Logic-based Agent Communication Protocols

Ulle Endriss[1], Nicolas Maudet[2], Fariba Sadri[1], and Francesca Toni[1,3]

[1] Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2AZ (UK)
Email: {ue,fs,ft}@doc.ic.ac.uk
[2] LAMSADE, Université Paris 9 Dauphine
75775 Paris Cedex 16 (France)
Email: maudet@lamsade.dauphine.fr
[3] Dipartimento di Informatica, Università di Pisa
Via F. Buonarroti 2, 56127 Pisa (Italy)
Email: toni@di.unipi.it

Abstract. An agent communication protocol specifies the rules of interaction governing a dialogue between agents in a multiagent system. In non-cooperative interactions (such as negotiation dialogues) occurring in open societies, the problem of checking an agent's conformance to such a protocol is a central issue. We identify different levels of conformance (weak, exhaustive, and robust conformance) and explore, for a specific class of logic-based agents and an appropriate class of protocols, how to check an agent's conformance to a protocol *a priori*, purely on the basis of the agent's specification.

1 Introduction

Protocols play a central role in agent communication. A protocol specifies the rules of interaction between two or more communicating agents by restricting the range of allowed follow-up utterances for each agent at any stage during a communicative interaction (dialogue). Such a protocol may be imposed by the designer of a particular system or it may have been agreed upon by the agents taking part in a particular communicative interaction before that interaction takes place.

Protocol are *public*, i.e. they are (at least in principle) known to all participating agents (and possibly also to any outside observers). As several authors have pointed out, some form of public protocol is a necessary requirement for the definition of a suitable semantics of an agent communication language [13, 16]. Without public conventions (as specified by a protocol), it would not be possible to assign meaning to an agent's utterances.

This *"conventionalist"* view stands in contrast to the *"mentalistic"* approach taken, for instance, in the definition of FIPA-ACL [8], where the legality of utterances is specified in terms of the mental states of the agents participating in an interaction. This is not to say that an agent's mental state is not relevant to communication. On the contrary, an agent's goals and beliefs will strongly influence that agent's communicative behaviour. However, these mental attitudes should not have to be taken into account when we define what constitutes a *legal* utterance at a given point in time. We therefore distinguish an agent's communication *strategy*, which may be *private* and will be determined by the agent's goals and beliefs, from the public protocol which lays down the conventions of communication in a given multiagent system in terms of publicly observable events (i.e., in particular, previous utterances) alone.

By the very nature of protocols as public conventions, it is desirable to use a formal language to represent them. In particular, when used in connection with agents that are specified or even implemented using some form of executable logic, a logic-based representation language for communication protocols has many advantages. This paper summarises some of our recent work in the field of logic-based agent communication protocols, which has been initiated in [4] and further developed in [5]. It extends [4, 5] in that it discusses possible avenues for future work and it provides a more detailed comparison between our approach and related work in the area. In particular, Sections 5 and 6 are new.

Paper overview. In Section 2, we discuss two options for representing interaction protocols: finite automata and our logic-based language. In Section 3 we then introduce three different levels of conformance to such a protocol: weak, exhaustive, and robust conformance. In Section 4 we show that our logic-based representation greatly facilities checking whether a given agent is guaranteed to always behave in conformance to a particular protocol. We briefly recall the definition of a class of logic-based agents introduced in [14] and present sufficient criteria for such agents to be either weakly or exhaustively conformant to a protocol. Section 5 discusses potential generalisations of our protocol language and Section 6 discusses related work. Section 7 concludes.

2 Representing Agent Communication Protocols

In this section we present a logic-based representation formalism for a simple yet expressive class of interaction protocols. We assume some restrictions on the kind of interactions that we want to model. The dialogues we consider only involve *two agents* which *sequentially alternate* dialogue moves (utterances). These restrictions (notably avoiding concurrency) allow us to concentrate on a particular class of protocols, namely those representable by means of *deterministic finite automata* (DFAs), of which there are numerous examples in the literature.

Deterministic Finite Automata-based protocols. A DFA consists of (i) a set of states (including an initial state and a set of final states), (ii) an input alphabet, and (iii) a transition function δ which maps pairs of states and elements of the input alphabet to states [12]. In the context of communication protocols, elements of the input alphabet are dialogue moves and states are the possible stages of the interaction.

A protocol based on such a DFA representation determines a class of well-formed dialogues where each and every dialogue move is a *legal* continuation of the interaction that has taken place so far:

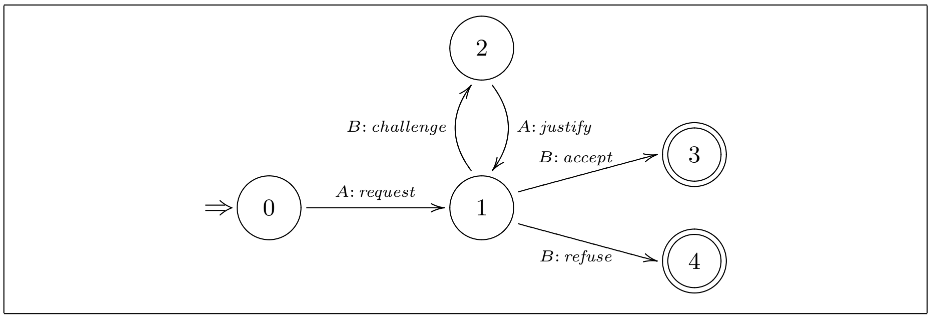

Fig. 1. A simple negotiation protocol

> *Given a protocol based on a DFA with transition function δ, a dialogue move P is a legal continuation (or simply a legal move) with respect to a state S iff there exists a state S' such that $S' = \delta(S, P)$.*

Example. Fig. 1 shows a simple example of a DFA-based protocol which regulates a class of negotiation dialogues between two agents A and B. It specifies that after a *request* made by A (in an initial state 0), B may either *accept* that request, *refuse* it, or choose to *challenge* it. Either one of the first two options would end the negotiation dialogue (bringing the dialogue into a final state). In case agent B decides to *challenge*, agent A has to reply with a *justify* move, which takes the dialogue back to the state where B can either *accept*, *refuse*, or *challenge*.

