

Ensuring Consistency in the Joint Beliefs of Interacting Agents

Shamimabi Paurobally
University of Southampton
Southampton SO17 1BJ, UK.
sp@ecs.soton.ac.uk

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

Nicholas R. Jennings
University of Southampton
Southampton SO17 1BJ, UK.
nrj@ecs.soton.ac.uk

ABSTRACT

Agent interaction in realistic applications is subject to many forms of uncertainty – including *information and network uncertainty, trust of and conflicts with other participants, lack of stability in a deal and risks about agreements and commitments*. However, one of the most common forms of uncertainty occurs when a group has divergent beliefs about the interaction they are engaged in – some agents believe an agreement has been reached, while others believe it has been rejected or that they are still bargaining. Such misunderstandings can arise because of loss of network performance, spurious connections, message loss or delays. Against this background, this paper develops synchronisation protocols for a group of agents to attain the same beliefs about an interaction, *independent of the reliability of the underlying communication layer*. This paper includes and proves theorems about a group's mutual beliefs, on which the safety of an interaction relies. Specifically, protocols for message exchange and belief revision and the reasoning for reachability of states during interactions are presented. Each protocol is proved to show that an increasing level of mutual and consistent belief is reached, thereby guaranteeing an interaction's integrity.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed AI

General Terms

Reliability Languages Verification

Keywords

agent, belief, protocol, synchronisation, interaction state

1. INTRODUCTION

Social interactions, such as cooperation, coordination and negotiation, are a fundamental feature of multi-agent systems. They are enacted through a variety of interaction protocols, here regarded as the public rules or norms for

communications of the participants of a group when carrying out some social encounter. In this context, the protocol ensures that all group participants following it can expect certain responses from others and can coordinate meaningfully towards a goal. But in many cases it is not always clear what it means for a group of agents to follow a protocol. In particular, issues arise when unexpected events occur, for example agents do not comply with or misunderstand the interaction protocol [10], or the communication is faulty. To deal with such issues and the semantics of agent communications, we regard an interaction as a joint process between agents. The steps in such a joint process progress by virtue of the propositions believed by the group. For the purpose of reasoning about the beliefs of a group of agents, we consider the state of an interaction as entailed from the propositions believed by all the agents about that interaction and derivable from an interaction protocol. For example, let a common protocol P , for a joint negotiation between two agents X and Y , specify that after agent X has browsed a catalogue, it may make an offer which should be followed by an agreement or a rejection from Y . A state of negotiation such as *offered* means that both agents believe an action *offer* has been made and on reaching an agreement the state changes from *offered* to *agreed*. In this work, we view common beliefs as relating to the shared beliefs of a group of agents about each other, and joint beliefs as deriving from the union of all the individual beliefs of agents in a group [8]. We use *belief*, instead of *knowledge*, because the property of *knowledge* being true [8] makes it far harder to attain than belief in a practical context. Whereas knowledge must be true, beliefs only require consistency and we assume this in our solutions. For example, an agent X may believe an agreement has been reached with agent Y when in fact it has misunderstood or mis-implemented the protocol, or there has been a security breach with a malicious agent impersonating Y . When these uncertainties are compounded with network unreliability, it is easier for an agent to believe some state than to know that state.

While an interaction protocol may be considered as common belief in a group of agents, the participants' individual beliefs about the progress of a particular interaction are liable to differ between the time a message is sent and received. For example, a sender has added beliefs about the message it sent compared to another agent which has not received that message. Therefore, some degree of common belief, and consistency in joint beliefs about the interaction state, are necessary to safely progress in an interaction and to avoid contradictions and any confusion that may arise

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
Copyright 2003 ACM 1-58113-683-8/03/0007 ...\$5.00.

Interaction Protocols Speech–act like messages
Synchronisation Protocols Knowledge–based protocols for message exchange and belief revision
Communication Protocols Session, TCP/IP layers

Figure 1: Layers of Protocols

from a participant persisting with differing belief about the interaction state¹. The key issue here is that the interaction must not continue without ensuring that all agents believe the same state at some point. For example, if messages are lost in an interaction following the protocol P introduced above, then one agent may believe the state to be *agreed* while another believes it to be *browsed*. Such discrepancies may lead to disputes worsened when monetary, time or safety-critical information are involved.

Given these requirements, we develop *synchronisation protocols* that should be followed by all agents for message exchange and belief revision in an interaction. These protocols lie in a layer below interaction protocols but above communication (network) protocols (see figure 1). Such synchronisation protocols specify the steps for sending messages, acknowledgments and for belief update. Their aim is to produce a degree of shared belief, if not common belief, about the state of an interaction, sufficient to remove uncertainty about that state. Moreover, we prove that our synchronisation protocols allow a group to attain the same beliefs about an interaction state before each transition and that an interaction safely terminates with no uncertainty about its result. If the interaction has failed, then all agents believe so.

This paper advances the state of the art in multi-agent interactions; by developing an approach that ensures their safe progression through a number of identifiable states and termination in a consistent manner. Existing multi-agent system work in this area typically relegates this problem to the ISO transport layer and assumes that messages safely reach their destination [7], [11]. However, this is inappropriate in situations in which *an agent* has to choose compensating or alternative actions in cases of network lag or failure. For example, let an agent X send an agreement to n agents and then receive a confirmation from m agents ($m \leq n$). If any *one* of the n agent's confirmations are lost through the network, the group as a whole may have to coordinate to ratify X 's agreement and inform the others of the ratification, let alone deal with delayed receipt of messages. Thus, because perfect communication cannot be guaranteed in most multi-agent systems, the problem of reasoning about message exchange is not strictly a transport layer issue. On the other hand, the issue of sending messages and acknowledgements should not be dealt with at the interaction layer because at this level an agent does not want to concern itself with the intricacies behind message exchange. The interaction level should rather concentrate on higher-level reasoning. Consequently, we develop the concept of synchronisation protocols to lie between interaction and communication protocols. We reason about beliefs rather than knowledge because common knowledge can be impossible to reach when communication is imperfect [3], [8].

