Verifying the Contract Net Protocol: A Case Study in Interaction Protocol and Agent Communication Language Semantics

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

May 14, 2004

Abstract

Multi-agent conversations are built upon two components: agent communication languages (ACLs) that specify the individual messages that can be exchanged and interaction protocols (IPs) that specify the sequences in which these message can be arranged. Although informative, the semantic definition proposed for the most standard ACL (FIPA 1997) is complicated and contentious, while published IPs tend to be ambiguous, incomplete, and unverified with respect to message semantics. As a case study to clarify and help rectify these problems, we have investigated verification of the contract net protocol when its messages are presumed to be expressed in FIPA ACL. In order to help both informal comprehension and formal verification we separate several concerns. We suggest a revised and simpler core semantics for many of the FIPA ACL speech acts, using the same belief-intention style of logic, although the underlying ideas are not dependent on this detail. An extended form of propositional dynamic logic and statecharts is used to express IPs. States are interpreted using mutual beliefs and intention, and properties such as termination and consistency of joint beliefs are shown.

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 agent communication languages (ACLs) and interaction protocols (IPs). An ACL (for example

KQML [3], FIPA ACL [4]) specifies the individual communicative acts (CAs), typically as classes of asynchronous messages modelled on the Theories of Speech Acts enunciated by Austin [1] and Searle [12] . The 1997 semantic specification for FIPA ACL is expressed using a logic of belief and intention and derived from work associated with Sadek [11]. This specification is informative has been criticised on various grounds [10] [6], not least that it is unverifiable [16]. Relatedly, an IP (for example the contract net protocol (CNP) [15] or an English auction protocol) specifies message sequences that can lead towards a goal state. However to date, may of the published specifications for these protocols suffer from ambiguities and incompleteness [8]. This lack of precision can arise from the inherent inexpressiveness of diagrammatic representations such as Petri-nets [7] and AUML [4], or from the level of abstraction chosen when using informal language or formal logic. Such representations can nevertheless be informative and helpful for comprehension. Commitment-based semantics have been used for modeling multi-agent interactions [17]. However this work remains focussed on the creation, fulfilment and discharge of an agreement, tends to be centralised, through for example an instituion, and discourages interactions whose goals are to share experience and model their environment. The legal issues also are often not considered.

Against this background, we use the CNP (arguably the most widely adopted IP) as a case study in this paper, to expose and apply a simpler semantics for an ACL, and to provide a compatible semantics for the protocol, which itself is represented in extended propositional dynamic logic (PDL [5]) and statecharts. Together these allow us to prove termination and consistency regarding some of the group's beliefs when interpreting the CNP. At the same time, in the light of criticisms about ambiguities in ACL semantics, our proposed semantics for an ACL and IPs serve as an example of how belief semantics [2] can provide insight in the clear specification of agent interactions.

Thus, this paper addresses the unresolved problem of a suitable ACL semantics for expediting agent interactions. Our first contribution is to show how to separate treatment of message delivery, sincerity, and implicit protocol issues within the existing style of FIPA-ACL logic (in the current work they are all implicitly assumed). This separation of issues has enabled us to simplify verification of the contract net protocol with respect to the message semantics. Our semantics are not dependent on the actual choice of ACL or the particular style of logic that we use. This means that a protocol specified in extended PDL does not depend on the actual ACL being used or the semantics of the ACL. The second contribution is also partly methodological. We overcome the lack of expressiveness of weaker graphical representations by using an enhanced form of statechart and extended PDL to represent the protocol both visually and using a logical theory. Finally, we show how the definitions of the states allows a semantic representation using a logic in the belief-intention style. This enables us to derive more specific properties of the protocol in terms of joint beliefs. We refer to a belief-intention *style* of logic but we do not provide a formal definition of any such logic in this paper (see section 5). Monadic modal operators for a belief (B_i) and an intention (I_i) of an agent i are used as intellectual props for deductive inference and treated as independent except for explicit interaction axioms. A belief logic is a useful ideal for giving epistemic status to the consistent but not necessarily true internal propositions that can be used by a designer to express and reason about information internal to an agent. Treating intention in a similar way is also a useful ideal for succinct reasoning about a goal state without getting into more temporal reasoning. The durability of the Belief-Desire-Intention paradigm for practical deliberative agents also provides a heuristic justification for this sort of reasoning.

The remainder of this paper is structured in the following way. In the next section we provide an informal summary of the separate issues in simplifying reasoning about the FIPA ACL. Section 3 critically analyses a representation of the CNP in Petri nets and those aspects of the specification that cannot be captured with in this approach. Section 4 develops a novel formal representation of the CNP in extended PDL and extended statecharts. In section 5, we discuss the axioms and assumptions of the belief logics and ACL. Sections 6 and 7, respectively, summarize our use of communicative acts and interpret the states in the CNP. In section 8, we validate our approach by proving desirable properties of the CNP in our framework. Section 9 presents our conclusions and future work.