Logic-based protocols. Protocols such as that of Fig. 1 can alternatively be represented as sets of if-then-rules which specify the set of correct responses for a particular incoming dialogue move. For example, to express that agent B could react to a *request* move sent by A either by accepting, refusing, or challenging the request, we may use the following if-then rule:

$$\begin{aligned} \mathit{tell}(A, B, \mathit{request}, D, T) \Rightarrow\ & \mathit{tell}(B, A, \mathit{accept}, D, T{+}1) \vee \\ & \mathit{tell}(B, A, \mathit{refuse}, D, T{+}1) \vee \\ & \mathit{tell}(B, A, \mathit{challenge}, D, T{+}1) \end{aligned}$$

Here, the variable B refers to the name of the agent whose communicative behaviour is restricted by the rule and A represents the agent sending a message to B. Note that variables in the above formula are implicitly universally quantified over the whole formula. [4] In general, in this logic-based representation, dialogue moves are instances of the following schema:

$$\mathit{tell}(X, Y, \mathit{Subject}, D, T)$$

[4] In our chosen representation, in cases where there are variables that only appear on the righthand side of an implication, these variables are understood to be existentially quantified.

Here, X is the utterer, Y is the receiver ($X \neq Y$), D is the identifier of the dialogue, [5] and T the time when the move is uttered. *Subject* is the type of the dialogue move, i.e. a performative (such as *request*) of the communication language, possibly together with a content (as in $\mathit{request}(\mathit{item}_{17})$). For most of this paper, we are going to omit the content of dialogue moves, as it is usually not relevant to the definition of legality conditions for automata-based protocols and similar formalisms. Also, we shall mostly use the abbreviated form $P(T)$ for dialogue moves (where P stands for the performative of the respective move and T stands for the time of the move), thereby omitting the parameters not relevant to our discussion. For example, our earlier concrete rule could be represented in short as

$$\begin{aligned} \mathit{request}(T) \Rightarrow\ & \mathit{accept}(T{+}1) \vee \\ & \mathit{refuse}(T{+}1) \vee \\ & \mathit{challenge}(T{+}1) \end{aligned}$$

For the sake of simplicity, we will assume that the start of the protocol is triggered by some external event *START*—it is possible to conceive this as the result of some meta-level negotiation process to agree on a particular protocol. The start signal $\mathit{START}(X, Y, D, T)$ is sent by the system to agent Y to sanction at time T the beginning of a dialogue with identifier D amongst agent Y and agent X. We will assume that the system sends such a signal exactly once and exactly to one agent during a dialogue. We will also assume that each time this signal is sent to an agent, it has a new dialogue identifier. Similarly, a dialogue ends once one of the agents sends the signal *STOP* to the system. $\mathit{STOP}(X, Y, D, T)$ is sent by agent X to the system at time T to sanction the end of a dialogue with identifier D between X and Y. Dialogue inputs for an agent are either dialogue moves sent by other agents or a *START* signal sent by the system.

Going back to the example of Fig. 1, we observe that this automaton in fact represents two subprotocols, one for the *initiator* of a dialogue, and one for its *partner* (naturally, each agent can serve as initiator and as partner of a dialogue, at different times). We will refer to these two subprotocols as $\mathcal{P}_i$ and $\mathcal{P}_p$. They can be translated into a set (composed of two subsets) of if-then-rules. Here is the subprotocol for the initiator:

$$\mathcal{P}_i : \begin{cases} \mathit{START}(X, Y, D, T) & \Rightarrow \mathit{tell}(Y, X, \mathit{request}, D, T{+}1) \\ \mathit{tell}(X, Y, \mathit{accept}, D, T) & \Rightarrow \mathit{STOP}(Y, X, D, T{+}1) \\ \mathit{tell}(X, Y, \mathit{refuse}, D, T) & \Rightarrow \mathit{STOP}(Y, X, D, T{+}1) \\ \mathit{tell}(X, Y, \mathit{challenge}, D, T) & \Rightarrow \mathit{tell}(Y, X, \mathit{justify}, D, T{+}1) \end{cases}$$

Note that agent Y does not have any choices when replying to messages received from agent X (at least not as far as the performatives of the dialogue moves are concerned). Also note that agent Y is responsible for transmitting the *STOP* signal to the system after having received either one of the terminating moves

[5] In general, the identifier might be a function of a protocol name.

from X. The subprotocol for the partner in a dialogue consists of two rules:

$$\mathcal{P}_p : \begin{cases} \mathit{tell}(X, Y, \mathit{request}, D, T) \Rightarrow \mathit{tell}(Y, X, \mathit{accept}, D, T{+}1) \vee \\ \qquad\qquad\qquad\qquad\qquad\quad \mathit{tell}(Y, X, \mathit{refuse}, D, T{+}1) \vee \\ \qquad\qquad\qquad\qquad\qquad\quad \mathit{tell}(Y, X, \mathit{challenge}, D, T{+}1) \\ \mathit{tell}(X, Y, \mathit{justify}, D, T) \ \Rightarrow \mathit{tell}(Y, X, \mathit{accept}, D, T{+}1) \vee \\ \qquad\qquad\qquad\qquad\qquad\quad \mathit{tell}(Y, X, \mathit{refuse}, D, T{+}1) \vee \\ \qquad\qquad\qquad\qquad\qquad\quad \mathit{tell}(Y, X, \mathit{challenge}, D, T{+}1) \end{cases}$$

Note that agent Y have multiple choices when replying to messages received from agent X.