The next section discusses the beliefs of a group of in-

¹This issue of consistency is an important problem because unreliable communication is the norm in the communication networks and infrastructures in which agents are most likely to be deployed.

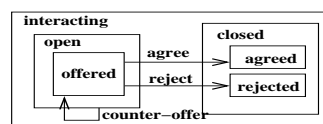


Figure 2: A Hierarchy of Negotiation States

teracting agents and proves basic theorems underlying our work. Section 3 presents our assumptions and definitions for synchronising the beliefs of a group. Then sections 4 and 5 specify synchronisation protocols for bilateral interactions, through statecharts [6], and an extension of Propositional Dynamic Logic [10]. Each protocol is proved to safely ensure both consistency and mutual beliefs during and at the end of an interaction. Section 6 details related work and section 7 concludes with future work.

2. AN INTERACTING GROUP'S BELIEFS

We distinguish between the state of an agent and the state of an interaction. The former remains private to that agent. In a shopping scenario, a retailer agent displays and trades its goods, but may not be aware of the state of browsing customers. On the other hand, the state of an interaction exists only by virtue of the shared beliefs of a group of interacting agents. The interaction state is a conjunction of propositions, may have a hierarchical structure and is specified by the protocol. When a state holds, any of its parent states also hold. Figure 2 shows a hierarchy of states where *rejected* and *agreed* are sub-states of *closed*. An *agree* action from an *offered* state triggers an *agreed* state. The state conveys information about the point an interaction has reached. It needs to be believed by all participants in a group, and changes depending on the actions of an agent. Partial information is available when each group member believes the same parent state holds e.g. *closed*, whereas more specific information is the belief that *agreed* or *rejected* holds.

2.1 Individual and Common Beliefs of a Group

Epistemic logic defines modal operators for knowledge and belief. It has been shown to provide a good basis for understanding the design and verification of protocols [5]. For the sake of conciseness and familiarity, this paper henceforth uses for group beliefs the operator symbols E and C normally denoting group and common knowledge (as in [3]). This is allowable because we are redefining the semantics of the operators to denote beliefs utilizing preconditions about knowledge operators. An axiomatic system for belief may be defined in terms of axioms for consistency and introspection [3], [8]. We assume that each agent in a group has such a system of beliefs and is aware that others do.

The formula $B_i\alpha$ is read as agent i believes α , $E_G\alpha$ is read as *everyone in a group of agents, G , believes α* and $C_G\alpha$ means α is common belief among the agents in G . It is conventional to express $C_G\alpha$ in terms of $E_G\alpha$, where $C_G\alpha$ is true if everyone in G believes α , everyone in G believes that everyone in G believes α and so forth [8]: ($C_G\alpha \equiv E_G\alpha \wedge E_G^2\alpha \wedge \dots \wedge E_G^m\alpha \wedge \dots$) where $E_G^1\alpha \equiv E_G\alpha$ and $E_G^{k+1}\alpha \equiv E_GE_G^k\alpha$, for $k > 0$. With this standard definition, certain properties of protocols can be proved.

We assume an interaction protocol is common belief in a group, where all agents are aware of permissible states and actions and believe that other participants have the same beliefs. Thus an agent X may believe that it follows a protocol and believes that others in the group both believe and

use the same protocol and believe what X believes about the protocol. The individual beliefs, B_i , of an agent i extend the common beliefs of the group and may include agent i 's beliefs about its interactions, environment and strategies. An agent revises its individual beliefs as the interaction progresses. Thus the individual beliefs of the agents in a group about the history and state of an interaction may differ between each other while a message is being relayed.

2.2 Consistency of Joint Beliefs

The *joint beliefs* of a group are the sentences that are consequences of the union of the individual beliefs of the agents. Thus the joint beliefs are extensions of the individual and common beliefs of a group. The joint beliefs relate to the beliefs of all the agents in a group and unless all agents publicise their beliefs, no agent is privy to the joint beliefs of the group. Given the individual beliefs of the agents differ, the joint beliefs of a group need not be completely consistent.

However consistency of the joint beliefs *about an interaction state* is needed before progressing to the next state. Consider a multi-lateral interaction between n agents, where the interaction state is *motioned* (after raising a motion). This entails that all n agents believe the state to be (*motioned* \wedge \neg *seconded*). An agent i sends a message m to second the motion to eventually trigger a *seconded* state. On sending m , agent i may believe the new interaction state is (*seconded* \wedge *motioned*). Agent i 's belief about the interaction state is inconsistent with the other participants. On p agents receiving m , they update their beliefs about the interaction state to (*seconded* \wedge *motioned*). Now $(p+1)$ agents beliefs are inconsistent with $(n-p-1)$ agents. This is as expected given delays in message transfer over networks. It is important though that those $(p+1)$ agents do not continue with the interaction with voting and agreement while some of the group still believe (*motioned* \wedge \neg *seconded*), since this would result in agents receiving unexpected messages. An agent still believing the state to be \neg *seconded* could then receive a message about the result of all votes, which should only follow after a *seconded* and *voting* state.

Thus we propose synchronisation protocols to ensure that all agents believe the interaction state α at world w before proceeding on to the next state at world w_1 , resulting in the formula $E_G\alpha$ holding in group G at world w . In fact an interaction state α emerges from it being a consistent and joint belief in a group at w .

Definition 1. A protocol is complete if all states are well-defined, that is either a state or its negation holds at any instance. A protocol is consistent if both a state and its negation cannot hold at any instance.

