{array}{l}
S ::= p \mid \neg S \mid S \vee S \mid S \wedge S \mid A\mathcal{P} \mid E\mathcal{P} \mid C \\
\mathcal{P} ::= S \mid \alpha \mid \mathcal{P} \vee \mathcal{P} \mid X^+\mathcal{P} \mid X^-\mathcal{P} \mid \mathcal{P}\ U^+\ \mathcal{P} \mid \mathcal{P}\ U^-\ \mathcal{P} \\
C ::= SC^p(Ag_1, Ag_2, \mathcal{P}) \mid SC^c(Ag_1, Ag_2, \mathcal{P}, \mathcal{P}) \\
\alpha ::= Create(Ag_1, C) \mid Fulfill(Ag_1, C) \mid Violate(Ag_1, C) \mid Withdraw(Ag_1, C) \\
\qquad \mid Release(Ag_2, C) \mid Assign(Ag_2, Ag_3, C) \mid Delegate(Ag_1, Ag_3, C)
\end{array}
$$

3.2 The Semantics of CTL^{*sc}

In this section, we define the formal model, which corresponds to the computational model representing the protocol the agents use to communicate in which the actions reflect the protocol's progress. Thereafter, we explain how a real system, namely NetBill protocol is represented based on the proposed model.

The Formal Model

Our formal model M for $\mathcal{L}$, which is used to interpret the formulae of CTL^{*sc}, is based on a Kripke-like structure and defined as follows: $M = \langle \mathbb{S}, \mathtt{ACT}, \mathtt{AGT}, \mathbb{T}, R, \mathbb{V}, \mathbb{R}_{scp}, \mathbb{R}_{scc}, \mathbb{F} \rangle$, where: $\mathbb{S} = \{s_0, s_1, s_2, \ldots\}$ is a finite set of states; ACT and AGT are defined in Section 3.1; $\mathbb{T} : \mathbb{S} \rightarrow$ TP is a function assigning to each state the

corresponding time-stamp from $\mathtt{TP}$; $R \subseteq \mathbb{S} \times \mathtt{ACT} \times \mathbb{S}$ is a total transition relation, that is, $\forall s_i \in \mathbb{S}$ and $\alpha_{i+1} \in \mathtt{ACT}$, $\exists s_{i+1} \in \mathbb{S} : (s_i, \alpha_{i+1}, s_{i+1}) \in R$, indicating labelled transitions between states in branching time in which if there exists a transition $(s_i, \alpha_{i+1}, s_{i+1}) \in R$, then we have $T(s_i) < T(s_{i+1})$; $\mathbb{V} : \Phi_p \rightarrow 2^{\mathbb{S}}$ is a valuation function that assigns to each atomic proposition a set of states where the proposition is true; $\mathbb{R}_{scp} : \mathbb{S} \times \mathtt{AGT} \times \mathtt{AGT} \rightarrow 2^{\sigma}$, where σ is the set of all paths, is a function producing an accessibility modal relation for unconditional commitments; $\mathbb{R}_{scc} : \mathbb{S} \times \mathtt{AGT} \times \mathtt{AGT} \rightarrow 2^{\sigma}$ is a function producing an accessibility modal relation for conditional commitments; and $\mathbb{F} : \mathcal{L} \rightarrow \mathcal{L}^{-}$ is a function associating to each formula in $\mathcal{L}$ a corresponding formula in $\mathcal{L}^{-}$.

A path P_i is an infinite sequence of states starting at s_i. Through this paper we use a ternary relation $(s_i, \alpha_{i+1}, s_{i+1})$ interchangeably with infix notation $s_i \xrightarrow{\alpha_{i+1}} s_{i+1}$ for temporal transition relations. Thus, the paths that path formulae are interpreted over have the form $P_i = s_i \xrightarrow{\alpha_{i+1}} s_{i+1} \xrightarrow{\alpha_{i+2}} s_{i+2} \ldots$ with $i \geq 0$. The set of all paths starting at state s_i is denoted by σ^{s_i}. When there is no need to show the actions, the paths will be represented as follows: $P_i = \langle s_i, s_{i+1}, \ldots \rangle$.

The function $\mathbb{R}_{scp}$ associates to a state s_i the set of paths starting at s_i along which an agent commits towards another agent. Such paths are conceived as merely "possible", and as paths where the commitments contents made in s_i are true. The computational interpretation of this accessibility relation is as follows: the paths over the model M are seen as computations, and the accessible paths from a state s_i are the computations satisfying (i.e. computing) the formulae representing the contents of commitments made at that state by a given agent towards another given agent. For example, if we have: $P'_i \in \mathbb{R}_{scp}(s_i, Ag_1, Ag_2)$, then the commitments that are made in the state s_i by Ag_1 towards Ag_2 about ϕ are satisfied along the path $P'_i \in \sigma^{s_i}$. $\mathbb{R}_{scc}$ is similar to $\mathbb{R}_{scp}$ and it gives us the paths along which the resulting unconditional commitment is satisfied if the underlying condition is true. Because it is possible to decide if a path satisfies a formula (see the semantics in this section), the model presented here is computationally grounded [27]. In fact, the accessible relations map commitment content formulae into a set of paths that simulate the behavior of interacting agents. The accessibility modal relations $\mathbb{R}_{scp}$ and $\mathbb{R}_{scc}$ are serials [2] and the logic of unconditional and conditional commitments is an KD4 modal logic. The function $\mathbb{F}$ is used to remove the temporal operators from a formula in $\mathcal{L}$. For example: $\mathbb{F}(X^{+}X^{+}p) = p$ and $\mathbb{F}(SC^{p}(Ag_1, Ag_2, X^{+}p_1)) = SC^{p}(Ag_1, Ag_2, p_1)$.

The Semantics of Temporal Operators and Commitments

Having explained our formal model, in this section we define the semantics of the elements of $\mathcal{L}$ relative to a model M, state s_i and path P_i. The notation $\langle s_i, P_i \rangle$ refers to the path P_i starting at s_i (i.e., $P_i \in \sigma^{s_i}$). If P_i is a path starting at a given state s_i, then $\mathtt{prefix}$ of P_i starting at a state s_j $(\mathbb{T}(s_j) < \mathbb{T}(s_i))$ is a path denoted by $P_i \downarrow s_j$ and $\mathtt{suffix}$ of P_i starting at a state s_k $(\mathbb{T}(s_i) < \mathbb{T}(s_k))$ is a path denoted by $P_i \uparrow s_k$. Because the past is linear, s_j is simply a state in the unique past of s_i such that P_i is a part of $P_i \downarrow s_j$. The state s_k is in the future of s_i over the path P_i such that $P_i \uparrow s_k$ is part of P_i. Using prefix and suffix notions, the following holds: $P_i = P_i \uparrow s_i = P_i \downarrow s_i$. If s_i is a state, then

we assume that s_{i-1} is the previous state in the linear past and s_{i+1} is the next state on a given path.

To define the semantics of the formulae in the object language $\mathcal{L}$, we use the following meta-symbols: & means "and", and $\Rightarrow$ means "implies that". The logical equivalence is denoted $\equiv$. As in CTL^*, we have two types of formulae: state formulae evaluated over states and path formulae evaluated over paths [12]. $M, \langle s_i \rangle \models \phi$ means "the model M satisfies the state formula ϕ at s_i". $M, \langle s_i, P_i \rangle \models \phi$ means "the model M satisfies the path formula ϕ along the path P_i starting at s_i". A state formula ϕ is satisfiable iff there are some M and s_i such that $M, \langle s_i \rangle \models \phi$. A path formula ϕ is satisfiable iff there are some M, P_i and s_i such that $M, \langle s_i, P_i \rangle \models \phi$. A state formula is valid when it is satisfied in all models M, in all states s_i in M. A path formula is valid when it is satisfied in all models M, in all paths P_i in M, in all states s_i. A path satisfies a state formula if the initial state in the path does so (i.e., $M, \langle s_i, P_i \rangle \models \phi$ iff $M, \langle s_i \rangle \models \phi$). The formal semantics of CTL^* and SC^p and SC^c is illustrated in Table 2.