Shallowness. In our example we have simply translated an automata-based protocol into if-then-rules where we have a single performative on the lefthand side. We call protocols that permit such a straightforward translation *shallow*. Shallow protocols correspond to DFAs where it is possible to determine the next state of the dialogue on the sole basis of the previous move. Of course, this is not always the case for all protocols, since in some protocols it may be necessary to refer to the current state of the dialogue to determine the new state (think of two transitions with the same label leaving two different states and leading to two different states).

In principle, any automata-based protocol can be transformed into a protocol that is shallow in this sense (by simply renaming any duplicate transitions). In fact, many of the automata-based protocols proposed in the multiagent systems literature happen to be shallow already or could at least be made shallow by renaming only a small number of transitions.

Well-formedness requirements. In the light of the above remarks, we will generally represent shallow protocols as two sets of rules of the following form:

$$P(T) \ \Rightarrow \ P'_1(T{+}1) \vee P'_2(T{+}1) \vee \cdots \vee P'_k(T{+}1)$$

We will call these rules *protocol rules*. The righthand side of a protocol rule defines the possible continuations with respect to the protocol after the input P (which we will sometimes refer to as the *trigger* of the rule). The set of all triggers appearing in a subprotocol for a given agent is called the set of *expected inputs* for that agent. To ensure that this protocol is *well-formed* we will require that the two sets of rules meet a number of requirements (R1–R5):

- (R1, *initiation*): there is at least one rule with START on the lefthand side in the protocol, and START may never occur on the righthand side of a rule;
- (R2, *matching*): any dialogue move except STOP occurring on the righthand side of one subprotocol also occurs on the lefthand side of the other, and vice versa;
- (R3, *non-concurrency*): every subprotocol includes the following additional rule to avoid concurrent moves ($\bot$ stands for $false$):

$$\mathit{tell}(X, Y, S_1, T, D) \wedge \mathit{tell}(X, Y, S_2, T, D) \wedge S_1 \neq S_2 \Rightarrow \bot;$$

- (R4, *alternating*): for each rule occurring in a subprotocol, if X is the receiver and Y the utterer of the dialogue move occurring on the lefthand side, it must be the case that X is the utterer and Y the receiver of every dialogue move occurring on the righthand side (except for $START$ and $STOP$);
- (R5, *distinct triggers*): in each subprotocol, all dialogue moves occurring on the lefthand side of the rules are distinct from each other.

We note here that these are very simple requirements. Any protocol that is well-formed in this sense will provide a *complete* description of what constitutes a sequence of legal dialogue moves. R2 (matching) is the central requirement here; it ensures that for any move that is itself a legal continuation of the dialogue that has taken place so far, there will be a protocol rule that determines the range of legal follow-ups for that move. Ensuring this property in non-shallow protocols is more complicated as we shall see in Section 5.

3 Levels of Conformance

Broadly speaking, an agent is *conformant* to a given protocol if its behaviour is legal with respect to that protocol. We have found it useful to distinguish three *levels* of conformance to a protocol, which we are going to discuss next. Note that we are going to define these notions on the basis of the *observable* conversational behaviour of the agents (i.e. what they utter or not) alone, without making further assumptions on how they actually come to generate these utterances.

Weak conformance. We start with the notion of *weak conformance:*

> *An agent is weakly conformant to a protocol $\mathcal{P}$ iff it never utters any dialogue move which is not a legal continuation (with respect to $\mathcal{P}$) of any state of the dialogue the agent might be in.*

It is clear that any application governed by a protocol at least requires the level of weak conformance—otherwise it would not make sense to define a protocol in the first place. This is true at least if, as in this paper, we perceive protocols as (syntactic) rules that define the legality of an utterance as a follow-up in a given dialogue. If we adopt a broader notion of protocols, however, levels of conformance that are less restrictive than our weak conformance may also be considered. Yolum and Singh [18], for instance, advocate a *flexible* approach to interaction protocols in which agents may skip steps in a protocol as long as this does not render the interaction as a whole meaningless. Such protocols cannot be specified purely syntactically any more, but have to capture the "intrinsic meanings of actions" [18] for us to be able to decide which "shortcuts" are admissible and which are not.

Exhaustive conformance. The notion of weak conformance captures that the agent does *not* utter any illegal moves, but does not actually require that the agent utters any dialogue move at all. For interactions where "silent moves" are undesirable, a stronger version of conformance is usually required. We make this idea precise with the notion of *exhaustive conformance:*

> *An agent is exhaustively conformant to a protocol $\mathcal{P}$ iff it is weakly conformant to $\mathcal{P}$ and it will utter at least one dialogue move which is a legal continuation of any legal input of $\mathcal{P}$ it receives.*

Exhaustive conformance is certainly what is intuitively expected in *most* interactions—it is indeed often preferred to avoid considering silent moves as part of a protocol, at least to avoid confusion with lost messages. One may then argue that exhaustive conformance should be the minimum requirement for any interaction.