THEOREM 1. *A complete and consistent protocol allows a group of agents to attain consistency in their joint beliefs.*

PROOF. The proof of theorem 1 is a ramification of Robinson's Consistency theorem [2]. (See [9]). \square

Convention 1. The formula $E_G\alpha$ holds about interaction state α in group G before any further transition to a subsequent state β .

In addition to consistency in joint beliefs, safe termination of an interaction requires that the final state becomes a common belief of all agents. For example, if an interaction terminates in an *agreed* state, each agent has to believe that

everyone believes the same state and that everyone believes that it believes the state is *agreed*, etc. There is then no uncertainty about joint commitments if a commitment state like *agreed* is common belief. In such situations, an agent can fulfill its part of the bargain, believing that the other agents have the same beliefs as itself about the state. Ideally not only the terminal state but also the different states of an interaction should be common belief in a group. However in imperfect communication environments, achieving common beliefs about each state before carrying on to the next state would require an indefinite number of acknowledgements and is unfeasible. Instead we propose to settle for shared beliefs which bring about common belief as argued in the following section.

2.3 Shared Beliefs

While not achieving conventional common belief about the state of an interaction in a group, we endeavour to ensure an increasing level of mutual belief to remove uncertainty about an interaction's progress. Following convention 1, during an interaction, as long as the formula $E_G\alpha$ about interaction state α holds, the agents can continue onto the next state transition. Subsequent messages about successive state transitions implicitly increase the mutual beliefs about the state α being part of the history of the interaction. As an example, let an interaction perform the succession of states 1 to 4 corresponding to (*requested* \rightarrow *proposed* \rightarrow *offered* \rightarrow *agreed*). At step 2, before a *proposed* state, the formula E_G *requested* holds. At step 3, before an *offered* state, the formula $(E_G$ *proposed* \wedge $E_G^2\gamma$) holds where γ entails that *requested* was valid before the current state *proposed*. At step 4, before *agreed*, the formula $(E_G$ *offered* \wedge $E_G^2\rho \wedge E_G^3\gamma)$ holds where ρ entails that *proposed* was valid before the current state *offered*. After step 4, the state of interaction is *agreed* yielding the formula $(E_G$ *agreed* \wedge $E_G^2\sigma \wedge E_G^3\rho \wedge E_G^4\gamma)$ where σ entails *offered* was a preceding state. In this way, the history of an interaction can be derived from the interaction protocol and the interaction state.

Definition 2. The state α is *shared belief* if everyone believes the state α and believes that everyone believes so too - $(E_G\alpha \wedge E_G^2\alpha)$.

If *agreed* is a terminal state, then a group of agents must ensure not only E_G *agreed*, but also there is a degree of common belief about *agreed*. We settle for a group believing the first two terms of common beliefs which we call *shared belief* because shared belief about an interaction state allows the group to reach common belief, as in theorem 2.

2.4 Sufficiency of Shared Beliefs

Theorem 2 gives the conditions for deriving common belief about the state of an interaction in a group from shared beliefs. If each agent is aware of the other agents' reasoning system, then through positive and negative introspection about the possible interaction states, the group can reach common beliefs about these states.

THEOREM 2. *Let G be a group of agents and α be an interaction state. Assuming (1) the systems of beliefs of all agents are consistent and their consistency is a common belief in G , (2) the interaction and synchronisation protocols are common in G , then shared belief in G about α implies common belief in G about α .*

PROOF. It is sufficient to prove $(E_G\alpha \wedge E_G^2\alpha) \rightarrow E_G^3\alpha$ which results in $(E_G\alpha \wedge E_G^2\alpha) \rightarrow (E_G\alpha \wedge E_G^2\alpha \wedge \dots \wedge E_G^m\alpha \wedge$

...). Take $(E_G\alpha \wedge E_G^2\alpha)$ as equivalent to $E_G(\alpha \wedge E_G\alpha)$. Assume shared belief about α in G i.e. $(E_G\alpha \wedge E_G^2\alpha)$. Then $\forall i \in G \bullet B_i(\alpha \wedge E_G\alpha)$. Assume that agent i believes that the other agents j have the same reasoning system for beliefs as itself. Given that agent i also believes that every agent is following the same protocols, then everything that agent i believes has been achieved, it believes everyone believes too, from positive introspection. Therefore $\forall i, j \in G \bullet (B_i B_j(\alpha \wedge E_G\alpha))$ implying that $\forall i \in G \bullet (B_i E_G(\alpha \wedge E_G\alpha))$, that is $E_G^3\alpha$ (and whence by induction $C_G\alpha$). \square

Common belief about α is not guaranteed without assumption (1) in theorem 2, since an inconsistent common belief may lead a rational agent to disbelieve α . Because an interaction protocol defines only a finite number of states, positive and negative introspection about the end states is possible and does not require beliefs about interim states to be complete. In our case, an agent's introspection concerns only beliefs about what is being communicated in a specific interaction and more particularly termination. In the case of unsuccessful termination such as timeouts, negative introspection allows all agents to believe that not all agents believe the end state. A safe protocol thus allows all agents to believe any failure that occurs.

COROLLARY 1. *If the terminal state α is shared belief in a group G , then an interaction terminates without uncertainty.*

The corollary follows from the state of interaction being common belief, given the preconditions of theorem 2. Thus one advantage of reasoning about beliefs instead of knowledge is the ability to attain common beliefs about the (terminal) state of an interaction.

3. DEFINITIONS AND ASSUMPTIONS

In a perfect communication medium, where message exchange is error-free and instantaneous, belief revision after state transition messages can be instantaneous. In this case, the synchronisation protocol is straightforward. However this is not the case for time-constrained interactions in unreliable networks. To this end, we specify synchronisation protocols between two interacting agents to ensure their interaction progresses only with consistent joint beliefs about the interaction state. Our protocol allows a sender to eventually believe whether others have successfully received its message and receivers to believe whether their acknowledgments have been received. This enables each agent to believe the same interaction state before, during and after each state transition with an increasing degree of shared belief.