Table 2. Semantics of CTL^* and SC^p and SC^c modalities

M1.	$M, \langle s_i \rangle \models p$	iff	$s_i \in \mathbb{V}(p)$ where $p \in \Phi_p$
M2.	$M, \langle s_i \rangle \models \neg\phi$	iff	$M, \langle s_i \rangle \not\models \phi$
M3.	$M, \langle s_i \rangle \models \phi \vee \psi$	iff	$M, \langle s_i \rangle \models \phi$ or $M, \langle s_i \rangle \models \psi$
M4.	$M, \langle s_i \rangle \models \phi \wedge \psi$	iff	$M, \langle s_i \rangle \models \phi$ and $M, \langle s_i \rangle \models \psi$
M5.	$M, \langle s_i \rangle \models A\phi$	iff	$\forall P_i \in \sigma^{s_i} : M, \langle s_i, P_i \rangle \models \phi$
M6.	$M, \langle s_i \rangle \models E\phi$	iff	$\exists P_i \in \sigma^{s_i} : M, \langle s_i, P_i \rangle \models \phi$
M7.	$M, \langle s_i \rangle \models SC^p(Ag_1, Ag_2, \phi)$	iff	$\forall P_i \in \mathbb{R}_{scp}(s_i, Ag_1, Ag_2) : M, \langle s_i, P_i \rangle \models \phi$
M8.	$M, \langle s_i \rangle \models SC^c(Ag_1, Ag_2, \tau, \phi)$	iff	$\forall P_i \in \mathbb{R}_{scc}(s_i, Ag_1, Ag_2) : M, \langle s_i, P_i \rangle \models \tau$ $\Rightarrow M, \langle s_i, P_i \rangle \models SC^p(Ag_1, Ag_2, \phi)$
M9.	$M, \langle s_i, P_i \rangle \models \phi$	iff	$M, \langle s_i \rangle \models \phi$
M10.	$M, \langle s_i, P_i \rangle \models \neg\phi$	iff	$M, \langle s_i, P_i \rangle \not\models \phi$
M11.	$M, \langle s_i, P_i \rangle \models \phi \vee \psi$	iff	$M, \langle s_i, P_i \rangle \models \phi$ or $M, \langle s_i, P_i \rangle \models \psi$
M12.	$M, \langle s_i, P_i \rangle \models \phi \wedge \psi$	iff	$M, \langle s_i, P_i \rangle \models \phi$ and $M, \langle s_i, P_i \rangle \models \psi$
M13.	$M, \langle s_i, P_i \rangle \models X^+\phi$	iff	$M, \langle s_{i+1}, P_i \uparrow s_{i+1} \rangle \models \phi$
M14.	$M, \langle s_i, P_i \rangle \models \phi\ U^+\ \psi$	iff	$\exists j \geq i : M, \langle s_j, P_i \uparrow s_j \rangle \models \psi$ & $\forall i \leq k < j\ , M, \langle s_k, P_i \uparrow s_k \rangle \models \phi$
M15.	$M, \langle s_i, P_i \rangle \models X^-\phi$	iff	$M, \langle s_{i-1}, P_i \downarrow s_{i-1} \rangle \models \phi$
M16.	$M, \langle s_i, P_i \rangle \models \phi\ U^-\ \psi$	iff	$\exists j \leq i : M, \langle s_j, P_i \downarrow s_j \rangle \models \psi$ & $\forall j < k \leq i\ , M, \langle s_k, P_i \downarrow s_k \rangle \models \phi$

The semantics of state formulae is given from $M1$ to $M8$ and that of path formulae is given from $M9$ to $M16$. For space limit reasons, we only explain the semantics of formulae that are not in CTL^*. $M7$ gives the semantics of propositional commitment, where the state formula is satisfied in the model M at s_i iff the content ϕ is true in all accessible paths P_i starting at s_i using $\mathbb{R}_{scp}$. Similarly, $M8$ gives the semantics of conditional commitment, where the state

formula is satisfied in the model M at s_i iff in all accessible paths P_i using $\mathbb{R}_{scc}$ if the condition τ is true, then the underlying unconditional commitment is also true. The semantics of past operators X^- and U^- is given by considering the linear past of the current state s_i as prefix of the path P_i.

The Semantics of Action Formulae

Having defined the semantics of commitments, below we define the semantics of some socially relevant operations used to manipulate those commitments and to capture the dynamic behaviors of agents. Such operations are of two categorizes: two-party operations (which need only two agents to be performed): *Create, Withdraw, Fulfill, Violate* and *Release*, and three-party operations: *Assign* and *Delegate* because assign and delegate need a third agent to which the new commitment is assigned or delegated. The context and detailed exposition of these operations are given in [17,18,21]. In fact, the semantics of these operations is compatible with the philosophical interpretation of actions in which by performing an action the agent selects a path or history among the available paths or histories at the moment of performing the action.

Creation action: The semantics of creation action of a propositional commitment (see Table 3) is satisfied in the model M at state s_i along path P_i iff the commitment is established in s_{i+1} as a result of performing the creation action and the created commitment holds along the suffix $P_i \uparrow s_{i+1}$ of the path P_i.

Table 3. Semantics of creation action relative to SC^p

M17.	$M, \langle s_i, P_i\rangle \models Create(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff $(s_i, Create, s_{i+1}) \in R$ & $M, \langle s_{i+1}, P_i \uparrow s_{i+1}\rangle \models SC^p(Ag_1, Ag_2, \phi)$

The semantics of creation action of a conditional commitment (see Table 4) is defined in the same way.

Table 4. Semantics of creation action relative to SC^c

M18.	$M, \langle s_i, P_i\rangle \models Create(Ag_1, SC^c(Ag_1, Ag_2, \tau, \phi))$ iff $(s_i, Create, s_{i+1}) \in R$ & $M, \langle s_{i+1}, P_i \uparrow s_{i+1}\rangle \models SC^c(Ag_1, Ag_2, \tau, \phi)$

Example 1. Let us consider the NetBill protocol, which corresponds to the formal model M discussed at the beginning of this section. The purpose is to use this protocol as a running example to illustrate the semantics of different action formulae. As shown in Figure 2, the protocol begins with a Customer (Cus) requesting a quote for some goods (rfq) at state s_0, followed by the Merchant (Mer) sending the quote as an offer at state s_1, but the Cus agent can release after receiving the offer. Moreover, when the customer pays for the goods, then the

merchant will deliver the goods, withdraw, assign the delivery to another merchant or not deliver the requested goods. The offer message at state s_1 means that Mer creates a conditional commitment *Create (Mer,SC^c(Mer, Cus, pay, delivergoods))* meaning that if the payment is received, then Mer commits to deliver the goods to Cus. $\langle s_1, s_2, s_3, \ldots\rangle$, $\langle s_1, s_2, s_4, s_6, \ldots\rangle$ and $\langle s_1, s_2, s_4, s_8, \ldots\rangle$ are not accessible paths for this commitment (i.e. are not in $\mathbb{R}_{scc}(s_1, Mer, Cus)$. However, $\langle s_1, s_2, s_4, s_5, \ldots\rangle$ is an accessible path (i.e. is in $\mathbb{R}_{scc}(s_1, Mer, Cus)$. When the condition is true through $\langle s_1, s_2, s_4, s_5, \ldots\rangle$ (the customer pays), the conditional commitment becomes unconditional commitment: *SC^p(Mer, Cus, delivergoods)* along the same accessible path. When the Cus agent receives an offer he can negotiate the circumstances of this commitment to reach a mutual agreement. In this sense, the Mer agent needs to use different identifiers to distinguish between different offers as we explained in Section 2.