We believe, however, it is worth making the distinction between weak and exhaustive conformance. The first reason is that there are examples where the lack of response can be considered to be part of the protocol. In such circumstances, it can be sufficient to design a weakly conformant agent, provided that silent moves will not have undesirable consequences. For instance, in a Dutch auction process "when there is no signal of acceptance from the other parties in the auction (other agents in the negotiation) the auctioneer makes a new offer which he believes more acceptable (by reducing the price). Here, because of the convention (protocol) under which the auction operates, a lack of response is sufficient feedback for the auctioneer to infer a lack of acceptance." [10]. In this case, the agent can safely be designed to react appropriately only to the proposals it is ready to accept. But if we consider recent argumentation-based protocols inspired by dialectical models it is sometimes assumed that "silence means consent" [2]. In this case, a lack of response can commit the receiver to some propositions—this is a typical case where it is crucial that agents are exhaustively conformant. The second reason for our distinction of weak and exhaustive conformance is that they are *conceptually* different since weak conformance only involves *not* uttering (any illegal moves), while exhaustive conformance involves uttering (some legal move). This implies substantially different approaches when the issues of checking and enforcing conformance are raised, as we shall see later.

Robust conformance. Another important problem of agent communication is the need to deal with illegal incoming messages, and to react appropriately to recover from such violations. For instance, any FIPA-compliant communicative agent has to integrate a performative `not-understood` as part of its language [8]. This motivates us to introduce the following notion of *robust conformance:*

> *An agent is robustly conformant to a protocol $\mathcal{P}$ iff it is exhaustively conformant to $\mathcal{P}$ and for any illegal input move received from the other agent it will utter a special dialogue move (such as* `not-understood`*) indicating this violation.*

Robust conformance goes a step further than exhaustive conformance since it requires that an appropriate response is uttered also in reply to illegal input moves. Technically, this necessitates that the agent is able to identify the legality of an incoming dialogue move, i.e. it needs to be able to check conformance with respect to the *other* agent's subprotocol.

Note also that in the case where all agents in the society are known to be weakly conformant, it is theoretically unnecessary to deal with robust conformance (since no agent will ever utter an illegal move). The same applies to systems that are "policed" in the sense that messages not conforming to the protocol will simply not be delivered to the intended recipient. Such an assumption would, however, somewhat contradict the "spirit" of an open society. We should also point out that in dialogues with a very high contingent of illegal utterances the additional `not-understood` moves may in fact burden communication channels unnecessarily, and, therefore, simply ignoring illegal moves would in fact be a better strategy.

4 Checking Conformance

When *checking* an agent's conformance to a publicly agreed interaction protocol we can distinguish two cases: checking conformance *at runtime* and checking conformance *a priori*.[6] The former means checking the legality of the moves as they occur in a dialogue. This would enable a society of agents or a particular agent to determine the legality of an observed dialogue move. Checking conformance *a priori* means checking the legality of an agent's communicative behaviour on the basis of its specification. In other words, *a priori* conformance allows us to guarantee in advance that a computee will be conformant to a given protocol.

In general, checking *a priori* whether an agent will always behave in conformance to a given set of protocols is difficult, if not impossible. Firstly, the *privacy* requirement of a society of agents makes it problematic for the society to access the agent's private specification, and secondly the *complexity* of the specifications makes it hard—even when access to that specification is granted (for the agent itself, for instance)—to actually decide whether the agent will be conformant or not. In particular, the behaviour of the agent will typically depend on some hardly tractable notions, such as beliefs and intentions. As we shall see in this section, for a particular class of logic-based agents and for our shallow protocols we can overcome these difficulties, at least in the case of weak conformance. We are also going to discuss how to extend these results to checking exhaustive conformance, although—as far as our privacy requirements are concerned—our results will necessarily be less satisfying in this case.

Logic-based agents. We are now going to consider the case of a specific class of agents based on abductive logic programming that have recently been used in the context of negotiation scenarios [14]. The communication *strategy* $\mathcal{S}$ of such an agent (which forms part of its *knowledge base* $\mathcal{K}$) is represented as a set of if-then rules of the following form:

$$P(T) \wedge C \Rightarrow P'(T{+}1)$$

[6] Guerin and Pitt [9] refer to the latter as *compliance at design time*.

On receiving dialogue move P at time T, [7] an agent implementing this rule would utter P' at time $T+1$, provided condition C is entailed by its (*private*) knowledge base. Again, variables are understood to be implicitly quantified in the same way as our protocol-rules. The dialogue moves P and P' will be based on the agent's communication language. Below, we refer to if-then rules as the above as *strategy rules.*

Response spaces. In preparation for defining a suitable criterion for guaranteed weak conformance to a given protocol, we introduce the notion of *response space* for a logic-based agent. Intuitively, the response space of an agent specifies the possible moves that the agent can make when using a given strategy $\mathcal{S}$, *without* considering the specific conditions relating to its private knowledge base. This abstraction from an agent's communicative behaviour is related to the idea of an agent automaton proposed by Singh [15].

The response space $\mathcal{S}^*$ of an agent with strategy $\mathcal{S}$ based on the communication language $\mathcal{L}$ is defined as follows:

$$\{P(T) \Rightarrow \bigvee\{P'(T{+}1) \mid [P(T) \wedge C \Rightarrow P'(T{+}1)] \in \mathcal{S}\} \mid P \in \mathcal{L}\} \text{ with } \bigvee\{\} = \bot$$

That is, the response space is, essentially, the set of protocol rules we get by first dropping all private conditions C and then conjoining implications with identical antecedents by collecting the corresponding consequents into a single disjunction. For example, the strategy

$$\begin{aligned} S = \{&\mathit{request}(T) \wedge \mathit{happy} \Rightarrow \mathit{accept}(T{+}1), \\ &\mathit{request}(T) \wedge \mathit{unhappy} \Rightarrow \mathit{refuse}(T{+}1)\} \end{aligned}$$

determines the following response space:

$$S^* = \{\mathit{request}(T) \Rightarrow \mathit{accept}(T{+}1) \vee \mathit{refuse}(T{+}1)\}$$

Checking weak conformance. We are now going to state a simple criterion that offers an elegant way of checking weak conformance *a priori* for a logic-based agent. In particular, it avoids dealing with the dialogue history, and it does not make any assumptions on the content of the agent's knowledge base (except to require that it is possible to extract the response space, as previously described). The following is a sufficient criterion for weak conformance:

> *An agent is weakly conformant to a protocol $\mathcal{P}$ whenever that protocol is a logical consequence of the agent's response space.*

This result is proved in [5]. Observe, however, this is not a necessary criterion for weak conformance, because, looking at the form of strategies, it is clear that *private* conditions may prevent the agent from uttering a particular dialogue move. In other words, it could be the case that $\mathcal{S}^*$ does not entail $\mathcal{P}$ but that the agent is still weakly conformant to $\mathcal{P}$ because of its specific knowledge base.