Several characteristics of a communication medium influence the design and choice of a synchronisation protocol. The interaction itself may be time-constrained or all the participants may benefit from unbounded time.

- *Perfect communication.* Message exchange is trivially delayed, error-free, not duplicated and never lost.
- *Imperfect communication but guaranteed receipt of message.* The network may delay, duplicate or corrupt messages but does not fail permanently so that a repeatedly sent message eventually reaches its destination. On not receiving an expected message, a receiver believes that either the message will eventually arrive or that the sender has crashed. A protocol using repeated messaging is suitable when the participants

have unbounded time or their deadline is far. However such protocols increase the risk of network bottlenecks.

- *Non-guaranteed receipt of messages.* In this case, a message may never get relayed. Non-receipt of an expected message can be for any reason where a sender or the network has failed.

A family of protocols are possible for synchronisation in bilateral and multi-lateral interactions depending on the complexity of solutions an agent chooses in dealing with communication failures. We provide synchronisation protocols for guaranteed and non-guaranteed receipt of messages in a bilateral interaction. These protocols trivially apply when communication is perfect.

3.1 Assumptions

This section provides reasonable assumptions in order for the synchronisation protocols to have the desired effect of ensuring the safety of an interaction. These assumptions are the foundations of our reasoning and proofs.

Assumption 1. The protocols (interaction, synchronisation and communication) are common beliefs. Before an interaction, all participants have agreed on which protocol to comply with. For example a market-place or an auctioneer broadcasts the types of auction it supports.

Assumption 2. Following assumption 1, an interaction is initiated in a commonly believed state. For example in an auction, if the overall parent state *auctioning* has sub-states *open* and *closed*, then the state of the auction before its start is \neg *auctioning* [10].

Assumption 3. To avoid infinite acknowledgments in our protocol, we assume that the communication layer informs a sender if it fails to deliver its message (usually 100% packet loss can be detected).

Assumption 4. All agents have perfect recall. An agent does not lose information erroneously about its beliefs, resulting in persistency of the states in an interaction.

3.2 Acknowledgments and Message Structure

We adopt a tuple for the structure for a message: (*message-number, sender, receiver, action, new-state*). A *message-number* associated with the sender avoids confusion in case of messages arriving in the wrong order or ghost messages. The fields *sender* and *receiver* are the identities of the sending and receiving agents respectively. A hash value can be used to detect whether a message has been corrupted by the network. The parameter *action* is a process from the sender for transition to the state *new-state* according to an interaction protocol. An *action* may be a FIPA-ACL or KQML [7] performative. Unlike standard approaches to agent communication where belief revision is specified at a higher level than the message structure, here the state of the interaction is part of the message. The first reason for this is that a protocol may allow an action to trigger two different states, especially with the nesting of interactions. Another reason is that the *action* may be a complex process involving several consecutive state changes.

We define two types of acknowledgment to a message: *implicit* and *explicit*. An *implicit acknowledgment* contains a performative for a state change, whereas an *explicit acknowledgment* is just an acknowledgment without any performative. An explicit acknowledgment includes an *ack* keyword

as an action instead of a performative. Let an agent a send a message, $m1$, containing a performative *offer* and a message number i , to agent b according to an interaction protocol P . If protocol P specifies that receiver b has to respond with a performative (*agree* or *reject*), then the next message, $m2$, from b contains one of these two performatives. Message $m2$ is also an implicit acknowledgment that b has received message $m1$. When a receives $m2$ and the message number is $(i+1)$, it can believe that b has received $m1$. Thus $m2$ is said to be an implicit acknowledgment for $m1$. Implicit acknowledgements reduce the redundancy occurring in protocols where only explicit acknowledgements can be sent after each message. Implicit acknowledgements are not a new concept [5] but constitute another reason for having the state as a parameter in a message.

When sending implicit acknowledgements and messages, we propose that an agent takes an *initiator* role – *initiator*(X) holds if the protocol allows X to send the next action causing a state transition. An agent chooses a role if the protocol allows it and according to its strategy for achieving its goals. Our synchronisation protocols can support (but do not enforce) the agents adopting new roles dynamically where an agent can implicitly be a sender or a receiver of a message. Therefore, an agent X can send a sequence of messages if the condition *initiator*(X) holds during that sequence (as when continually bidding a higher price in auctions) or it can decide to be a receiver.

3.3 Terminating an Interaction

We specify that an interaction should be terminated by exchanging *four* consecutive explicit acknowledgments for achieving shared beliefs about the final state. The sender of the last performative leading to the terminal state sends the last (fourth) acknowledgment. As shown in section 4.1.2, two acknowledgements ensure that all agents believe $(\alpha \wedge E_G \alpha)$ and four acknowledgments result in all agents believing $(\alpha \wedge E_G \alpha \wedge E_G^2 \alpha)$, where α is the terminal state. Thus four acknowledgments force clean termination of an interaction with shared beliefs about the final interaction state.