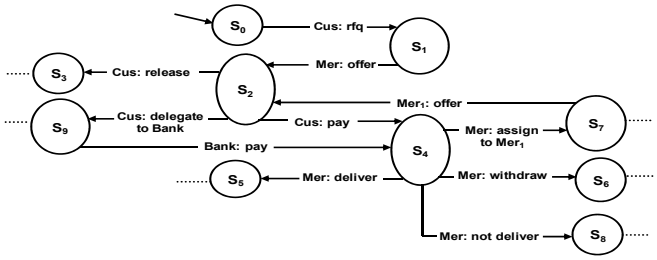

Fig. 2. Representation of NetBill Protocol

Withdrawal action: The semantics of withdrawal action of a propositional commitment (see Table 5) is satisfied in the model M at s_i along path P_i iff (i) the commitment was established in the past at s_j (after performing the creation action) through the prefix $P_i \downarrow s_j$ (ii) the withdrawal action is performed, which means $(s_i, Withdraw, s_{i+1}) \in R$ and the prefix $P_{i+1} \downarrow s_j$ is not one of the accessible paths using $\mathbb{R}_{scp}$ and (iii) at the current state s_i, there is still a possibility of satisfying the commitment since there is a path P_i' whose the prefix $P_i' \downarrow s_j$ is still accessible using $\mathbb{R}_{scp}$. Notice, furthermore, that the first argument of $\mathbb{R}_{scp}$ is s_j where the commitment has been established as in our approach the accessible paths start from the state where the commitment is established (i.e., the next state after the creation operation).

Table 5. Semantics of withdrawal action

M19.	$M, \langle s_i, P_i\rangle \models Withdraw(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff
	(i) $\exists j \leq i : M, \langle s_j, P_i \downarrow s_j\rangle \models SC^p(Ag_1, Ag_2, \phi))$ &
	(ii) $(s_i, Withdraw, s_{i+1}) \in R$ & $P_{i+1} \downarrow s_j \notin \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$ &
	(iii) $\exists P_i' \in \sigma^{s_i} : P_i' \downarrow s_j \in \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$

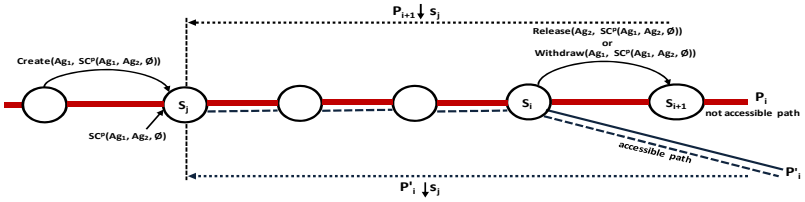

Fig. 3. Withdraw and Release actions at the state s_i along path P_i

Intuitively, when a commitment is withdrawn along a path, the prefix of this path from the established state does not correspond to an accessible path (condition ii). Furthermore, a commitment can be withdrawn when its satisfaction is still possible (condition iii), which is captured by the existence, starting at the current moment, of an accessible path the agent can choose (see Figure 3). In other words, the agent Ag_1 has another choice at the current state, which is continuing in the direction of satisfying its commitment.

Example 2. The Mer agent, before delivering the goods to the Cus agent, can withdraw the offer. Thus, there is no accessible path for the commitment between Mer and Cus at s_6. At the same time, Mer still has a possibility to satisfy its offer at state s_4 through the accessible path $\langle s_1, s_2, s_4, s_5 \ldots \rangle$ (see Figure 2).

Fulfillment action: The semantics of fulfillment action (see Table 6) is defined in the same way as withdrawal action. In (ii), all paths starting at state s_{i+1} (the state resulting from the fulfillment action) using $\mathbb{R}_{scp}$ at state s_j (where the commitment has been established) are accessible paths; and in (iii), at the current state s_i, there is still a possible choice of not satisfying the commitment since a non-accessible path $P_i'' \downarrow s_j$ starting at s_i exists. We notice that being accessible means that the content ϕ is true along all the paths $P'_{i+1} \downarrow s_j$ and fulfillment occurs automatically when the content holds by the deadline. As for withdrawal, fulfillment action makes sense only when a non-fulfilment action is still possible.

Table 6. Semantics of fulfillment action

M20.	$M, \langle s_i, P_i \rangle \models Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff
	(i) $\exists j \leq i : M, \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi)$ &
	(ii) $(s_i, Fulfill, s_{i+1}) \in R$ & $\forall P'_{i+1} \in \sigma^{s_{i+1}} : P'_{i+1} \downarrow s_j \in \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$ &
	(iii) $\exists P_i'' \in \sigma^{s_i} : P_i'' \downarrow s_j \notin \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$

Example 3. When the Cus agent pays for the goods and the Mer agent delivers the goods before the deadline expires, Mer satisfies his commitment through all the accessible paths $\langle s_1, s_2, s_4, s_5, \ldots \rangle$ for all possible continuations from s_5. At the moment of satisfying the commitment, Mer has still a possibility of not satisfying it through the non-accessible path $\langle s_1, s_2, s_4, s_8, \ldots \rangle$ (see Figure 2).

Violation action: The semantics of violation action (see Table 7) is almost similar to the semantics of withdrawal action. The main difference is related to the fact that when a commitment is violated within a specific time, then all paths P'_{i+1} starting at state s_{i+1} (after violation action has been performed), their suffix $P'_{i+1} \downarrow s_j$ are non-accessible paths using $\mathbb{R}_{scp}$ at state s_j (condition ii). Whereas, in withdrawal action there is one path which is non-accessible along which withdrawal action has been performed. As for fulfillment action, a non-accessible path means that the content of commitment ϕ is false. Here again, violation action makes sense when a choice of satisfying the commitment is still possible at the current state (condition iii).

Table 7. Semantics of violation action

M21. $M, \langle s_i, P_i \rangle \models Violate(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $\exists j \leq i : M, \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi)$ & (ii) $(s_i, Violate, s_{i+1}) \in R$ & $\forall P'_{i+1} \in \sigma^{s_{i+1}} : P'_{i+1} \downarrow s_j \notin \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$ & (iii) $\exists P''_i \in \sigma^{s_i} : P''_i \downarrow s_j \in \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$

Example 4. When the Cus agent pays for the goods, but the Mer agent does not deliver them within a specified time, then Mer violates his commitment. Through the path $\langle s_1, s_2, s_4, s_8, \ldots \rangle$ the content *Delivergoods* is false (see Figure 2).

Release action: The semantics of release action (see Table 8) is similar to the semantics of withdrawal action. The only difference is that the release action is performed by the creditor while withdrawal action is performed by the debtor (see Figure 3).

Table 8. Semantics of release action

M22. $M, \langle s_i, P_i \rangle \models Release(Ag_2, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $\exists j \leq i : M, \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi))$ & (ii) $(s_i, Release, s_{i+1}) \in R$ & $P_{i+1} \downarrow s_j \notin \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$ & (iii) $\exists P'_i \in \sigma^{s_i} : P'_i \downarrow s_j \in \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$

Example 5. The Cus agent, before paying for the goods, can release the offer. Thus, no accessible path exists between the Cus and Mer agents from s_1. However, an accessible path still exists from s_2 (see Figure 2).