[7] As earlier in the paper, we represent dialogue moves in short simply by referring to the performative in the move and to the time of the move.

The above result shows that, in the case of weak conformance, it is possible to check conformance *a priori* by inspecting only a relatively small part of an agent's specification (namely, what we could call its "communication module" or communication strategy). In particular, we are *not* required to make any judgements based on the content of its (probably dynamically changing) knowledge base in general.

Checking exhaustive conformance. In the case of exhaustive conformance, the situation is rather different. To understand why, recall that as well as requiring weak conformance, exhaustive conformance requires the property of uttering at least one legal move for any legal input. The latter property, which we shall simply refer to as *exhaustiveness* (of an agent) may be considered independently from a particular protocol. In [14], for instance, the authors define a notion of exhaustiveness with respect to a given communication language (as being able to utter a response for any incoming move belonging to that language). Even for our agents, whose communicative behaviour is determined by if-then rules of the form $P(T) \wedge C \Rightarrow P'(T{+}1)$, it is not generally possible to guarantee exhaustiveness (be it with respect to a given protocol, language, or in general). We cannot generally ensure that one of these rules will indeed "fire" for an incoming move $P(T)$, because none of the additional conditions C may be entailed by the current state of the agent's knowledge base.

As shown in [3], one way of ensuring exhaustive conformance would be to rely on logical truths that are independent from the (possibly dynamic) knowledge base of the agent. For a strategy $\mathcal{S}$ and any performative P in a given communication language, let $\mathrm{COND}_{\mathcal{S}}(P)$ denote the disjunction of all the private conditions that appear in a strategy rule in $\mathcal{S}$ together with the trigger $P(T)$, i.e.:

$$\mathrm{COND}_{\mathcal{S}}(P) = \bigvee\{C \mid [P(T) \wedge C \Rightarrow P'(T{+}1)] \in \mathcal{S}\} \text{ with } \bigvee\{\,\} = \bot$$

Now, if $\mathrm{COND}_{\mathcal{S}}(P)$ is a tautology for every performative P appearing on the lefthand side of the relevant subprotocol of a protocol $\mathcal{P}$, then any agent implementing the strategy $\mathcal{S}$ is guaranteed to utter *some* move for any input expected in $\mathcal{P}$. Hence, we obtain a useful sufficient criterion for exhaustive conformance (again, with respect to our shallow protocols):

> *An agent with strategy $\mathcal{S}$ is exhaustively conformant to a protocol $\mathcal{P}$ whenever it is weakly conformant to $\mathcal{P}$ and* $\mathrm{COND}_{\mathcal{S}}(P)$ *is a tautology for every expected input P (for that agent, with respect to $\mathcal{P}$).*

Of course, generally speaking, checking this condition is an undecidable problem because verifying theoremhood in first-order logic is. In practice, however, we would not expect this to be an issue given the simplicity of typical cases. As an example, consider a protocol consisting of only the following rule stipulating that any request by another agent X should be either accepted or refused:

$$\mathit{request}(X, T) \Rightarrow \mathit{accept}(T{+}1) \vee \mathit{refuse}(T{+}1)$$

An agent may implement the following simple strategy:

$$\begin{aligned} \mathit{request}(X,T) \wedge \mathit{friend}(X) \quad &\Rightarrow \mathit{accept}(T+1) \\ \mathit{request}(X,T) \wedge \neg \mathit{friend}(X) &\Rightarrow \mathit{refuse}(T+1) \end{aligned}$$

The disjunction $\neg \mathit{friend}(X) \vee \mathit{friend}(X)$, with X being implicitly universally quantified, is a theorem. Hence, our agent would be exhaustively conformant (note that the agent is certainly going to be weakly conformant, because the protocol is a consequence of its response space—in fact, the two are actually identical here). A similar idea is also present in [14], although not in the context of issues pertaining to protocol conformance. Fulfilling the above criterion is not an unreasonable requirement for a well-designed communication strategy $\mathcal{S}$ that is intended to be used for interactions governed by a given protocol $\mathcal{P}$.

We continue our discussion of exhaustiveness by observing that, in cases where we can identify a *static* part of an agent's knowledge base (beyond the set of rules making up its communication strategy), we can give an even more general sufficiency criterion that guarantees exhaustive conformance:

> *An agent with strategy $\mathcal{S}$ is exhaustively conformant to a protocol $\mathcal{P}$ whenever it is weakly conformant to $\mathcal{P}$ and* $\mathrm{COND}_{\mathcal{S}}(P)$ *is a logical consequence of the static part of the agent's knowledge base for every expected input P.*

To illustrate the idea, we slightly change our earlier example and replace the agent's second strategy rule with the following strategy rule:

$$\mathit{request}(X,T) \wedge \mathit{enemy}(X) \Rightarrow \mathit{refuse}(T+1)$$

That is, our agent will refuse any request by X if it considers X to be an enemy. Now our first criterion does not apply anymore; we cannot ensure exhaustive conformance. However, if the agent's knowledge base includes a formula such as $\neg \mathit{enemy}(X) \Rightarrow \mathit{friend}(X)$, expressing that anyone who is not an enemy should be considered a friend, then we can show that $\mathit{friend}(X) \vee \mathit{enemy}(X)$ is a logical consequence of that knowledge base and, thereby, that our agent will be exhaustively conformant to the protocol. Note that this agent may generate two responses for a single input, namely in cases where both $\mathit{friend}(X)$ and $\mathit{enemy}(X)$ are true, which would conflict with the non-concurrency requirement of our protocols (see well-formedness requirement R3).

