Verifying Realizability and Reachability in Recursive Interaction Protocol Specifications

(Extended Abstract)

Hywel Dunn-Davies
Imperial College London
London SW7 2BZ, UK
hdd2@doc.ic.ac.uk

Jim Cunningham
Imperial College London
London SW7 2BZ, UK
rjc@doc.ic.ac.uk

ABSTRACT

The Propositional Statechart language described in [4] makes it possible to provide unambiguous joint diagrammatic specifications of complex recursive interaction protocols such as the multilateral voting protocol. Recursive protocols described using Propositional Statecharts are not necessarily finite state, meaning that verification of properties of these protocols cannot be achieved solely by means of finite state model checking. In this paper we discuss the verification of realizability, and reachability analysis, in recursive joint protocol specifications expressed using joint Propositional Statecharts.

Categories and Subject Descriptors

D.2.2 [**Design Tools and Techniques**]: State Diagrams

General Terms

Design, Languages

Keywords

Verification, Interaction Protocols, Propositional Statecharts

1. INTRODUCTION

The current paper considers the verification of two key properties of joint Propositional Statechart representations (henceforth known as joint PSCs)[1]. These are realizability [6] (also referred to by some authors as enactability [3]), and the reachability of states.

In section 2 of this paper we introduce the concept of realizability and discuss the verification of realizability in joint PSCs. Then, in section 3, we discuss reachability analysis in joint PSCs. We present a brief discussion in section 4.

[1]Joint PSCs, described in [4], represent the state of a protocol as a whole, while agent-centred PSCs, described in [5], represent a protocol from the point of view of each individual agent.

Cite as: Verifying Realizability and Reachability in Recursive Interaction Protocol Specifications, (Extended Abstract), Hywel Dunn-Davies, Jim Cunningham, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1233–1234

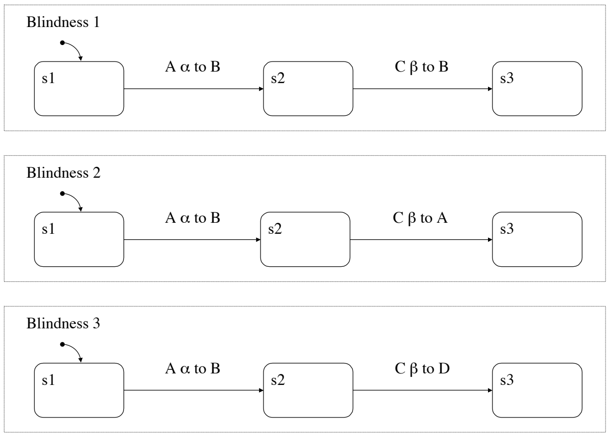

Figure 1: Three examples of blindness. *'A α to B' should be read as 'Agent A sends message α to agent B'.*

2. VERIFYING REALIZABILITY

A joint protocol representation is taken to be realizable (or enactable) if there exists an implementation of a protocol (defined a set of agent-centred protocol representations) that generates exactly the set of behaviours described in the joint specification [6]. Here we consider a number of situations in which joint protocol representations may not be realizable, presenting examples where appropriate using joint PSCs. We then discuss whether polynomial time algorithms exist for the verification of realizability of joint PSCs.

2.1 Problems Related to Realizablility

An important class of realizability problem is that of *blindness*. Blindness may arise in interactions involving three or more agents, and refers to the possibility of a joint specification instructing some agent to make a decision based on the state of a private sub-interaction in which that agent is not involved. Joint protocol specifications that exhibit blindness problems are often difficult to implement in practice without violating the privacy of the conversations involved.

Three examples of blindness are shown in figure 1. Each of these joint PSC representations is deemed unacceptable, since it is not possible to produce agent-centred representations that correspond exactly to the joint representations.

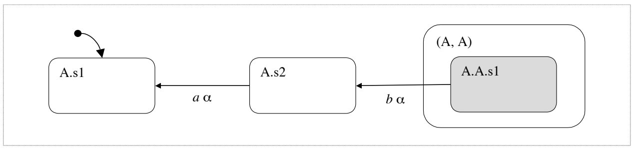

Figure 2: A simple recursive joint PSC description.

2.2 Verifying Realizablility in PSCs

In essence, the blindness problem illustrated in figure 1 arises when a protocol specifies that a message may be sent by some agent while the protocol is in a state which may have been preceded by a transition event in which that agent was not involved[2]. Since this is the case, realizability of joint protocol definitions produced using PSCs can be checked by annotating every state with a set of valid senders, defined as the intersection of the set of agents involved in each transition event that leads to that state, and the set of agents that are involved in the interaction in that state, as specified using braces as part of the state label.