Assignment action: The semantics of assignment action of a propositional commitment (see Table 9) is satisfied in the model M at s_i along path P_i iff (i) the creditor Ag_2 releases the current commitment at s_i through P_i and (ii) a new commitment with the same debtor and a new creditor is created at state s_i along path P_i, so that the formula $SC^p(Ag_1, Ag_3, \phi')$ is true in the model M at s_{i+1}. The most important issue in this semantics is that the content ϕ' of the

new commitment is not necessarily the same as for the assigned one ϕ, but there is a logical relationship between them. This is because the second commitment is established after the previous one. Thus, we need to consider the temporal component specifying the deadline of the first commitment.

Table 9. Semantics of assignment action

M23. $M, \langle s_i, P_i\rangle \models Assign(Ag_2, Ag_3, SC^p(Ag_1, Ag_2, \phi))$ iff
(i) $M, \langle s_i, P_i\rangle \models Release(Ag_2, SC^p(Ag_1, Ag_2, \phi))$ &
(ii) $\exists j \leq i : M, \langle s_j, P_i \downarrow s_j\rangle \models SC^p(Ag_1, Ag_2, \phi)$ & $(s_i, Create, s_{i+1}) \in R$ &
$M, \langle s_{i+1}\rangle \models SC^p(Ag_1, Ag_3, \phi')$ such that
(1) $\forall P'_{i+1} \in \sigma^{s_{i+1}},\ M, \langle s_j, P'_{i+1} \downarrow s_j\rangle \models \phi \Leftrightarrow M, \langle s_{i+1}, P'_{i+1}\rangle \models \phi'$ &
(2) $\mathbb{F}(\phi) \equiv \mathbb{F}(\phi')$

The logical relationship between ϕ and ϕ' is as follows: (1) ϕ' is true at the state s_{i+1} through a given path P'_{i+} iff ϕ is true at s_j where the original commitment has been established through the prefix $P'_{i+1} \downarrow s_j$ and (2) the two contents are logically equivalent when the temporal operators are removed. We consider the current state s_{i+1} in (1) as the new content ϕ' should be true starting from the moment where the new commitment is created (see Figure 4). To clarify this notion, suppose that the content of the assigned commitment is $\phi = X^+X^+p$ where p is an atomic proposition and the assignment action takes place at the next moment after the creation action. The content of the resulting commitment should be then $\phi' = X^+p$, which is the content we obtain by satisfying the conditions (1) and (2). By (1) we have X^+p is true at a state s_{i+1} through a path P'_{i+1} iff X^+X^+p is true at the state s_j ($s_j = s_{i-1}$) through $P'_{i+1} \downarrow s_j$ and by (2) we have $\mathbb{F}(X^+X^+p) \equiv \mathbb{F}(X^+p)$. The second condition is added to guarantee that the relationship between the contents is not arbitrary.

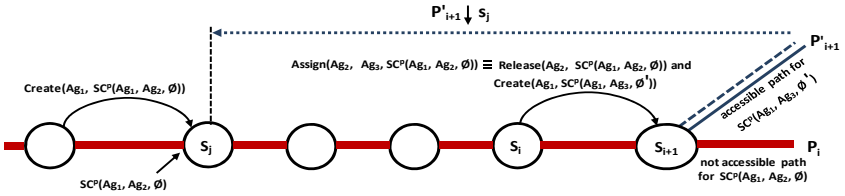

Fig. 4. Assign action at the state s_i along the path P_i

Example 6. Suppose the Cus agent commits to pay \$200 to the Mer agent in two days. After one day, Mer, for some reasons, assigns this commitment to another agent Mer_1 at s_7. We suppose that Cus and Mer_1 have reached to agreement after negotiating the conditions of this commitment, then Mer releases this commitment and a new commitment between Cus and Mer_1 is created to pay \$200 after only one day.

In the semantics proposed in previous frameworks (for example in [17,18] and [23]), the two commitments have the same content, which implicitly suppose that the creation of the commitment and its assignment take place at the same moment. The previous example cannot be managed using this assumption.

Delegation action: The semantics of delegation action (see Table 10) is similar to the semantics of assignment. The only difference is that delegation is performed by the debtor while assignment is performed by the creditor. Therefore, instead of release action, the semantics is defined in terms of withdrawal action.

Table 10. Semantics of delegation action

M24. $M, \langle s_i, P_i\rangle \models Delegate(Ag_1, Ag_3, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $M, \langle s_i, P_i\rangle \models Withdraw(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ & (ii) $\exists j \leq i : M, \langle s_j, P_i \downarrow s_j\rangle \models SC^p(Ag_1, Ag_2, \phi)$ & $(s_i, Create, s_{i+1}) \in R$ & $M, \langle s_{i+1}\rangle \models SC^p(Ag_3, Ag_2, \phi')$ such that (1) $\forall P'_{i+1} \in \sigma^{s_{i+1}},\ M, \langle s_j, P'_{i+1} \downarrow s_j\rangle \models \phi \Leftrightarrow M, \langle s_{i+1}, P'_{i+1}\rangle \models \phi'$ & (2) $\mathbb{F}(\phi) \equiv \mathbb{F}(\phi')$

Example 7. Suppose the Cus agent commits to pay \$200 to the Mer agent in two days. After one day, Cus, for some reasons, delegates this commitment to a financial company ($Bank$) to pay \$200 to Mer on his behalf. Thus, Cus withdraws his commitment and a new commitment between $Bank$ and Mer is created to pay \$200 after only one day.

4 Commitment Properties

The aim of this section is to prove that the model aforementioned in the previous section presents a satisfactory *logic of commitment*. We show some of desirable properties related to the semantics of commitment operations, which are fundamental for soundness considerations (i.e., this semantics never produces a computation that does not satisfy these properties) and for checking commitment protocols. Such properties hold when the commitments are aligned [8] among the agents. That is, the interacting agents observe all the exchanged commitments (although they use asynchronous messages to communicate) and do agree on the meanings of messages exchanged. In the rest of this paper, the set of all models is denoted by $\mathcal{M}$ and "$\rightarrow$" stands for logical implication.

Proposition 1. *Once fulfilled, a commitment cannot be fulfilled again in the future.*

$$AG^+\big[Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi)) \rightarrow X^+AG^+\neg Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi))\big]$$

Proof. Let M be a model in $\mathcal{M}$, s_i a state in $\mathbb{S}$, and P_i a path in σ. Also, suppose that: $M, \langle s_i, P_i\rangle \models Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi))$

(Semantics of fulfillment action)

$\Rightarrow \exists j \leq i : M, \langle s_j, P_i \downarrow s_j\rangle \models SC^p(Ag_1, Ag_2, \phi)$ &
$(s_i, Fufill, s_{i+1}) \in R$ & $\forall P'_{i+1} \in \sigma^{s_{i+1}} : P'_{i+1} \downarrow s_j \in \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$ (1)

Let us now suppose that:

$M, \langle s_i, P_i\rangle \models X^+EF^+\ Fufill(Ag_1, SC^p(Ag_1, Ag_2, \phi))$

(Semantics of fulfillment action, X^+ and EF^+ operators)

$\Rightarrow \exists k \geq i+1 : (s_k, Fufill, s_{k+1}) \in R$ & $\exists P'_k \in \sigma^{s_k} : P'_k \downarrow s_j \notin \mathbb{R}_{scp}(s_j, Ag_1, Ag_2)$

There is then a contradiction with (1) because P'_k is a suffix of at least one of the accessible paths P'_{i+1} starting at s_{i+1}. Consequently:

$M, \langle s_i, P_i\rangle \models \neg X^+EF^+Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi))$
So, we obtain: $M, \langle s_i, P_i\rangle \models X^+AG^+\neg Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi))$

We can generalize this proposition to the following one and prove it in the same way using the semantics of withdrawal, fulfillment, violation, release, assignment and delegation actions, X^+ and EF^+ operators.