Enforcing conformance. Finally, even when an agent cannot be shown to be conformant *a priori*, it may still be possible to constrain its behaviour at runtime by simply forcing it to comply to the rules of the protocol. The problem of *enforcing* conformance (referred to as *regimentation* by Jones and Sergot [11]) is then to try to find easy (and hopefully automatic) ways to ensure that an agent will always be conformant to a given protocol. As shown in [5], for any shallow protocol $\mathcal{P}$, a logic-based agent generating its dialogue moves from a knowledge base of the form $\mathcal{K} \cup \mathcal{P}$ will be weakly conformant to $\mathcal{P}$.

That is, the agent could simply "download" the appropriate protocol when entering a society and thereby guarantee conformance (and avoid possible penalties) *without* requiring any additional reasoning machinery. The intuition behind the proof of this result is that the additional protocol rules given by $\mathcal{P}$ (together with the non-concurrency rule of well-formedness requirement R3) would render any branches in the agent's internal derivation tree corresponding to illegal dialogue moves inconsistent and thereby actively prevent the agent from uttering such moves.

5 Beyond Shallow Protocols

In this section, we are going to highlight a number of ways in which our logic-based representation language for protocols may be extended to describe a wider class of communication protocols.

Protocols with several triggers. Shallow protocols can be seen as a special case of what could be called *n-triggers protocols*, which can be represented by if-then rules whose lefthand side may refer to any of the n previous utterances (both the agent's and its partner's) rather than just the very last utterance. Such if-then rules (referred to below as *n-trigger protocol rules* have the form:

$$P_1(T-1) \vee \cdots \vee P_n(T-n) \;\Rightarrow\; P_1'(T) \vee \cdots \vee P_k'(T)$$

For this class of protocols, we may or may not require a trigger to be present for *every* time point from $T-n$ to $T-1$. The latter seems to be more convenient for most examples. However, if the range of time points referred to in the list of triggers on the lefthand side is not always the same, then it becomes more difficult to formulate appropriate well-formedness conditions for this class of protocols. It is more complicated to determine if a given set of protocol rules is contradictory in the sense of forcing different responses in the same situation, and whether it is complete, in the sense of providing a rule for every possible state a dialogue following these rules may end up in.

Using n-trigger protocol rules, we can describe protocols where the last n moves matter. However, a DFA can express more complicated features. To illustrate this point, consider the automaton of Fig. 2. This is an example for a two phase negotiation protocol. Starting in state 0, agent A sends a first *request* which takes us into state 1. In this state, agent B may *challenge* the other agent, *accept* or *refuse* the request, or use its right to *veto*. In the case of a challenge, we enter a justification loop until either A *retracts* the request or B stops challenging. In state 6 the situation is very similar to state 1, but in state 6 agent B is not allowed to veto , nor is it allowed to refuse the request (but it can still challenge or accept the request). So it is a bit of a gamble for B to veto A's request and ask for a new request by A, as B can never refuse or veto the new request. Now, after receiving either a *request* or a *justify* move, it is crucial to be able to determine whether one is in state 1 or in state 6, because these states are different, as we have just seen. Intuitively, it is easy to see that in

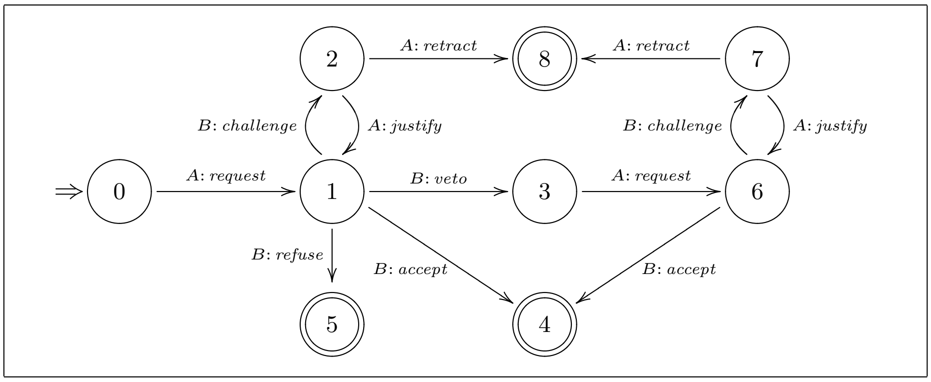

Fig. 2. A protocol that is problematic for n-triggers rules

order to be in the state 6, B must have sent a *veto* move at some point during the dialogue. But because of the loops, it is not possible to specify when this dialogue move has occurred. So, this protocol cannot be represented as a set of n-trigger protocol rules.

If we allow for arbitrary time references (rather than merely specific time points in the past) together with temporal constraints on the lefthand side of a rule then we *can* express that *veto* must have occurred at some point. In such an enriched language, we would be able to write protocol rules for state 6:

$$\begin{array}{rcl} \mathit{request}(T-1) \wedge \mathit{veto}(T') \wedge (T' < T) & \Rightarrow & \mathit{accept}(T) \vee \mathit{challenge}(T) \\ \mathit{justify}(T-1) \wedge \mathit{veto}(T') \wedge (T' < T) & \Rightarrow & \mathit{accept}(T) \vee \mathit{challenge}(T) \end{array}$$

However, writing similar rules for state 1 is still not possible, as we cannot express that *veto* has *not* occurred in the past.