2 Issues in Reasoning about the FIPA ACL

As defined in FIPA-ACL 1997 there is a feasibility pre-condition (FP) and a rational effect (RE) associated with each CA. The FP conjoins a sincerity condition (SC), with a typically more complex Gricean condition (GC) to preclude a redundant message. The RE expresses the condition that the sender may use in planning the communicative act. The FIPA *inform*, (or KQML *tell*) is the basic CA. As a message it has parameter for sender s, receiver r and propositional content ϕ. The SC is $B_s\ \phi$ and the RE is $B_r\ \phi$ The GC expresses the belief that the sender does not believe the receiver believes ϕ, or is uncertain about it.

The obvious criticism is that the Gricean condition introduces inessential complexity, even if the term “inform” is inappropriate without it, but there are deeper issues. The semantic conditions as expressed are sender oriented, and there is no overt association between the semantic conditions and the occurrence of the message itself, after all, sincerity and non-redundancy are social conditions, not mechanical. Although these concerns have been pointed out in [13], [16] we take a new step in removing much of their impact by re-expressing the semantics in the belief-intention logic itself.

The key step is not obvious and indeed exploits an obvious ”hack” in the logic, which is to avoid the detailed expression of temporality or causation by using the special proposition $done(a, act)$, for any agent a and action act. Using PDL notation, we express $done(a, act)$ as $done(a.act)$. Now the propositions $I_s\ done(s.m)$ and $B_r\ done(s.m)$ respectively can express that the sender agent s intends that m be sent, and receiver agent r believes that m has been sent by s. These are the tightest pre- and post-conditions we can express in the logic, and suggest the following PDL axiom schema for such messages, assuming there is a sender *s* and a receiver *r*:

$$I_s done(s.m) \leftrightarrow [s.m] B_r done(s.m)$$

This causality style of schema is potentially verifiable in a logic that is grounded in machine states, but for our purposes it enables us to separate the FP and RE from the message transport. FIPA-like semantics are re-instated by assuming that the receiver believes that the message transport post-condition entails the message RE ($B_r done(s.m) \rightarrow B_r RE_m$), and that the pre-condition $I_s done(s.m)$ is itself a primitive plan by which the sender can attain the RE. The FIPA semantics do not assume the sender to intend the RE, but one can take the view that the FP should be strenghened to entail $I_s done(s.m)$.

Indeed, each agent can access a common ontology of messages and infer from the receipt of a message what it means, and this may vary with any other environmental context that is available. We will simply drop the GC part of the FP, on the grounds that it is really a social protocol for human communication which need not be presumed in the context of any other interaction protocol, but retain the SC part as simpler FIPA-like pre-condition, and allow the FIPA RE as a trustworthiness assumption (see section 5.2).

It turns out that the remaining FIPA-ACL messages that we use can be re-expressed as special cases of the *inform* act in context. This was known when the standard was prepared, but in some cases obscured and made erroneous by the inessential complexity we have sought to remove. For example a *propose* message, sent by a potential CNP contractor to the manager is an *inform* with the propositional content that *if* the sender believes that the receiver intends that the action be done by the sender, *then* the sender (will) intend this (too). So a sincere *accept* or *reject* requires belief by the manager in the propositional content of such a proposal, and becomes an *inform* message with content expressing the manager's intention, so that the contractor can discharge the conditional "promise". Given these issues in the semantics of ACLs, there also exist issue in the current representations of IPs.

3 Issues in Representing IPs

To illustrate the required expressiveness of a specification language for realistic IPs, we consider the CNP since this is probably the most widely used protocol in the field. Figure 1, from [7], presents the CNP as a "Coloured" Petri Net. The interaction is started by a manager issuing a *call for proposals (cfp)*. Potential contractors respond with *proposals*, which the manager either rejects or accepts. Accepted proposals can be either cancelled by the manager or executed by the proposer, who later informs the manager of success or failure of their execution. The manager may also re-select other proposals or issue a new *cfp*.
However figure 1 is both ambiguous and incomplete:

- A manager has no means of deciding whether the overall contract net process has succeeded or failed. For example, if out of *n* proposals, *m* are done and *p* fail, the state of the overall process is undefined.
- This Petri net illustrates the same states and triggers for all three contractors. This is schematically redundant and is particularly problematic in open multi-agent systems; it would be hard to express the state of many contractors, which join the interaction dynamically, without more formal techniques.