Proposition 2. *Once fulfilled, violated, withdrawn or released a commitment cannot be fulfilled, violated, withdrawn, released, assigned or delegated again in the future.*

$$\begin{aligned}\forall Ag \in \mathtt{AGT}, AG^+\big[&Ful\text{-}Vio\text{-}Wit\text{-}Rel(Ag_1, SC^p(Ag_1, Ag_2, \phi)) \rightarrow \\ &X^+AG^+\neg\big[Fulfill(Ag_1, SC^p(Ag_1, Ag_2, \phi)) \\ &\vee Violate(Ag_1, SC^p(Ag_1, Ag_2, \phi)) \\ &\vee Withdraw(Ag_1, SC^p(Ag_1, Ag_2, \phi)) \\ &\vee Release(Ag_2, SC^p(Ag_1, Ag_2, \phi)) \\ &\vee Assign(Ag_2, Ag, SC^p(Ag_1, Ag_2, \phi)) \\ &\vee Delegate(Ag_1, Ag, SC^p(Ag_1, Ag_2, \phi))\big]\big]\end{aligned}$$

where $Ful\text{-}Vio\text{-}Wit\text{-}Rel \in \{Fulfill, Violate, Withdraw, Release\}$

In our approach, the semantics of two-party operations, such as withdrawal and fulfillment actions, is stable meaning that when each one of them has been performed, then it holds forever. While, the semantics of three-party operations is not stable. These actions have the property of "leaving the agents sensitive to race conditions over commitments" [8].

5 Verifying Agent Interactions

A rigorous semantics opens up the way for the automatic verification of logic-based protocols that govern business behaviors of autonomous agents and conformance of MASs specifications against some temporal properties. Specifically,

this section presents model checking to verify the proposed logical model with respect to a NetBill protocol. The protocol and temporal properties are specified by the proposed language $\mathcal{L}$. Finally, we present an implementation and report on experimental results of this protocol using the NuSMV and MCMAS tools.

5.1 Concrete Interpretations

Having defined, in Section 3.2, a Kripke-like structure $M = \langle \mathbb{S}, \mathtt{ACT}, \mathtt{AGT}, \mathbb{T}, R, \mathbb{V}, \mathbb{R}_{scp}, \mathbb{R}_{scc}, \mathbb{F} \rangle$ and accessibility relations that make our semantics computationally grounded, here we give a concrete interpretation of this model from the perspective of a transition system by adding a function $\mathbb{L}$ and initial state s_0 and giving a concrete (computational) interpretation of accessibility relations $\mathbb{R}_{scp}$ and $\mathbb{R}_{scc}$ using the existential operator E. Such a computational interpretation of the philosophical intuition of the accessibility relations is important for concrete model checking. The function $\mathbb{L} : \mathbb{S} \rightarrow 2^{\mathtt{AGT} \times \mathtt{AGT}}$ associates to each state a set of pairs and each pair represents the two interacting agents, which are the debtor and creditor of a commitment made in this state. Consequently, our interpreted system is defined as follows: $M' = \langle \mathbb{S}, \mathtt{ACT}, \mathtt{AGT}, \mathbb{T}, R, \mathbb{V}, \mathbb{L}, \mathbb{F}, s_0 \rangle$. By doing this, we can define the concrete interpretation of commitments and associated operations as shown in Table 11. For example, a commitment made by Ag_1 towards Ag_2 is satisfied at state s_i iff there is a path starting at this state (i.e., a possible computation) along which the commitment holds. The intuitive interpretation is as follows: when an agent Ag_1 commits towards another agent Ag_2 to bringing about ϕ at state s_i, this means that there is at least a possible computation starting at this state satisfying ϕ.

For withdraw action, a computation P_i starting at s_i satisfies $Withdraw(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ in the model M' iff the commitment has been established in the past and $Withdraw$ is in the label of the first transition on this path such that along the prefix of this computation, the content of commitment is false and at the current moment there is another possible computation to satisfy the content of commitment. In the same way, we redefine the semantics of fulfillment, violation, and release actions (see Table 11). Note, furthermore, that the semantics of create, assign, and delegate actions is already in the concrete interpretation as these actions do not need accessability relation.

5.2 Symbolic Model Checking

In a nutshell, given the model M' representing NetBill protocol and a logical formula ϕ describing a property, the model checking is defined as the problem of establishing whether the model M' satisfies ϕ (i.e., $M' \models \phi$) or not (i.e., $M' \not\models \phi$). Like proposed in [10] for CTL^* logic, in our approach the problem of model checking of CTL^{*sc} formulae can be reduced to the problem of checking LTL^{sc} and CTL^{sc} formulae, which correspond to LTL and CTL formulae [10] augmented with social commitments and associated operations. Specifically, we use the MCMAS [16] and NuSMV [5] symbolic model checkers to verify the NetBill protocol against some temporal properties. MCMAS focuses on checking

the properties expressed in CTL^{sc}, while NuSMV is used to check the properties expressed in LTL^{sc}, which is not included in MCMAS specification, as well as the properties expressed in CTL^{sc}. Moreover, we elect these two symbolic model checkers as they efficiently perform the automatic verification over extremely large state space and make our representation more compact.

Table 11. A concrete interpretation of action formulae

	Concrete Interpretation
SC^p	$M', \langle s_i \rangle \models SC^p(Ag_1, Ag_2, \phi)$ iff $(Ag_1, Ag_2) \in \mathbb{L}(s_i)$ & $M', \langle s_i \rangle \models E\phi$
SC^c	$M', \langle s_i \rangle \models SC^c(Ag_1, Ag_2, \tau, \phi)$ iff $(Ag_1, Ag_2) \in \mathbb{L}(s_i)$ & $M', \langle s_i \rangle \models E(\tau \rightarrow SC^p(Ag_1, Ag_2, \phi))$
Withdrawal	$M', \langle s_i, P_i \rangle \models Withdraw(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $\exists j \leq i : M', \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi))$ & (ii) $(s_i, Withdraw, s_{i+1}) \in R$ & $M', \langle s_j, P_{i+1} \downarrow s_j \rangle \models \neg\phi$ & (iii) $\exists P'_i \in \sigma^{s_i} : M', \langle s_j, P'_i \downarrow s_j \rangle \models \phi$
Fulfillment	$M', \langle s_i, P_i \rangle \models Fufill(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $\exists j \leq i : M', \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi))$ & (ii) $(s_i, Fulfill, s_{i+1}) \in R$ & $\forall P'_{i+1} \in \sigma^{s_{i+1}} : M', \langle s_j, P'_{i+1} \downarrow s_j \rangle \models \phi$ & (iii) $\exists P''_i \in \sigma^{s_i} : M', \langle s_j, P''_i \downarrow s_j \rangle \models \neg\phi$
Violation	$M', \langle s_i, P_i \rangle \models Violate(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $\exists j \leq i : M', \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi))$ & (ii) $(s_i, Violate, s_{i+1}) \in R$ & $\forall P'_{i+1} \in \sigma^{s_{i+1}} : M', \langle s_j, P_{i+1} \downarrow s_j \rangle \models \neg\phi$ & (iii) $\exists P''_i \in \sigma^{s_i} : M', \langle s_j, P''_i \downarrow s_j \rangle \models \phi$
Release	$M', \langle s_i, P_i \rangle \models Release(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff (i) $\exists j \leq i : M', \langle s_j, P_i \downarrow s_j \rangle \models SC^p(Ag_1, Ag_2, \phi))$ & (ii) $(s_i, Release, s_{i+1}) \in R$ & $M', \langle s_j, P_{i+1} \downarrow s_j \rangle \models \neg\phi$ & (iii) $\exists P'_i \in \sigma^{s_i} : M', \langle s_j, P'_i \downarrow s_j \rangle \models \phi$

In fact, MCMAS is introduced particulary for multi-agent systems with ISPL (Interpreted Systems Programming Language), which allows us to describe agents. It is based on Ordered Binary Decision Diagrams (OBDDs) technique. The multi-agent system is distinguished into environment agent and standard agents. Environment agent is used to describe boundary conditions, infrastructures and the observation variables shared by standard agents. The agents are modeled as non-deterministic automaton in the form of a set of of instantaneous local states, a set of actions, protocol functions and evolution functions[1]. The main step in our verification workflow is to translate protocol specification into ISPL program. We start by translating the set of interacting agents directly into

[1] The ISPL code of our approach can be downloaded from:
`http://users.encs.concordia.ca/~bentahar/Publications/code/MCMAS.zip`

standard agents in `Agent` section and social commitments into local variables in `Vars` section. Such variables are enumeration type including all possible commitment states, which directly verify whether the protocol is in a conformant state or not. Finally, actions on commitments are directly expressed in `Actions` section in which these actions work as constraints to trigger or stop transitions between commitment states.

