Towards Model Checking Interpreted Systems[*]

F. Raimondi, A. Lomuscio
Department of Computer Science
King's College
London WC2R 2LS, UK
{franco,alessio}@dcs.kcl.ac.uk

M.J. Sergot
Department of Computing
Imperial College
London SW7 2BZ, UK
mjs@doc.ic.ac.uk

ABSTRACT

We show how it is possible to pair the NuSMV model checker with Akka, a software platform used to check validity of propositional modal formulas, to verify static properties of multi-agent systems formalised on interpreted systems semantics.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence-Multiagent systems

General Terms

Verification

Keywords

Model checking, Interpreted Systems, Epistemic Logic, Deontic Logic

1. INTRODUCTION

Though J. Halpern and M. Vardi suggested the use of model checking techniques in the verification of multi-agent systems in 1991 ([6]), it is only recently that results along these lines have been achieved ([2, 11, 12, 10, 7]). In this paper we try to bring together interpreted systems semantics [5] and model checking [4] on a concrete and well-defined scenario—a variant of the bit transmission problem. The way we tackle the problem is as follows:

1. We study our scenario formally using the formalism of deontic interpreted systems [9].
2. We code this representation in NuSMV and feed it to a NuSMV checker [3].
3. We use the NuSMV checker to produce the set of runs of the system, and deduce from there the set of reachable states of the system.
4. We feed these into the modal checker Akka [1].
5. We use the Akka front-end to check the epistemic properties of the scenario.

The scenario examined here was investigated (together with a more complex variation) in [8] without model checking techniques.

Due to space considerations we assume some familiarity with interpreted systems, model checking, and deontic interpreted systems [9].

2. THE BIT TRANSMISSION PROBLEM

The bit-transmission problem [5] involves two agents, a *sender* S, and a *receiver* R, communicating over a faulty communication channel E. S wants to communicate some information to R. The channel may drop messages but will not change the value of a bit being sent. Communication is established when R knows the value of the bit and S knows that R knows. We represent this scenario in interpreted systems semantics.

For the sender S, it is enough to consider four possible local states. They represent the value of the bit S is attempting to transmit, and whether or not S has received an acknowledgement from R. Three possible local states are enough to capture the state of R: the value of the received bit, and ϵ representing a circumstance under which no bit has been received yet. To model the environment we consider four local states, representing the possible combinations of messages that have been sent in the current round:

$$L_S = \{0, 1, (0, ack), (1, ack)\}, \quad L_R = \{0, 1, \epsilon\}.$$
$$L_E = \{(.,.), (sendbit, .), (., sendack), (sendbit, sendack)\},$$

where '.' represents configurations in which no message has been sent by the corresponding agent.

For every agent in the system, the set of actions is:

$$Act_S = \{sendbit(\mathit{0}), sendbit(\mathit{1}), \lambda\}, \quad Act_R = \{sendack, \lambda\}.$$
$$Act_E = \{S{-}R,\ S{\rightarrow},\ {\leftarrow}R,\ -\}$$

Here λ stands for no action ('no-op'). The actions Act_E for the environment correspond to the actual delivery of messages between S and R on the unreliable communication channel.

We can model the evolution of the system by means of a function $\pi : G \times Act \to G$, where $Act = Act_S \times Act_R \times Act_E$ is the set of joint actions for the system and $G \subseteq L_S \times L_R \times L_E$ is the set of global states. For the example under consideration the protocols can be defined as follows:

$$P_S(0) = sendbit(0), \quad P_S(1) = sendbit(1),$$
$$P_S((0, ack)) = P_S((1, ack)) = \lambda,$$
$$P_R(\epsilon) = \lambda, \quad P_R(0) = P_R(1) = sendack,$$
$$P_E(l_E) = Act_E = \{S{-}R,\ S{\rightarrow},\ {\leftarrow}R,\ -\}, \quad \text{for all } l_E \in L_E.$$

Given the description above, we can implement the scenario in NuSMV by representing the local states as NuSMV variables and translating the protocol functions and system evolution function π into the syntax of NuSMV.

We modified the NuSMV code to generate the reachable global states of the system, producing a Kripke model in the syntax of Akka [1]. Akka offers a Kripke model editor and supports model testing. We are now in a position to check any epistemic property of the system. To this end, let us name IS_{b} the model obtained by following the process described above, on which an appropriate set

[*] The authors acknowledge support from EU project ALFEBIITE, IST-1999-20298 and from the Nuffield Foundation through grant NAL/00690/G.

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

of propositional variables is interpreted in a natural way [8]. The following propositions are easily translated into the syntax of Akka. As expected, the propositions can be tested to be valid on IS_{b}.

$$IS_{\mathrm{b}} \models \mathbf{recbit} \rightarrow \big(K_R\,(\mathbf{bit} = \mathbf{0}) \vee K_R\,(\mathbf{bit} = \mathbf{1})\big)$$
$$IS_{\mathrm{b}} \models \mathbf{recack} \wedge (\mathbf{bit} = \mathbf{0}) \rightarrow K_S\,K_R\,(\mathbf{bit} = \mathbf{0})$$
$$IS_{\mathrm{b}} \models \mathbf{recack} \rightarrow K_S\big(K_R\,(\mathbf{bit} = \mathbf{0}) \vee K_R\,(\mathbf{bit} = \mathbf{1})\big)$$

