Symbolic Model Checking for Agent Interactions (Extended Abstract)

Mohamed El-Menshawy, Wei Wan
Departement of Electrical and Computer Eng.
Concordia University, Montreal, Canada
{m_elme,w_wan}@encs.concordia.ca

Jamal Bentahar, Rachida Dssouli
Concordia Institute for Information System Eng.
Concordia University, Montreal, Canada
{bentahar,dssouli}@ciise.concordia.ca

ABSTRACT

In this paper, we address the issue of the specification and verification of commitment protocols having a social semantics. We begin with developing a new language to formally specify these protocols and desirable properties by enhancing CTL^* logic with modalities of commitments and actions on these commitments. We also present a symbolic model checking algorithm for commitments and their actions based on OBDDs. Finally, we present an implementation and experimental results of the proposed protocol using the NuSMV and MCMAS symbolic model checkers.

Categories and Subject Descriptors

D.2.4 [**Software/Program Verification**]: Model Checking

General Terms

Design, Verification

Keywords

Commitment Protocol, Symbolic Model Checking

1. THE LOGICAL MODEL

In the past years, some interesting approaches have been proposed for defining a formal specification of commitment protocols using different logics such as CTL [4]. However, specifying protocols that ensure flexible interactions is necessary, but not sufficient to automatically verify the conformance of these protocols with some desirable properties. This paper addresses the above challenge by developing a new language that enhances CTL^* [2] with commitments and actions on these commitments to specify commitment protocols and verify them using symbolic model checking. The resulting logic is called $ACTL^{*sc}$.

The syntax of our language $\mathcal{L}$ contains two new modalities: SC^p for unconditional commitments and SC^c for conditional commitments as well as action formulae applied to commitments. We also use $\Phi_p = \{p, p_1, \ldots\}$ for a set of atomic propositions, $\Phi = \{\phi, \tau, \ldots\}$ for a set of propositional formulae, $\mathtt{AGT} = \{Ag, Ag_1, \ldots\}$ for a set of agent names and $\mathtt{ACT} = \{\alpha, \alpha_1, \ldots\}$ for a set of actions. Table 1 gives the formal syntax of our logic $ACTL^{*sc}$ expressed in BNF where "::=" and "|" are meta-symbols of this grammar.

Table 1: The Syntax of $ACTL^{*sc}$ Logic

$\mathcal{S} ::= p \mid \neg\, \mathcal{S} \mid \mathcal{S} \vee \mathcal{S} \mid \mathcal{S} \wedge \mathcal{S} \mid E\mathcal{P} \mid A\mathcal{P} \mid \mathcal{C}$
$\mathcal{P} ::= \mathcal{S} \mid \mathcal{P} \vee \mathcal{P} \mid \mathcal{P} \wedge \mathcal{P} \mid X\mathcal{P} \mid \mathcal{P}\ U\mathcal{P} \mid \alpha(Ag, \mathcal{C})$
$\mathcal{C} ::= SC^p(Ag_1, Ag_2, \mathcal{P}) \mid SC^c(Ag_1, Ag_2, \mathcal{P}, \mathcal{P})$

The intuitive meanings of these temporal operators are straightforward from CTL^* [2]. $SC^p(Ag_1, Ag_2, \phi)$ means Ag_1, the debtor, is committed towards Ag_2, the creditor, to bring about ϕ, $SC^c(Ag_1, Ag_2, \tau, \phi)$ means Ag_1 is only committed towards Ag_2 to bring about ϕ when the condition τ holds and $\alpha(Ag, \mathcal{C})$ means an action α is performed by the agent Ag on the commitment $\mathcal{C}$. Such actions are either two-party actions: create, fulfill, violate, and release; or three-party actions: delegate and assign.