On the other hand, NuSMV is a symbolic model checker for CTL and LTL written in ANSI C. It is a reimplementation and extension of SMV, the most widely cited model checker based on Ordered Binary Decision Diagrams (OBDDs). It is able to process files written in an extension of the SMV language. In this language, the different components and functionalities of the system are described by finite state machines (FSMs) and translated into an isolated modules. In our approach, the set of interacting agents are translated into an isolated modules and instantiated in the main module and the commitment states are defined in SMV variables, `VAR`. Such states with actions are used as reasoning rules to evolve the state changes. The transition relation between states and actions is described within `ASSIGN` section where all necessary transitions are captured, initial conditions are defined within `init` statement and evolution is defined within `next` statement[2].

In order to simplify the notations in the next section, rather than using commitments and associated operations explicitly as we discussed in Section 3.2, we use the messages in Figure 2 of NetBill protocol representation. For example, we simply use *offer(Mer)* message in lieu of using *create* action with the associated parameters.

5.3 Experimental Verification Results

We have defined the commitment properties in Section 4, here we introduce other kinds of properties that are important in modeling commitment protocols. These properties can be classified into: Fairness, Safety, Liveness and Reachability properties, which are inspired by similar properties commonly used to check communication protocols in distributed systems.

1. `Fairness constraint`: It is needed to rule out unwanted behaviors of agents (e.g. a communication channel being continuously noisy or a printer being locked forever by a single agent) [10]. In our semantics, if we define the formula: $AG^{+}(AF^{+}\ \neg Delegate(Bank))$ as a fairness constraint, then a computation path is fair iff infinitely often the $Bank$ agent does not delegate commitments. This constraint will enable us to avoid situations such as: a bank B has firstly accepted a delegated commitment but for some reason, it needs to delegate this commitment to another bank B_1, which delegates the commitment back to the bank B. The banks B and B_1 delegate the commitment back and forth infinitely. Thus, by considering fairness constraints, the

[2] The SMV code of our approach can be downloaded from:
`http://users.encs.concordia.ca/~bentahar/Publications/code/NuSMV.zip`

protocol's fairness paths include only the paths that the interacting agents can follow to satisfy their desired states fairly.

2. Safety: Means that "something bad never happens". For example, a bad situation, which should be avoided, is the existence of a path so that in its future the Mer agent delivers the goods, but it is not the case that there is no delivery until the payment has been received: $\neg EF(Deliver(Mer) \wedge \neg A(\neg Deliver(Mer)\ U^{+} Pay(Cus)))$.
3. Liveness: Means that "something good will eventually happen". For example, whenever the Mer agent is in "withdraw" state or "not deliver" state, then he will eventually, in all paths starting at these states, refund the payment to the Cus agent: $AG^{+}(Withdraw(Mer) \vee Not\ Deliver(Mer) \rightarrow AF^{+} Refund(Mer))$.
4. `Reachability`: A particular situation can be reached from the initial state via some computation sequences. For example, in all paths where the Cus agent is always in a request state, a certain state of delivering goods by the Mer agent is eventually reachable: $AG^{+}(Request(Cus) \rightarrow EF^{+} Deliver(Mer))$. Also, this property could be used to show the absence of deadlock in our protocol. Formally: $AG^{+}(Request(Cus) \wedge AG^{+} \neg Deliver(Mer))$, which means that the deadlock is the negation of the reachability property.

The above are only some examples of the possibilities of our language $\mathcal{L}$. In the following, we present the results of three experiments we have conducted to verify the NetBill protocol explained in Section 3.2 with the MCMAS and NuSMV model checkers. In the first experiment we only consider two agents (Cus and Mer agents), in the second one we add the assigned agent (i.e., $Mer1$ agent) and in the third one we add the delegated agent (i.e., $Bank$ agent). Figure 5

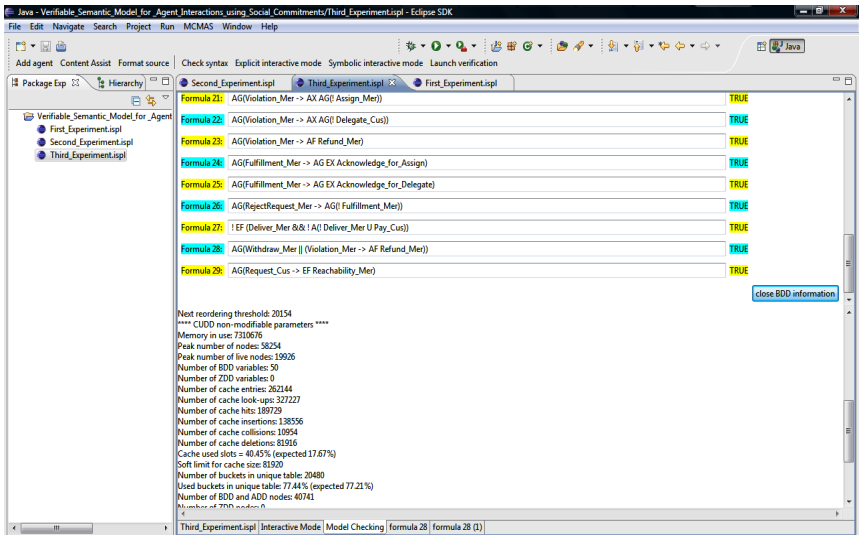

Fig. 5. The report generated by the MCMAS tool

and Figure 6 depict the results of checking such properties of action formulae and the properties discussed above in the third experiment using MCMAS and NuSMV respectively. We only report on the results obtained by MCMAS and NuSMV for checking CTL^{sc} formulae on a laptop running under Windows Vista and equipped with 1.67 GHz Intel(R) Core 2 Duo and 2038 MB of RAM.