3.4 Formalisation of Protocols

We can formalise our synchronisation protocols in ANML, (Agent Negotiation Meta-Language) [10] which is a multimodal, propositional dynamic logic-based meta-language for representing and reasoning about the states and processes of a negotiation (and thereby interaction). The syntax of ANML is an adaptation of the program logic described in [4], where a process may be expressed in terms of its sub-processes which can be coupled with the agent or agents executing that sub-process. We associate an agent with processes by prefixing the process with an agent in the same way an object is suffixed by its methods e.g., $r:retailer.display$ means *retailer* r executes the *display* process. Usually we omit the agent type and denote a joint process between two parties with set notation as in $\{c,r\}.shopping$. The process denoted by $a;b$ is composed of the sequence a followed by b , $a \cup b$ denotes process alternation, a^* denotes zero or more iterations and a^+ , one or more iterations. A state test operator '?' allows sequential composition to follow only if successful. For example $\{c.browse?;c.choose\}$ is the process $c.choose$ if $c.browse$ succeeds, otherwise it fails. (see [9] for more on ANML). For example, the protocol of figure 2 is defined by the following theory between agents X and Y :

$$\begin{aligned} interacting &\leftrightarrow one_of([open, closed]) \\ closed &\leftrightarrow one_of([agreed, rejected]) \\ open &\leftrightarrow offered \\ offered(X) &\leftrightarrow ([Y.counter_offer] offered(Y) \vee [Y.agree] \\ &\quad agreed(Y) \vee [Y.reject] rejected(Y)) \end{aligned}$$

For the purposes of presentation in this paper, synchronisation protocols will be partly expressed as statecharts [6] where separate states are distinct propositions and arcs represent processes. The statecharts are accompanied by tables for explicating the propositions and processes at particular states and actions. The ANML theory for the whole synchronisation protocol can then be obtained jointly from a statechart and its table. For example, table 1 and figure 3 specify the synchronisation protocol \mathcal{R} . Abbreviations $[X \setminus Y]$ and $B_X s_i$ are explained in section 4.

4. GUARANTEED RECEIPT OF MESSAGES

In this and the next section, we specify and validate synchronisation protocols between two agents for systems in which there is guaranteed, (figure 3, protocol \mathcal{R}), and non-guaranteed, (figure 5, protocols \mathcal{T} and $\mathcal{R} + \mathcal{T}$), receipt of messages. Unlike the protocols in [3], the sender and receiver roles are not reserved to one agent. We allow agents to change their roles dynamically because any agent may send a message in an interaction according to the common interaction protocol. This is signified by the meta-process $[X \setminus Y]$ where Y replaces X (and vice versa). In an interaction, a sender of a message may become the receiver of the next message. Therefore we cannot separate the synchronisation protocol of the sender from that of the receiver (as do [3]).

The basic idea behind our synchronisation protocols is that a sender does not revise its beliefs with the state transition it is proposing, but rather waits for an acknowledgement of the receipt of its message before doing so. A sender can predict the next interaction state and the group's future beliefs, provided its message is successfully received, of which it can only be sure when receiving an acknowledgement. As an interaction progresses with acknowledgements, an agent's predictions are discharged while increasing the level of mutual beliefs. In what follows, we analyse the case of guaranteed receipt of a message, when even though communication is imperfect, the network has not failed entirely.

The synchronisation protocol, called \mathcal{R} , in figure 3 compensates for message delays and loss by repeatedly sending a message with the assumption that it is eventually delivered and the agents update their beliefs about the interaction state. Protocol \mathcal{R} considers an interaction between two agents X and Y . Let the states of the bilateral interaction follow the sequence $s_0 \dots s_i \dots s_n$ and let s_i denote a terminal state where there are no further actions possible from s_i . The formula $B_X s_i$ is used to express the proposition that X believes the state is s_i . The states SM , ($1 \leq M \leq 10$), and actions AM , ($1 \leq A \leq 13$), in figure 3 and the protocol \mathcal{R} are defined in table 1.

In the synchronisation protocol \mathcal{R} , if agent X is the sender of a message, then it keeps sending its message until it believes that agent Y has received it. As a sender, X does not yet believe the new state is s_i which it proposes, until it receives Y 's acknowledgement. On receiving an explicit acknowledgement, X then believes the state is s_i along with Y 's beliefs about both X and s_i . If instead X receives an implicit acknowledgement with an action to change to state

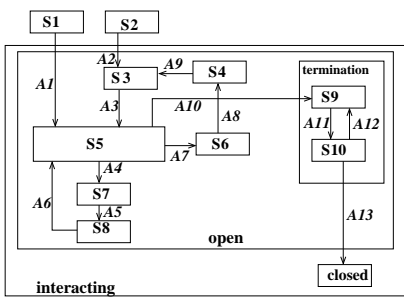


Figure 3: Synchronisation Protocol \mathcal{R}

Table 1: The States and Processes of Protocol \mathcal{R}

S1	$\neg initiator(Y), B_Y \neg interacting$
S2	$initiator(X), B_X s_{i-1}, (s_{i-1} \leftrightarrow \neg interacting)$
S3	$msg_sent(X), B_X s_{i-1}$
S4	$expl_ack_received(X), B_X s_i, B_X B_Y s_i, B_X B_Y B_X s_{i-1}$
S5	$msg_received(Y), B_Y s_i, B_Y B_X s_{i-1}$
S6	$expl_ack_sent(Y), B_Y s_i$
S7	$impl_ack_sent(Y), B_Y s_i$
S8	$impl_ack_received(X), B_X s_{i+1}, B_X B_Y s_i, B_X B_Y B_X s_{i-1}$
S9	$sent_ack_m(Y), \neg E_G^2 s_t, B_Y E_G^j s_t$
S10	$received_ack_m(X), B_X E_G^{j+1} s_t$
A1	$Y.receive(i, X, Y, a_i, s_i)$
A2	$(B_X B_Y s_i?; X.send(i, X, Y, a_i, s_i))^+$
A3	$Y.receive(i, X, Y, a_i, s_i)$
A4	$initiator(Y)?; (B_Y B_X s_{i+1}?; Y.send((i+1, Y, X, a_{i+1}, s_{i+1})))^+$
A5	$X.receive(i+1, Y, X, a_{i+1}, s_{i+1})$
A6	$[X \setminus Y]; [i \setminus (i+1)]$
A7	$((B_Y B_X B_Y s_i?; Y.send(i, Y, X, ack, s_i))^+$
A8	$X.receive(i, Y, X, ack, s_i)$
A9	$(B_X B_Y s_{i+1}?; X.send(i+1, X, Y, a_{i+1}, s_{i+1}))^+; [B_X s_{i-1} \setminus B_X s_i]$
A10	$(s_i \leftrightarrow s_t)?; (B_Y B_X B_Y s_i?; Y.send(i, Y, X, ack_1, s_t))^+; [ack_m \setminus ack_1]; [j \setminus 0]$
A11	$X.receive(m, Y, X, ack_m, s_t)$
A12	$(\neg E_G^2 s_t?; (B_X E_G^{j+1} s_t?; X.send(m, X, Y, ack_{i+1}, s_t))^+)$
A13	$(E_G^2 s_t)?$