Of course, we could further enrich our language and allow for negation and explicit quantification in protocol constraints to also capture this kind of situation. However, the more we enrich our language the more difficult it will be to actually work with protocols expressed in that language. One major problem would be to formulate appropriate well-formedness conditions to ensure that protocols are non-contradictory and complete.

Logical state-based representation of DFAs. An alternative avenue of research would be to adopt a state-based representation in logic, encoding straightforwardly DFAs into two kinds of rules: *state maintenance rules* representing the transition function (that is, given a state and a move, determining the next state), of the form

$$\mathit{state}(N,T) \wedge P(T) \Rightarrow \mathit{state}(N',T+1)$$

and *legality rules* determining the set of legal continuations from this state of the dialogue, of the form

$$\mathit{state}(N,T) \Rightarrow \bigvee_{i \in I} P_i(T+1)$$

Conceptually however, if we want to be able to check conformance using the techniques described earlier in this paper, all agents should be required to express their own communication strategies in terms of the same DFA. This, we believe, is quite a strong restriction that we would like to avoid in open agent societies (in comparison, the only assumption so far has been that strategy rules refer at least to the latest move on the lefthand side of the protocol rules).

Other extensions. The protocols we have discussed so far all refer only to the performatives (such as *request*) of utterances when evaluating legality conditions. In some situations, we may also wish to refer to the *content* of an utterance (for instance, to prevent an agent from proposing a price lower than the previous proposed price in the course of an auction, to deal with deadlines, etc.). This content is expressed in a content language which may be infinite. To permit explicit reference to this content, protocols should be augmented with data structures allowing to keep track of this information (or at least of those contents deemed relevant to contrain the legal follow-ups of the interaction). Examples of such protocols can be found in argumentation-based approach to agent interaction [2], where it is often assumed that agents have access to so-called *commitment stores* to record each others arguments throughout the dialogue. In general, such protocols will be more expressive than the protocols described via DFAs. [8]

In [7], a number of abstract models for protocols where content items influence the range of legal follow-ups have been studied in terms of abstract machine models (such as pushdown automata). It would be interesting to combine these ideas with our approach and to design logic-based protocol representation languages that allow for the explicit reference to an utterance's content.

6 Related Work and Discussion

In this section we briefly discuss two related approaches to logic-based protocols for agent communication, namely the "social integrity constraints" of Alberti *et al.* [1] and the set-based protocol description language for the specification of logic-based electronic institutions introduced by Vasconcelos [17].

Social integrity constraints. Alberti *et al.* [1] put forward a logic-based representation formalism for communication protocols that is closely related to ours. Their "social integrity constraints" are similar to our protocol rules; however, they explicitly introduce operators to refer to events (such as agents uttering particular dialogue moves) that *happen* in an agent society and those that are *expected* to happen in the future (or, indeed, that should have happened in the past). In our system, these notions are implicit: the lefthand side of a protocol rule such as $\mathit{request}(T) \Rightarrow \mathit{accept}(T{+}1) \vee \mathit{refuse}(T{+}1)$ refers to events that have

[8] Note however that adding only a *set* recording contents of a *finite* language is not more expressive than a DFA (even if, practically, it can soon turn out to be tedious to use the automaton-like design).

just happened, while those on the righthand side specify the expectations put on the agent whose turn it is next.

Another difference is that the integrity constraints of Alberti *et al.* may include temporal constraints over the time parameters of the social events (i.e. dialogue moves) occurring in a rule. This allows for the representation of a wider class of protocols than just shallow protocols. On the downside, this added expressive power makes the design of well-formed protocols more difficult a task. Given a set of protocol rules in a rich description language, how do we check whether this protocol really covers every possible dialogue state?[9]

Similarly, the more expressive a protocol representation language, the more difficult will it be to check an agent's conformance to such a protocol. While in the case of automata-based protocols (and certainly shallow protocols) checking conformance *at runtime* is essentially a trivial problem, this becomes a major issue for systems where the legality of a given move at a given time cannot be decided once and for all at the time it occurs. Indeed, much of [1] is devoted to this issue for the case of social integrity constraints. Of course, checking conformance *a priori* is considerably more difficult. It firstly requires an abstraction from the agent's specification, expressed in the same language used to express protocols (in the case of our system, this abstraction is given by the very simple notion of response space). This abstraction then needs to be related to the actual protocol in order to define a suitable criterion for guaranteed conformance.

Logic-based electronic institutions. Vasconcelos [17] puts forward another logic-based system for specifying protocols in the context of developing electronic institutions [6]. Electronic institutions are an attempt to provide an "institutional" framework for the development of open agent societies. This includes, in particular, the provision of institutional rules that govern dialogue between agents inhabiting such a society, i.e. communication protocols.

The language proposed in [17] is based on first-order logic, but enriched with sets. This allows for the representation of relevant events in the past, which may influence the legality of follow-up moves. From a computation-theoretic point of view, Vasconcelos'system is related to the class of "protocols with a blackboard" described in [7]. As discussed before, protocols of this class extend automata-based protocols by allowing for utterances to be stored in a set (the blackboard) and for legality conditions that make reference to the elements of that set.