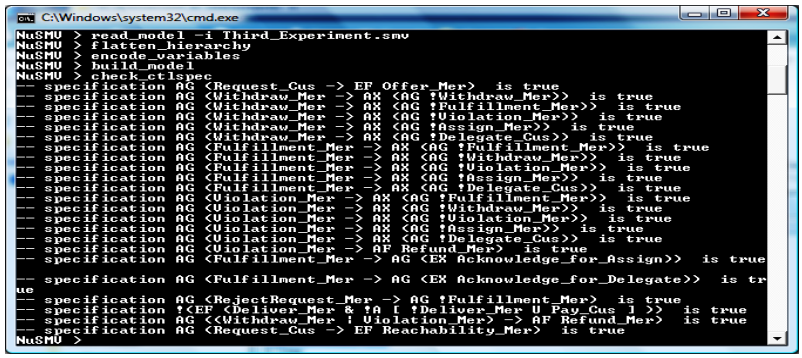

Fig. 6. The report generated by the NuSMV tool

Table 12. OBDDs Statistics Comparison

	First Experiment		Second Experiment		Third Experiment	
	NuSMV	MCMAS	NuSMV	MCMAS	NuSMV	MCMAS
OBDDs Variables	33	24	53	39	73	50
Model Size $\|\tilde{M}\|$	10^{10}	10^{7}	10^{16}	10^{11}	10^{22}	10^{15}
No. of Agents	2	2	3	3	4	4
Total Time	$\approx 0.15s$	$\approx 0.26s$	$\approx 0.34s$	$\approx 0.45s$	$\approx 0.56s$	$\approx 2s$

Table 12 depicts the statistics data of OBDDs for these tools. To evaluate the performance of model checking, we need to analyze the space (i.e., the size of the model) and time requirements (i.e., the time of building the model and time of verification). For example, the number of Boolean variables required to encode this protocol, in the third experiment, is 73 Boolean variables in NuSMV and 50 Boolean variables in MCMAS, then the size of the model (with 4 agents) is $2^{73} \approx 10^{22}$ and $2^{50} \approx 10^{15}$ respectively. The sixth row in Table 12 shows the time results obtained in the verification of the NetBill protocol using the model checkers NuSMV and MCMAS. Notice that, the total time increases when augmenting the number of agents from 2 to 4 agents (see Figure 7) and the verification time in the three experiments is approximately < 0.01s. To conclude this section, the MCMAS model checker is friendly user interface, underpins different logics (e.g., CTL-logic and epistemic logic), supports agents' definitions and performs moderately better than the NuSMV in terms of the model size.

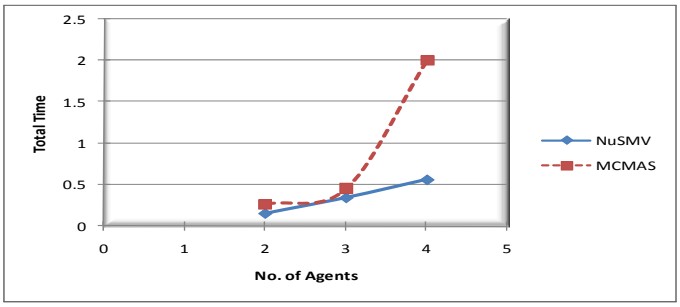

Fig. 7. Comparison of the three experimental results based on the total time

However, NuSMV is better than MCMAS in terms of the total time as some optimization techniques implemented in it, such as on-fly-model checking and cashing.

6 Relevant Literature and Discussion

From the perspective of social commitments, various semantic frameworks have previously been put forward for Agent Communication Language (ACL) using temporal logic. This is because social semantics allow tracing the status of existing commitments at any point in time given observed actions [26] and verifying interacting agents [3,6,28]. In this sense, it is unlike mental semantics that specifies the semantics of communicative acts in terms of pre-and post-conditions contingent on so-called agent's mental states (e.g. beliefs, desires and intentions).

In terms of defining commitment protocols, the commitments capture a high-level meaning of interactions among the agents and provide a rigorous basis for specifying multi-agent protocols in an abstract level. In this line of research, Yolum and Singh [30] have used commitment operations to show how to build and execute commitment protocols and how to reason about them using event calculus. In the same way, Mallya and Singh [18] have showed how to reason about subsumption among commitment protocols and how to refine and aggregate protocols based on commitment semantics and operations. Desai and Singh [11] have studied a composition of commitment protocols and concurrent operations. Yolum in [31] has presented the main generic properties that are required to develop commitment protocols at design time. These properties are categorized into three classes: effectiveness, consistency and robustness. Our proposal belongs to the same line of research and can be used to specify commitment protocols in terms of creation and manipulation of commitments using accessibility relations. It also complements the above frameworks by introducing the semantics of commitment operations and the automatic verification of these protocols against some desirable properties using symbolic model checking. Furthermore, our protocol's properties meet the requirements introduced in [31] in the sense that the reachability and deadlock-freedom can be used to satisfy the same objective of the effectiveness property. The consistency property is achieved in our

protocol by satisfying the safety property. Moreover, the robustness property is satisfied by considering liveness property and fairness paths accompanied with recover states (e.g., refund state) that capture the protocol failures. Hence, our approach can be applied to verify the protocol's properties defined in [31]. Furthermore, the commitment operations are fully captured at the semantic level and not only in terms of reasoning rules which are defined on top of the protocols as in [17,30].

Furthermore, our logical model provides a novel unified semantics not only for social commitments, but also for associated operations within the same framework using a new logic that extends CTL^*. Recently, Singh in [23] has delineated the model-theoretic semantics of commitments by postulating some rules as ways of using and reasoning with commitments. This model combines two commitments (practical and dialogical), in the sense that when a commitment arises within an argument and the content is satisfied with the same argument, then practical commitment would be satisfied. However, this model does not include the semantics of commitment operations. Chopra and Singh [8] have used the theoretical model proposed in [23] to study the semantics of commitment operations with message patterns that implement commitment operations with some constraints on agents' behaviors to tackle the problem of autonomy and heterogeneity in distributed systems and to define "commitment alignment"[8]. This semantics is expressed in terms of the set of propositions that can be inferred from the observation sequence that agents sent or received. Moreover, this semantics must correspond to the postulates introduced in [23], as well as the content of commitment is restricted in a disjunctive and a conjunctive normal forms [8]. However, the formal language of those postulates is based on enhancing LTL with social and dialogical commitments. Thus, this language is less expressive than the formal language introduced here, which is more compatible with agent choices by representing the content of commitment as a path formula.

Our semantics is based on a Kripke-like structure with accessibility relations, which makes the proposed model computationally grounded [27] and allows its verification using model checking. Our proposal is different from the semantics introduced by Singh in [23] which is not based on a Kripke-structure and therefore difficult to be model checked. Venkatraman and Singh [28] have presented an approach for testing locally whether the behavior of an agent in open systems complies with a commitment protocol specified in temporal logic (CTL). The proposed approach complements this work by introducing the symbolic model checking to verify the interactions between agents. Cheng [6] has introduced model checking using Promela and Spin to verify commitment-based business protocols and their compositions based on LTL logic. The specification language proposed here is not only LTL^{sc}, but also CTL^{sc} and we use the MCMAS and NuSMV model checkers, which underpin these logics and which are computationally more efficient than automata-based model checkers such as Spin.

In terms of the semantics of commitment operations, Boella et al. in [4] have defined the semantics of fulfillment of propositional commitment in terms of creditor's believes so that the commitment is fulfilled when the creditor does

not believe that the commitment's content is false and he cannot challenge it anymore. However, our semantics is concrete away from the internal design of agents and provides a social meaning to agent message exchanges (i.e., actions). The semantics defined here for conditional commitments is different from the semantics defined in [20] and [23]. In [20], conditional commitments are considered as intentions, while commitments are social artifacts and different from private intentions. In [23], Singh models conditional commitments as fundamental and unconditional commitments as special cases where the antecedent is true. In our semantics, the conditional commitments are transformed into unconditional commitments in all accessible paths where the underlying condition is true. The semantics proposed here is close to the semantics introduced in [2], but it does not suffer from the "recursion" problem as is the case in [2]. Recursion means the semantics of one operation depends on the semantics of one or more other operations. Consequently, the model checking technique for this logic is very complex and suffers from the "state explosion" problem in the early phases. On the contrary, the semantics we presented here is independent, for each operation, of the semantics of other operations.