s_{i+1} , X first believes the state s_i , that it previously proposed, has been reached, then it updates its beliefs to accept s_{i+1} as the new state and whatever it believes Y believes. Thus, X revises its belief about the interaction state consecutively twice from s_{i-1} to s_{i+1} . After receipt of an implicit acknowledgement, if X is not the initiator, it keeps sending an explicit acknowledgement until it believes Y has received it. At this stage, X and Y may switch roles. If X is the initiator after either an implicit or explicit acknowledgement, X keeps sending the next state transition in the interaction protocol until it receives another acknowledgement. Termination of the interaction is ensured with a final terminating loop by repeatedly exchanging explicit acknowledgements until each agent believes $E_G^2 s_t$, where the terminal state s_t holds in the *closed* state.

4.1 Proof of Safety of Protocol \mathcal{R}

Safety here means: (i) there is no inconsistency in the joint beliefs of the agents about the interaction state before and

after each state transition, (ii) that an interaction terminates in a state that is shared belief between the two agents. This section proves that protocol \mathcal{R} exhibits both these features.

4.1.1 Proof of Consistency of Joint Belief

Simultaneous changes in beliefs at both agents' sides about the interaction state are unfeasible because messages do not get relayed instantaneously. Given this, our work aims to ensure there is a point where every agent believes in the same interaction state before the next state transition. We prove by induction that the synchronisation protocol \mathcal{R} ensures consistency of joint beliefs are attained between agents X and Y after each state-triggering message (a message containing an action for a state transition). To do so, we need to prove that after each state-triggering message to s_i , both agents individually come to believe s_i , resulting in $E_G s_i$, before the next state transition.

PROOF. Base Case: From Assumption 2, at the start of the interaction the interaction protocol and the state of interaction are common belief.

Induction Hypothesis: At point k , assume agents X and Y believe the interaction state is s_1 .

Induction Proof: At point $(k+1)$, ($S2, A2$ in figure 3), let agent X keep sending a message m_2 for a state transition to s_2 until $B_X B_Y s_2$. When agent Y receives m_2 , (state $S5$ in figure 3), it starts believing the state is s_2 and repeatedly sends an implicit or explicit acknowledgement for s_2 at point $(k+2)$, (processes $A4, A7$). Agent X still believes the state to be s_1 . At point $(k+3)$, (state $S4$ or $S8$), on receiving either an explicit or implicit acknowledgement, agent X updates its belief to s_2 and stops sending m_2 . Both agents believe the state is s_2 at point $(k+3)$. If the acknowledgement was implicit, agent X then updates its beliefs to the next state s_3 . \square

From this, a receiver believes that the sender has not changed its beliefs about the interaction state, but the receiver can believe that the sender is predicting a change. Eventually each agent believes that both agents previously believed s_2 . Reasoning about the previous beliefs of an agent is possible through an interaction protocol [10], and helps to remove uncertainty about the history of the interaction.

4.1.2 Proof of Termination with Shared Belief

To end an interaction, X and Y send four acknowledgements between themselves, for the group and each agent to believe $E_G^2 s_t$ where s_t is the terminal state (see figure 4).

PROOF. At point k . Agent X repeatedly sends a message for closing the interaction in state s_t until $B_X B_Y s_t$. X believes a state prior to s_t . Y receives X 's message, updates its belief to s_t and predicts $E_G s_t$.

Point $(k+1)$: first acknowledgement Ack_1 . Y believes s_t and repeatedly sends Ack_1 until $B_Y B_X s_t$, for letting X believe that Y believes s_t .

Point $(k+2)$: second acknowledgement Ack_2 . X receives Y 's acknowledgement Ack_1 , stops sending s_t and revises its belief to $(B_X s_t \wedge B_X B_Y s_t)$. At this point everyone believes s_t (i.e. $E_G s_t$). X repeatedly sends Ack_2 until it receives Ack_3 from Y .

Point $(k+3)$, third acknowledgement Ack_3 . Y receives Ack_2 and stops sending Ack_1 . Y adds to its belief $B_Y B_X s_t$ and sends Ack_3 until it receives Ack_4 . Y does not believe $E_G^2 s_t$ yet because it cannot be sure that X has received Ack_3 .

Point $(k+4)$, fourth acknowledgement Ack_4 . X receives Y 's acknowledgement Ack_3 , stops sending Ack_2 and revises its belief to $(B_X s_t \wedge B_X B_Y s_t \wedge B_X B_Y B_X B_Y s_t)$. X believes $(s_t \wedge E_G s_t \wedge E_G^2 s_t)$. X repeatedly sends Ack_4 .