The formal model M associated with $ACTL^{*sc}$ corresponds to the commitment protocol (see Section 2) and is defined as a Kripke-structure as follows: $M = \langle \mathbb{S}, \mathtt{ACT}, \mathtt{AGT}, R_t, \mathbb{V}, \mathbb{R}_{scp}, \mathbb{R}_{scc}, \mathbb{L}, s_0 \rangle$ where: $\mathbb{S}$ is a set of states; $\mathtt{ACT}$ and $\mathtt{AGT}$ are defined above; $R_t \subseteq \mathbb{S} \times \mathtt{ACT} \times \mathbb{S}$ is a transition relation among states; $\mathbb{V} : \Phi_p \rightarrow 2^{\mathbb{S}}$ is an evaluation function; $\mathbb{R}_{scp} : \mathbb{S} \times \mathtt{AGT} \times \mathtt{AGT} \rightarrow 2^{\sigma}$, where σ is the set of all paths, is an accessibility modal relation that associates with a state s the set of possible paths along which the unconditional commitments made by the debtor towards the creditor at s hold; $\mathbb{R}_{scc} : \mathbb{S} \times \mathtt{AGT} \times \mathtt{AGT} \rightarrow 2^{\sigma}$ is an accessibility modal relation for conditional commitments; $\mathbb{L} : \mathbb{S} \rightarrow 2^{\mathtt{AGT} \times \mathtt{AGT}}$ associates a given state s with a set of pairs and each pair represents the two interacting agents in s; and $s_0 \in \mathbb{S}$ is the initial state.

Excluding commitment modalities, the semantics of $ACTL^{*sc}$ state formulae is as usual (semantics of CTL^*). $M, \langle s_i \rangle \models \phi$ means the model M satisfies the state formula ϕ at s_i. To make our semantics computationally grounded, the accessibility relations $\mathbb{R}_{scp}$ and $\mathbb{R}_{scc}$ should be given a concrete (computational) interpretation. This paper adopts a simple solution saying that a commitment about ϕ made by Ag_1 towards Ag_2 is satisfied at state s_i iff there is at least a possible computation starting at this state satisfying ϕ. Formally: $M, \langle s_i \rangle \models SC^p(Ag_1, Ag_2, \phi)$ iff $(Ag_1, Ag_2) \in \mathbb{L}(s_i)$ and $M, \langle s_i \rangle \models E\phi$

As for unconditional commitments, we have the following: $M, \langle s_i \rangle \models SC^c(Ag_1, Ag_2, \tau, \phi)$ iff $(Ag_1, Ag_2) \in \mathbb{L}(s_i)$ and $M, \langle s_i \rangle \models E(\tau \rightarrow SC^p(Ag_1, Ag_2, \phi))$ where "$\rightarrow$" stands for logical implication. A path P starting at s_i satisfies $\alpha(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ in the model M iff α is the label of the

Cite as: Symbolic Model Checking for Agent Interactions (Extended Abstract), M. El-Menshawy, W. Wan, J. Bentahar and R. Dssouli, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1555-1556

first transition on this path. Formally:
$M, \langle s_i, P\rangle \models \alpha(Ag_1, SC^p(Ag_1, Ag_2, \phi))$ iff
$(s_i, \alpha_{i+1}, s_{i+1}) \in R_t$ and $\alpha_{i+1} = \alpha$

2. COMMITMENT PROTOCOL

In this section, we consider NetBill protocol to clarify the commitment protocol specification. The protocol begins with a customer (Cus) requesting a quote for some goods, followed by a merchant (Mer) sending the quote as an offer or rejecting this request. The Mer agent, before delivering the goods to Cus, can withdraw the offer. If the Cus agent pays for the goods and the Mer agent delivers them, then Mer fulfills his commitment. If the Cus agent pays for the goods, but the Mer agent does not deliver them in a specified time, then the Mer agent violates his commitment and he must refund payment to Cus. The Cus agent can withdraw his commitment before paying for the goods. The Mer agent can assign the commitment to another merchant (say Mer_1). The Cus agent can delegate the commitment to a financial company (say $Bank$) to pay the Mer agent on his behalf. A protocol run is completed iff all the unconditional commitments that have been created are resolved [4].

3. SYMBOLIC MODEL CHECKING

Having defined a concrete interpretation of accessibility relations using the existential operator E, the concrete model becomes $M = \langle \mathbb{S}, \mathtt{ACT}, \mathtt{AGT}, R_t, \mathbb{V}, \mathbb{L}, s_0\rangle$. In this case, we can easily use the standard procedure introduced in [2] to encode each element in our model with OBDDs.