7 Conclusion

We have discussed conformance as the basic notion of ensuring that the behaviour of an agent is adapted to a public protocol regulating the interaction in

[9] It should be noted that, in fact, this is only a problematic issue if one requires a protocol to provide a *complete* specification of what constitutes legal behaviour in a communicative interaction. While we take the view that a protocol should provide such a complete specification in order to facilitate a complete interpretation of the dialogues actually taking place, Alberti *et al.* "tend to constrain agents' interaction a little as possible, only when needed" [1].

a multiagent system. Our approach starts from on an alternative representation formalism for communication protocols based on if-then-rules for the kinds of protocols that can be represented as deterministic finite automata. In particular, we have restricted ourselves to a class of protocols where it is not necessary to consider the history of the dialogue besides the latest move to determine the possible legal dialogue continuations (shallowness). We have then discussed the distinction of three levels of conformance: weak conformance which requires that an agent never makes an illegal move, exhaustive conformance which in addition requires an agent to actually make a (legal) move whenever it is his turn in a dialogue, and robust conformance which also requires an appropriate reaction to illegal incoming moves. In the case of weak and exhaustive conformance, we have provided sufficient criteria that can be used to determine whether an agent can be guaranteed to behave in conformance to a given protocol.

Competence. In this paper we have studied the concept of conformance to a communication protocol, which is undoubtedly one of the very central notion to be considered when working with protocols. However, the ability to merely *conform* to a protocol is not not sufficient to be a *competent* user of that protocol. Intuitively, we understand competence with respect to a protocol as the capacity of an agent to "deal adequately" with that protocol. Take, for instance, our negotiation protocol shown in Figure 1. Now assume an agent (supposed to take the role of agent B in that protocol) is engaging in a dialogue regulated by this protocol using the following response space:

$$\begin{aligned}\mathcal{S}^* = \{&\mathit{request}(T) \Rightarrow \mathit{refuse}(T{+}1),\\ &\mathit{justify}(T) \Rightarrow \mathit{refuse}(T{+}1)\}\end{aligned}$$

Even if this agent was exhaustively conformant to the protocol, it would intuitively not be competent as it could never reach either state 2 or 3 (and consequently the interaction could never terminate with an accepted request). A notion of competence that takes into account such issues is studied in [3].

Acknowledgements. This research has been funded by the European Union as part of the SOCS project (IST-2001-32530). The last author has been partially supported by the MIUR (Italian Ministery of Eduction, University, and Research) programme "Rientro dei Cervelli".

References

1. M. Alberti, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Specification and Verification of Agent Interactions using Social Integrity Constraints. In W. van der Hoek et al., editors, *Proceedings of the Workshop on Logic and Communication in Multi-Agent Systems (LCMAS-2003)*, 2003.
2. L. Amgoud, N. Maudet, and S. Parsons. Modelling Dialogues using Argumentation. In *Proceedings of 4th International Conference on MultiAgent Systems (ICMAS-2000).* IEEE, 2000.

3. U. Endriss, W. Lu, N. Maudet, and K. Stathis. Competent Agents and Customising Protocols. In A. Omicini et al., editors, *Proceedings of the 4th International Workshop on Engineering Societies in the Agents World (ESAW-2003)*, 2003.
4. U. Endriss, N. Maudet, F. Sadri, and F. Toni. Aspects of Protocol Conformance in Inter-agent Dialogue. In J. S. Rosenschein et al., editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*. ACM Press, 2003. Extended Abstract.
5. U. Endriss, N. Maudet, F. Sadri, and F. Toni. Protocol Conformance for Logic-based Agents. In G. Gottlob and T. Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*. Morgan Kaufmann Publishers, 2003.
6. M. Esteva, J.-A. Rodriǵuez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the Formal Specification of Electronic Institutions. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce: The European AgentLink Perspective*. Springer-Verlag, 2001.
7. R. Fernández and U. Endriss. Abstract Models for Dialogue Protocols. In M. Marx, editor, *Proceedings of the 5th International Tbilisi Symposium on Language, Logic and Computation*, 2003.
8. Foundation for Intelligent Physical Agents (FIPA). *Communicative Act Library Specification*, 2002. http://www.fipa.org/specs/fipa00037/.
9. F. Guerin and J. Pitt. Guaranteeing Properties for E-commerce Systems. In J. Padget et al., editors, *Agent-Mediated Electronic Commerce IV: Designing Mechanisms and Systems*. Springer-Verlag, 2002.
10. N. Jennings, S. Parsons, P. Noriega, and C. Sierra. On Argumentation-based Negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems (IWMAS-1998)*, 1998.
11. A. Jones and M. Sergot. On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. In *Deontic Logic in Computer Science: Normative System Specification*. John Wiley and Sons, 1993.
12. H. R. Lewis and C. H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall International, 2nd edition, 1998.
13. J. Pitt and A. Mamdani. A Protocol-based Semantics for an Agent Communication Language. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-1999)*. Morgan Kaufmann Publishers, 1999.
14. F. Sadri, F. Toni, and P. Torroni. Dialogues for Negotiation: Agent Varieties and Dialogue Sequences. In *Proceedings of the 8th International Workshop on Agent Theories, Architectures and Languages (ATAL-2001)*. Springer-Verlag, 2001.
15. M. P. Singh. A Customizable Coordination Service for Autonomous Agents. In *Proceedings of the 4th International Workshop on Agent Theories, Architectures and Languages (ATAL-1997)*, 1997.
16. M. P. Singh. Agent Communication Languages: Rethinking the Principles. *IEEE Computer*, 31(12):40–47, 1998.
17. W. W. Vasconcelos. Expressive Global Protocols via Logic-based Electronic Institutions. In J. S. Rosenschein et al., editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*. ACM Press, 2003. Extended Abstract.
18. P. Yolum and M. P. Singh. Flexible Protocol Specification and Execution: Applying Event Calculus Planning Using Commitments. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*. ACM Press, 2002.