Point $(k+5)$, Y receives Ack_4 . Y adds $B_Y B_X B_Y B_X s_t$ to its belief i.e. $B_Y E_G^2 s_t$. Thus each agent now believes the terminal state and shared beliefs about $s_t - (s_t \wedge E_G s_t \wedge E_G^2 s_t)$. \square

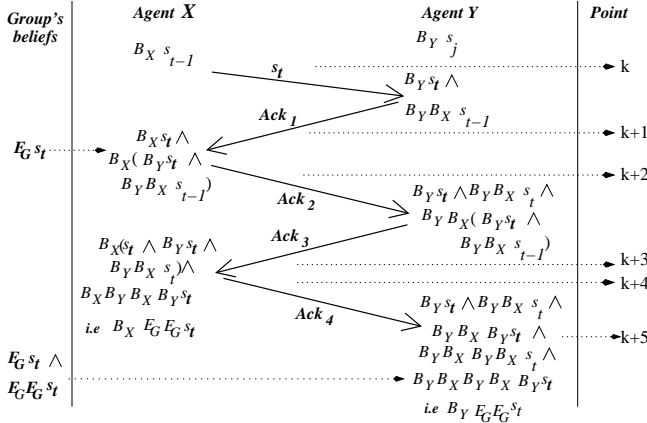


Figure 4: Termination with Shared Beliefs

5. NON-GUARANTEED RECEIPT OF MESSAGES

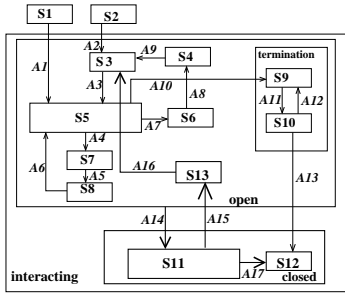


Figure 5: Synchronisation Protocols \mathcal{T} and $\mathcal{R} + \mathcal{T}$

Usually there is no guarantee that a message will eventually reach its destination. In such cases, according to assumption 3, a sender is informed if its message fails to be delivered. On the other hand, a receiver X , which is expecting a message that does not arrive, cannot know whether the communication layer is at fault or the other agent Y is failing to respond because it has crashed. To this end, we specify a simple synchronisation protocol \mathcal{T} , (figure 5), based on timeouts, where an agent waits for a message for a finite time t . Let the interaction state *timeout* hold after a timeout. Unlike for other interaction states ($S1$ to $S10$), given the failure of messages in reaching their destination, it is not possible to send four acknowledgements between the agents in order to achieve the shared belief $E_G^2 \text{timeout}$. Therefore, shared beliefs about timeouts are reached through negative introspection; every agent believes that not all agents believe in any regular interaction state and therefore every agent eventually believes *timeout*.

In protocol \mathcal{T} , messages and acknowledgements are not repeatedly sent as in protocol \mathcal{R} . Therefore for protocol \mathcal{T} , the actions $A1$ to $A13$ in figure 5 do not have the test

conditions and repeated sending of table 1. The states and actions (processes) in table 1 also holds for protocol \mathcal{T} after removing the test conditions and iterations from sending messages (in processes $A1$ to $A13$) in protocol \mathcal{R} . The new states in figure 5 for protocols \mathcal{T} are:

S11	$received_timeout(X), B_X \text{ timeout}$
S12	$E_G^2 s_t \vee E_G^2 \text{ timeout}$
S13	$restarted(X), \bar{E}_G s_t$
A14	$X.receive(i, -, X, \text{timeout}, received_timeout(X))$
A15	$X.send(n, X, Y, \text{restart}, s_t)$
A16	$[B_X s_{i-1} \setminus E_G s_i]$
A17	ε

If a sender X is informed of failure of delivery of its message, then it believes the interaction has failed in the *timeout* state. The same applies for termination of an interaction through four acknowledgements when the sender of an acknowledgement is informed that its acknowledgement cannot be delivered. Similarly, if an agent X as a receiver is waiting for a message which does not arrive before a timeout, then it believes the interaction state to be *timeout*. If an agent believes a timeout (state $S11$), it can restart (action $A15$) the interaction in the previous state that is commonly believed (leading to $S13$).

5.1 Proof of Safety of Protocol \mathcal{T}

As for protocol \mathcal{R} in section 4.1, two proofs are required – attaining consistency in the joint beliefs during an interaction and proof of safely terminating an interaction.

PROOF. (Consistency Proof). The proof for belief revision after a message or acknowledgement is similar to that of protocol \mathcal{R} when the messages are delivered. A sender does not update its beliefs about the message it sends, but a receiver updates its beliefs on receiving explicit or implicit acknowledgements. In the case of timeouts and restarts, consistency is ensured by restarting the interaction in a previous state that is believed by everyone. \square

PROOF. (Termination Proof). If there are no timeouts, the proof for safe termination for protocol \mathcal{R} also applies to protocol \mathcal{T} . If a timeout occurs, there are two cases for the termination proof depending on the role of an agent. First, if X is a sender and believes a *timeout* state from being notified by the network, from negative introspection the receiver Y also eventually believes a *timeout* state on not receiving X 's message or acknowledgement. Second, if X is a receiver and is the first to believe a *timeout* state, then the sender Y on not receiving an acknowledgement eventually believes the *timeout* state from negative introspection. From the property of negative introspection, all agents believe they are not aware what is the successful end state of an interaction. So they all believe that they all believe that not all agents believe the interaction is successful. Similarly if a timeout occurs during the exchange of the first 3 acknowledgements of a terminal state, then from negative introspection either agent will come to believe the *timeout* state because they will not receive the next acknowledgement. Regarding the fourth acknowledgement ack_4 , firstly if it cannot be delivered by the network then the sender will be notified and the receiver will not receive ack_4 . If the sender has crashed, then the receiver does not receive ack_4 . From negative introspection, whenever an agent does not receive ack_4 , it eventually believes *timeout*. \square