The semantics we have proposed here for assignment and delegation operations is entirely different from the ones given in [7,9,17] and [18]. Specifically, the assignment and delegation operations should consider that the content of the new resulting commitment could be different from, and has a logical relationship with the content of the assigned and delegated commitment. This issue is not captured in previous frameworks. In addition, unlike our semantics, the violation operation has been disregarded. Torroni et al. have defined in [26] an abstract framework based on computational logic and event calculus to formalize the evolution of commitments in time inspired by Mallya's work [17]. Unlike our work, they do not consider the semantics of commitment operations and logical relationship between delegated commitment and the original one. In fact, the proposed work can complement this framework with automatic verification using model checking in lieu of reasoning rules, which are defined in terms of event calculus to provide run time and static verification.

7 Conclusion

In this paper we have defined a new semantics for social commitment and associated operations by enhancing CTL^*. The resulting logic (CTL^{*sc}) allows us to specify interesting temporal properties for MASs. This semantics satisfies four criteria introduced in [22]: formal (based on logic of time), declarative (involves assertions about commitments), verifiable (using a symbolic model checking), and meaningful (every message can be expressed in terms of commitments). We have presented and implemented an approach to automatically verify interacting agents modeled as a Kripke-like structure using two symbolic model checkers: MCMAS and NuSMV. We tested our implementation by means of NetBill protocol through three experimental results. These experiments revealed that our framework can be employed successfully in verifying communication protocols

with large state spaces (in the order of $2^{73} \approx 10^{22}$). As future work, these results are encouraging to be applied in verifying various business processes such as web service compositions. Also, we plan to use our semantics to specify properties when commitments are not aligned meaning that the exchanged messages can be delayed so the participants do not observe the same commitments or the exchanged messages are understood with different meanings [8].

Acknowledgements

We would like to thank the reviewers for their valuable comments and suggestions. Jamal Bentahar would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC), Fonds québécois de la recherche sur la nature et les technologies (FQRNT), and Fonds québécois de la recherche sur la Société et la culture (FQRSC) for their financial support.

References

1. Bentahar, J., Moulin, B., Meyer, J.-J.C., Chaib-draa, B.: A Logical Model for Commitment and Argument Network for Agent Communication. In: 3rd International Joint Conference on AAMAS, pp. 792–799 (2004)
2. Bentahar, J., Moulin, B., Meyer, J.-J.C., Lespérance, Y.: A New Logical Semantics for Agent Communication. In: Inoue, K., Satoh, K., Toni, F. (eds.) CLIMA 2006. LNCS (LNAI), vol. 4371, pp. 151–170. Springer, Heidelberg (2007)
3. Bentahar, J., Meyer, J.-J.C., Wan, W.: Model Checking Communicative Agent-based Systems. Knowledge-Based Systems, Special Issue on Intelligent Software Design 22(3), 142–159 (2009)
4. Boella, G., Damiano, R., Hulstijn, J., Torre, L.: Distinguishing Propositional and Action Commitment in Agent Communication. In: Workshop of Computational Modeles of Natural Argument, CMNA 2007 (2007)
5. Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: NuSMV 2: An Open Source Tool for Symbolic Model Checking. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 241–268. Springer, Heidelberg (2002)
6. Cheng, Z.: Verifying Commitment based Business Protocols and their Compositions: Model Checking using Promela and Spin. PhD. thesis, North Carolina State University (2006)
7. Chopra, A.K., Singh, M.P.: Constitutive Interoperability. In: 7th International Joint Conference on AAMAS, vol. 2, pp. 797–804. ACM Press, New York (2008)
8. Chopra, A.K., Singh, M.P.: Multiagent Commitment Alignment. In: 8th International Joint Conference on AAMAS, vol. 2, pp. 937–944. ACM Press, New York (2009)
9. Chopra, A.K., Singh, M.P.: Nonmonotonic Commitment Machines. In: Dignum, F. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 183–200. Springer, Heidelberg (2004)
10. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge (1999)
11. Desai, N., Chopra, A.K., Singh, M.P.: Representing and Reasoning About Commitments in Business Processes. In: 22nd AAAI Conference on Artificial Intelligence, pp. 1328–1333 (2007)
12. Emerson, E.A., Halpern, J.Y.: Sometimes and Not Never, Revisited: on Branching versus Linear Time Temporal Logic. Journal of ACM 33(1), 151–178 (1986)

13. Huhns, M., Bridgeland, D.M.: Multi-Agent Truth Maintenance. IEEE Transactions on Systems, Man, and Cybernetics 21(6) (1991)
14. Fornara, N., Colombetti, M.: Operational Specification of a Commitment-based Agent Communication Language. In: 1st International Joint Conference on AAMAS, pp. 535–542 (2002)
15. Fornara, N., Viganò, F., Verdicchio, M., Colombetti, M.: Artificial Institutions: A Model of Institutional Reality for Open Multiagent Systems. AI and Law 16(1), 89–105 (2008)
16. Lomuscio, A., Raimondi, F.: MCMAS: A Model Checker for Multi-Agent Systems. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 450–454. Springer, Heidelberg (2006)
17. Mallya, A.U., Yolum, P., Singh, M.P.: Resolving Commitments among Autonomous Agents. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 166–182. Springer, Heidelberg (2004)
18. Mallya, A.U., Singh, M.P.: An Algebra for Commitment Protocols. Journal of Autonomous Agents and Multi-Agent Systems 14(2), 143–163 (2007)
19. El-Menshawy, M., Bentahar, J., Dssouli, R.: A New Semantics of Social Commitments using Branching Space-Time Logic. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009), Logics for Intelligent Agents and Multi-Agent Systems, vol. 3, pp. 492–496 (2009)
20. Shakil, M.K., Yves, L.: On the Semantics of Conditional Commitment. In: 5th International Joint Confernce on AAMAS, pp. 1337–1344 (2006)
21. Singh, M.P.: An Ontology for Commitments in Multi-Agent Systems: Toward a Unification of Normative Concepts. AI and Law 7, 97–113 (1999)
22. Singh, M.P.: A Social Semantics for Agent Communication Languages. In: Dignum, F.P.M., Greaves, M. (eds.) Issues in Agent Communication. LNCS, vol. 1916, pp. 31–45. Springer, Heidelberg (2000)
23. Singh, M.P.: Semantical Considerations on Dialectical and Pratical Commitments. In: 23rd AAAI Confernce on Artificial Intelligence, pp. 176–181 (2008)
24. Singh, M.P.: Agent Communication Languages: Rethinking the Principles. IEEE Computer 31(12), 40–47 (1998)
25. Telang, P.R., Singh, M.P.: Business Modeling via Commitments. In: Vo, Q.B. (ed.) SOCASE 2009. LNCS, vol. 5907, pp. 111–125. Springer, Heidelberg (2009)
26. Torroni, P., Chesani, F., Mello, P., Montali, M.: Social Commitments in Time: Satisfied or Compensated. In: Baldoni, M., van Riemsdijk, M.B. (eds.) DALT 2009. LNCS (LNAI), vol. 5948, pp. 232–247. Springer, Heidelberg (2009)
27. Wooldridge, M.: Computationally Grounded Theories of Agency. In: 4th International Conference on Multi-Agent Systems (ICMAS 2000). IEEE Press, Los Alamitos (2000)
28. Venkatraman, M., Singh, M.P.: Verifying Compliance with Commitment Protocols: Enabling Open Web-Based Multiagent Systems. In: The Autonomous Agents and Multi-Agent Systems, vol. 3, pp. 217–236 (1999)
29. Verdicchio, M., Colombetti, M.: A Logical Model of Social Commitment for Agent Communication. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 128–145. Springer, Heidelberg (2004)
30. Yolum, P., Singh, M.P.: Flexible Protocol Specification and Execution: Applying Event Calculus Planning using Commitments. In: 2nd International Joint Confernce on AAMAS, pp. 527–534 (2002)
31. Yolum, P.: Design Time Analysis of Multi-Agent Protocols. Journal of Data Knowladge Engineering 63(1), 137–154 (2007)