Table 2: Symbolic model checking procedures

$SMC_{scp}(Ag_1, Ag_2, \phi, M)$ { $X = SMC(E\phi, M)$; $Y = \{s \in \mathbb{S} \mid (Ag_1, Ag_2) \in \mathbb{L}(s)\}$; return $X \cap Y$; }
$SMC_{scc}(Ag_1, Ag_2, \tau, \phi, M)$ { $X = SMC(E\tau, M)$; return $\neg X \cap SMC_{scp}(Ag_1, Ag_2, \phi, M)$; }
$SMC_{act}(\alpha, Ag, \mathcal{C}, M)$ { $X = \{s \mid \exists s' \in \mathbb{S}$ and $R_t(s, \alpha, s')\}$; return X; }

This encoding makes our representation more compact and enables us to use symbolic model checking techniques. Let us consider for example, the encoding of the transition relations in R_t. Let $R_{t_1} = (s, \alpha, s')$ be a transition relation in R_t. Its Boolean representation is given by $\overline{v} \wedge \overline{\alpha} \wedge \overline{v'}$, where $\overline{v}$ and $\overline{v'}$ are the Boolean representation of states s and s' respectively and $\overline{\alpha}$ is the Boolean encoding for the action. Thus, the whole transition relation R_t is encoded by a Boolean formula as well by taking the disjunction of all the transition steps. The symbolic model checking procedures for SC^p, SC^c modalities and action formulae are shown in Table 2. Note that, $SMC(E\phi, M)$ (resp. $SMC(E\tau, M)$) is the standard procedure used to compute the set of states $[\![\phi]\!]$ (resp. $[\![\tau]\!]$) in which a formula ϕ (resp. τ) holds.

4. IMPLEMENTATION

As proposed in [2] for CTL^* logic, in our approach the problem of model checking $ACTL^{*sc}$ formulae can be reduced to the problem of checking $ALTL^{sc}$ and $ACTL^{sc}$ formulae that correspond to LTL and CTL formulae [2] augmented with commitments and actions on these commitments. Specifically, we use the MCMAS [3] and NuSMV symbolic model checkers to verify our commitment protocol against some given properties. MCMAS is used to check properties expressed in $ACTL^{sc}$, while NuSMV focuses on checking properties expressed in $ALTL^{sc}$ that is not included in MCMAS.

4.1 Experimental Results

We firstly introduce some desirable properties to verify the correctness of the proposed protocol using $ALTL^{sc}$ and $ACTL^{sc}$. Such properties are mainly related to fairness constraint (to avoid unwanted behaviors of agents), reachability (to reach to a particular situation from initial state), safety and liveness, commitment context (e.g., a commitment can only be created by the debtor), and commitment actions (e.g., if a commitment is withdrawn, then it cannot be withdrawn again). Following, we present two experimental results to verify the proposed protocol with the MCMAS and NuSMV model checkers. In the first experiment, we only consider two-party actions on commitments. In the second one, we add three-party actions: delegate and assign. We only report results obtained by MCMAS and NuSMV for checking $ACTL^{sc}$ formulae on a laptop running Windows XP SP2 on AMD Dual Core 2.20 GHz with 896 MB of RAM. We use the statistics data for OBDDs to evaluate the performance of our approach (see Table 3). In the first experiment, the

Table 3: OBDDs Statistics Comparison

	First Experiment		Second Experiment	
	NuSMV	MCMAS	NuSMV	MCMAS
OBDDs Variables	21	27	23	44
Model Size	$\approx 10^{12}$	$\approx 10^{16}$	$\approx 10^{14}$	$\approx 10^{26}$
Number of Agents	3	3	5	5

number of OBDDs variables in NuSMV (resp. MCMAS) is 21 (resp. 27), then the model size is $2^{21} \approx 10^6$ (resp. $2^{27} \approx 10^8$). In the second experiment, the model size is $2^{23} \approx 10^7$ in NuSMV and $2^{44} \approx 10^{13}$ in MCMAS. In terms of comparisons, the MCMAS model checker allows us to describe agents, but NuSMV is better in terms of the model size, which increases with the number of agents whilst the verification time is approximately the same < 0.01s.

Our approach is entirely different from the previous proposals that verify commitment protocols [4] and from the standard literature about the verification of interaction protocols (see for example, [1]) in terms of the proposed logic and symbolic model checking that synthesizes $ALTL^{sc}$ and $ACTL^{sc}$ algorithms.

5. REFERENCES

[1] M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and C. Schifanella. Verifying protocol conformance for logic-based communicating agents. In *CLIMA V*, pages 196–212, 2004.

[2] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, Cambridge, 1999.

[3] A. Lomuscio, H. Qu, and F. Raimondi. Mcmas: A model checker for the verification of multi-agent systems. In *In CAV*, pages pp. 682–688, 2009.

[4] M. Venkatraman and M. P. Singh. Verifying compliance with commitment protocols. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, 1999.