5.2 Repeated Messaging and Timeouts

Protocols \mathcal{R} and \mathcal{T} can be combined to give the hybrid protocol $\mathcal{R} + \mathcal{T}$ for more flexible behaviour in cases where there is both guaranteed and non-guaranteed receipt of messages. Protocol $\mathcal{R} + \mathcal{T}$ includes the timeouts from protocol \mathcal{T} and, from protocol \mathcal{R} , the ability for an agent to keep sending a message or an acknowledgement until it believes that its message or acknowledgement has been received. Thus, an agent repeatedly sends a message until it receives an acknowledgement or a timeout, in which case it restarts the interaction in a commonly believed previous state. For the protocol $\mathcal{R} + \mathcal{T}$ in figure 5, actions $A1$ to $A13$ are similar to those in table 1 for protocol \mathcal{R} . The states $S11$, $S12$, $S13$ and actions $A14$, $A16$, $A17$ are the same for both protocols \mathcal{T} and $\mathcal{R} + \mathcal{T}$. The action $A15$ in protocol \mathcal{T} is replaced by $((B_X B_Y \text{ timeout})?; X.send(n, X, Y, restart, s_i))^*$ for protocol $\mathcal{R} + \mathcal{T}$. When the communication framework guarantees delivery of a message, then the hybrid protocol $\mathcal{R} + \mathcal{T}$ can be used if the agents have deadlines. If an agent reaches its deadline while waiting for a message, it may choose to timeout (especially when the network is lagging). Realistically, an agent cannot keep sending a message for periods extending over days or weeks without the risk of network bottlenecks or wasting computation time and power. In these cases or in case of network lags, timeouts are needed to halt the interaction process. In the non-guaranteed case, repeated sending of a message increases the probability of it reaching the destination. Thus protocol $\mathcal{R} + \mathcal{T}$ combines the advantages of both protocols \mathcal{R} and \mathcal{T} for realistically coping with broader reasons for failures instead of specific defects in the communication medium.

PROOF. Both the safety proofs for protocol $\mathcal{R} + \mathcal{T}$ in the group reaching consistent joint beliefs about an interaction state and the safe termination of an interaction follow from the proofs for protocols \mathcal{R} and \mathcal{T} . Protocol $\mathcal{R} + \mathcal{T}$ inherits the properties of its parent protocols \mathcal{R} and \mathcal{T} . \square

6. RELATED WORK

Early work on knowledge-based reasoning about protocols can be found in [5] and [3], all focussing on the bit [1] and sequence transmission [3] problems using a first order modal formalism. Despite common knowledge being impossible to achieve in a faulty communication medium, such knowledge-based reasoning methods have predominantly been used for proving the correctness of protocols. However in this work, we adopt less idealistic solutions, and use belief modalities. Reasoning about belief promises more realistic solutions to interoperation between agents through interaction languages and protocols. In any case, work on bit or sequence transmission problems cannot be literally applied to higher-level agent interaction which has requirements different from sending a linearly-ordered sequence of bits over the network. In multi-agent interactions, instead of bits, proposals for action and state changes according to an interaction protocol are exchanged and the state of an interaction is itself a property of the mutual beliefs of a group of agents. Another difference from existing work concerns the dynamic roles of the sender and receiver in not only bilateral interactions, but also the diversity of roles possible in multi-lateral interactions.

Distributed database solutions cannot be applied here because in the case of timeouts or crashes, contrary to roll-backs in databases, an agent does not erase its beliefs and re-

turn to its previous state. Rather an interaction is restarted with some agents having increased beliefs and possibly possessing an advantage over their opponents. Our approach discourages such naïve solutions which may engender malicious agents thriving on communication failures.

7. CONCLUSION AND FURTHER WORK

Dealing with the consequences of communication failures in agent interaction has usually either been relegated to the ISO transport layer or has been analysed through knowledge-based reasoning about the transmission of bits over a network. This paper argues that agents need to reason about and compensate for message delays and loss to avoid contradictory beliefs about the interaction state. Thus, a belief-based approach is adopted for achieving a degree of common beliefs amongst a group of agents about an interaction. Reaching an equivalent degree of knowledge is more complex (if not impossible) given the property of knowledge being true. Specifically, we ensure and prove the safe progress and termination of an interaction through synchronisation protocols when an agent is capable of positive and negative introspection over terminal states. As further work, the difference between joint, common and shared beliefs and the justification for the latter can be extensively explored. A suite of synchronisation protocols for bilateral and multilateral interactions can be designed according to constraints. We also aim to explore synchronisation protocols in the context of mobile telecommunications as an analysis of the performance of agent interaction in realistic environments.

8. REFERENCES

- [1] K. A. Bartlet, R. A. Scantlebury, and P. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *Comms. of the ACM*, 12(5), 1969.
- [2] G. Boolos and R. Jeffrey. *Computability and Logic*. Cambridge University Press, 1989.
- [3] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
- [4] R. Goldblatt. *Logics of Time and Computation*. CSLI, 1987.
- [5] J. Halpern and L. Zuck. A little knowledge goes a long way: Knowledge-based derivations and correctness proofs for a family of protocols. *JACM*, 39(3), 1992.
- [6] D. Harel and M. Politi. *Modeling reactive systems with statecharts*. McGraw-Hill, 1998.
- [7] <http://www.fipa.org>. *Foundation for Intelligent Physical Agents, FIPA. Agent Communication Language*.
- [8] J. Meyer and W. Van Der Hoek. *Epistemic Logic for AI and Comp. Science*. Cambridge Univ. Press, 1995.
- [9] S. Paurobally. *Rational Agents and the Processes and States of Negotiation*. PhD thesis, Imperial College, 2002.
- [10] S. Paurobally and R. Cunningham. Verification of protocols for negotiation between agents. In *ECAI-15*, pages 43–48, 2002.
- [11] J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.
- [12] W. Van der Hoek and M. Wooldridge. Model checking knowledge and time. In *9th Int. Spin Workshop on model checking of software*, pages 95–111, 2002.