We can say that a joint protocol specification represented by a joint PSC is realizable, in that it does not display blindness, if and only if the sender of any message that forms part of some transition event is a member of the set of valid senders associated with the state from which the relevant transition emanates. The set of valid senders associated with any state can be computed by computing the transitive closure of a state hierarchy function, and taking the intersection of the sets of agents associated with each member of the resulting set. This can be computed in polynomial time. Since the set of senders of any message contained within a given event label can be computed in linear time, the realizability of the protocol represented by any given joint PSC can be computed in polynomial time.

3. REACHABILITY ANALYSIS

Reachability analysis provides a basis for the verification of a number of properties, such as safety and liveness properties, guaranteed termination, and atomicity properties. In our discussion we distinguish between diagram states and protocol states. We use the term *diagram state* to refer to any atomic state that is represented explicitly in the abstract PSC representation of an interaction protocol. We use the term *protocol state* to refer to a global state of the protocol. A recursive protocol specification expressed using a PSC will have a finite number of diagram states, while the set of protocol states may be unbounded.

Reachability analysis in joint PSC representations can be performed via their conversion into Recursive State Machines, or RSMs, discussed in [1]. It is possible to directly translate joint PSC representations into RSMs in linear time, and, if this is done in such a way that the resulting RSM has a single initial protocol state, this makes it possible to calculate the set of diagram states that is reachable from the initial protocol state, using the algorithm described in [1].

Reachability analysis in agent-centred PSC representations is undecidable for full PSCs, since they are sufficiently expressive to represent unbounded buffers. The problem of determining reachability in systems of communicating finite state automata with unbounded buffers was shown to be undecidable in [2]. If we consider a weaker variant of agent-centred PSCs in which finitely bounded buffers are assumed, reachability analysis is decidable if a maximum of one agent's representation of the protocol is defined recursively. This follows from the fact that reachability analysis in systems of communicating pushdown automata has been shown to be undecidable, as discussed in [1].

[2]We say that an agent is 'involved' in some message if it is a sender or recipient of that message.

4. DISCUSSION AND RELATED WORK

In this paper we have discussed the verification of two key properties of joint specifications of agent interaction protocols expressed using Propositional Statecharts. We have briefly described a technique for verifying realizability, and have also argued that the verification of reachability is decidable via translation to Recursive State Machines. We have focused on the verification of properties of joint specifications of agent interaction protocols expressed using Propositional Statecharts. Due to space restrictions the description presented here has been necessarily brief. However, a more comprehensive treatment will be provided in a forthcoming doctoral thesis by the first author of this paper.

Realizability in open systems has been studied in a number of areas, such as the analysis of web service compositions [6], [3]. The approach presented here differs from that put forward in [6], [3] in that it extends to recursively defined infinite state specifications as well as finite state specifications.

The most common approach to reachability analysis in interaction protocols for multi-agent systems to date has been through the use of finite state model checking techniques, which are not applicable to the infinite state systems that can be described using joint Propositional Statechart specifications. In light of this, the work presented here is more closely related to work on infinite state verification, discussed in [1].

5. REFERENCES

[1] Alur, R., Benedikt, M., Etessami, K., Godefroid, P., Reps, T., and Yannakakis. M., *Analysis of Recursive State Machines*, in ACM Trans. on Programming Languages and Systems, 27(4), 786-818, 2005.

[2] Brand, D. and Zafiropulo, P., *On Communicating Finite-State Machines*. J. ACM, 30(2):323-342, 1983.

[3] Desai, N. and Singh, M. P., *Checking Enactability of Business Protocol Specifications*, AAAI Conference on Artificial Intelligence (AAAI), 2008.

[4] Dunn-Davies, H.R., Cunningham, R.J. and Paurobally, S. *Modularity and Composition in Propositonal Statecharts*, In Proceedings of Diagrams 2006, LNCS 4045, Springer, 98-103, 2006.

[5] Dunn-Davies, H.R., Cunningham, R.J. and Paurobally, S. *Deriving Agent-Centred Representations of Protocols Described Using Propositional Statecharts*, In Proceedings of AAMAS 2007, ACM, 2007.

[6] Fu, X., Bultan, T, and Su, J. *Realizability of Conversation Protocols With Message Contents*, International Journal of Web Services Research, 2(4), 68-93, 2005.