2.1 Faulty Receiver

Suppose that the receiver R may send acknowledgements even when it is not supposed to, i.e., when it has not yet received the value of the bit. For this version of the problem we introduce new local states for the receiver R: (ϵ, f) is the local state in which R did not receive a bit but nevertheless R sent an acknowledgement. The local states $(0, f)$ and $(1, f)$ of R represent the case where R has received the value of the bit and has sent an erroneous acknowledgement at some time in the past. All the local states of S and E are green. We thus have:

$$\begin{aligned}&L''_S = G''_S = \{0, 1, (0, ack), (1, ack)\}, \quad R''_S = \emptyset,\\ &R''_E = \emptyset, L''_E = G''_E,\\ &G''_E = \{(.,.), (sendbit, .), (., sendack), (sendbit, sendack)\}.\end{aligned}$$

For R, we define the local states as follows:

$$G''_R = \{0, 1, \epsilon\},\ R''_R = \{(0, f), (1, f), (\epsilon, f)\},\ L''_R = G''_R \cup R''_R.$$

Given that the two sets of local states for S and E have not changed we can keep the functions P_S and P_E as for the basic version. We need to extend P_R so that it is defined also on the red local states of R.

$$\begin{aligned}&P''_R(\epsilon) = P_R(\epsilon) = \lambda,\\ &P''_R(0) = P''_R(1) = P_R(0) = P_R(1) = sendack,\\ &P''_R((0, f)) = P''_R((1, f)) = P''_R((\epsilon, f)) = Act_R = \{sendack, \lambda\}\end{aligned}$$

The NuSMV implementation of this version of the bit transmission problem is an extension of the code for the basic version. As in the previous case, NuSMV outputs the reachable global states, from which in turn we can create a model with epistemic relations for K_S and K_R. Further, it is straightforward to classify the global states into red and green states for R and so create a model on which the doubly relativised operator $\widehat{K}^R_S$ [9] can be interpreted. It is possible to check that none of the epistemic formulas presented in the earlier section hold in this version. However, a particular form of knowledge still holds. If S makes the assumption of R's correct functioning behaviour, then, upon receipt of an acknowledgement, it makes sense for S to assume that R does know the value of the bit; this is exactly what is captured by $\widehat{K}^R_S$. Indeed, using Akka we are able to check the validity of the following formulas in the model:

$$IS''_{\mathrm{b}} \models \mathbf{recack} \rightarrow \widehat{K}^R_S\big(K_R\,(\mathbf{bit} = \mathbf{0}) \vee K_R\,(\mathbf{bit} = \mathbf{1})\big)$$
$$IS''_{\mathrm{b}} \models \mathbf{recack} \wedge (\mathbf{bit} = \mathbf{0}) \rightarrow \widehat{K}^R_S\,K_R\,(\mathbf{bit} = \mathbf{0})$$

We refer to [8] for more details.

3. CONCLUSIONS

We see the contribution of this paper as being twofold. Firstly, we provide a simple methodology for checking static epistemic properties in interpreted systems. Secondly, we find the technical results on violations of the bit-transmission protocol interesting on their own merits.

Finally, we have tried to show that in some examples verifying *static* epistemic and deontic properties is sufficient to establish basic properties of the system. Still, we would like to extend the methodology to deal with the full *dynamic* case, i.e., to move to a system in which we can check temporal deontic and epistemic formulas.

4. ACKNOWLEDGEMENTS

The authors are grateful to Marco Pistore of the NuSMV development team for providing important directions for modifying the source code of NuSMV to deal with the collection of global states from the NuSMV software. We also thank Lex Hendriks for providing us with a new release of Akka.

5. REFERENCES

[1] Akka, A Workbench for Mathematical Logic. http://turing.wins.uva.nl/∼lhendrik/.

[2] M. Benerecetti, F. Giunchiglia, and L. Serafini. Model checking multiagent systems. *Journal of Logic and Computation*, 8(3):401–423, June 1998.

[3] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. NuSMV: A new symbolic model verifier. *Lecture Notes in Computer Science*, 1633, 1999.

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

[5] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.

[6] J. Halpern and M. Y. Vardi. Model checking vs. theorem proving: A manifesto. In J. Allen, R. E. Fikes, and E. Sandewall, editors, *Proceedings 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning, KR'91*, pages 325–334. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[7] A. Lomuscio and W. Penczek. Bounded model checking for interpreted systems. Technical report, Institute of Computer Science of the Polish Academy of Sciences, 2002.

[8] A. Lomuscio and M. Sergot. Violation, error recovery, and enforcement in the bit transmission problem. In *Proceedings of DEON'02*, London, May 2002. Elsevier. To appear in the Journal of Applied Logic.

[9] A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75, 2003.

[10] R. van der Meyden and N. V. Shilov. Model checking knowledge and time in systems with perfect recall. *FSTTCS: Foundations of Software Technology and Theoretical Computer Science*, 19, 1999.

[11] R. van der Meyden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. Submitted, 2002.

[12] M. Wooldridge, M. Fisher, M.-P. Huget, and S. Parsons. Model checking multi-agent systems with MABLE. In M. Gini, T. Ishida, C. Castelfranchi, and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 952–959. ACM Press, July 2002.