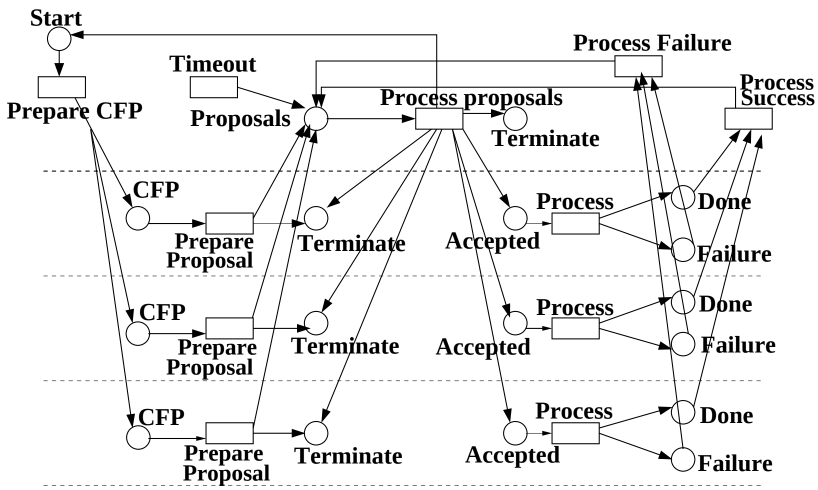

Figure 1: Contract Net conversation with three contractors in Coloured Petri Net

- Even if roles are used, as in AUML [4], there is no binding of an action to an agent's identity. Thus the manager should only send acceptance messages to those contractors who proposed, and only those proposers should inform the manager of the result of their execution.
- A deadlock will occur if all contractors refuse to bid or if the manager does not issue a new call. This is because for termination or for issuing a revised call, at least one contractor has to send a proposal to the current call.
- Alternative actions cannot be distinguished. For example, instead of the manager being able to both accept and reject each proposal, as in figure 1, it should either accept or reject each proposal.

There are also other aspects missing in the CNP representation. It is not shown that different contractors have different beliefs depending on what they sent and received. Thus, since proposals are not broadcasted, a contractor will not know who else other than itself has sent a proposal, whilst the manager knows all proposers. In turn, the manager has to ensure that it accepts or rejects only those agents which sent a proposal in the first place. Thus, there is a shared belief between a contractor and the manager, (e.g. whether that contractor has sent a proposal or whether its proposal has been accepted). In the same protocol, some states are public, some are private to an agent, while yet others are mutually believed. Sometimes, the private states of an agent have to be expressed; for example, the state when the manager is deliberating a proposal is needed in order to show that: *i)* there is a specific set of contractors whose proposals a manager has chosen to accept and another distinct set whose proposals are rejected; and *ii)* contractors wait for the manager to finish its deliberation. These required features of the CNP are captured through our representation in the next section.

Table 1: **Semantics of Extended PDL**

$M, w \models p$	iff $w \in V(p), p \in PROP$
$M, w \models [\gamma]A$	iff $\forall w_1(wR_\gamma w_1$ implies $M, w_1 \models A)$
$M, w \models A(X)$	iff $M, w \models A$ and $X \in Ag\_group$
$M \models (\gamma_1 :: \gamma_2)$	iff $R_{\gamma_1} \subseteq R_{\gamma_2}$
$M, w \models none\_of(S)$	iff $\forall A(A \in S$ implies $M, w \not\models A)$
$M, w \models one\_of(S)$	iff $\exists A_1 \forall A_2((A_1 \in S$ and $M, w \models A_1)$ and $(A_2 \in S \text{and} M, w \models A_2))$ implies $A_1 \leftrightarrow A_2)$
$R_{Ag\_group.\gamma} \subseteq R_\gamma$	
$R_{A?} = \{(w, w) : M, w \models A\}$	
$R_{\gamma?} = \{(w_1, w_2) : (w_1, w_2) \in R_\gamma\}$	

4 A Formal Representation for Interaction Protocols

Here we show how to represent IPs formally in extended PDL and graphically in extended statecharts. First, however we specify our extensions to PDL.

4.1 Extended PDL

We extend PDL to enable us to reason about the effect of processes on interaction states (refer to [8] for more details and a complete axiomatisation). Specifically, let *A* denote a formula and γ denote a process. In general, a formula may be in multi-modal logic, including beliefs, desires and intentions, as well as actions. The formula $[\gamma]A$ has the intended meaning: *A* holds after executing process γ. (The formula $[\gamma]A$ is also the weakest precondition for γ to terminate with *A*). The syntax of the extended logic is defined below, where $\mathcal{A}$ denotes one agent or a set of agents and the term *States* denotes set of formulae.

$$\text{Formulae:} \quad A ::= p \mid \bot \mid A_1 \rightarrow A_2 \mid [\gamma]A \mid B_{\mathcal{A}}A \mid I_{\mathcal{A}}A \mid A(\mathcal{A}) \mid \gamma_1 :: \gamma_2 \mid none\_of(States) \mid one\_of(States)$$

$$\text{Processes:} \quad \gamma ::= \varpi \mid \gamma_1;\gamma_2 \mid \gamma_1 \cup \gamma_2 \mid \gamma^* \mid A? \mid null \mid abort \mid \mathcal{A} \,.\, \gamma \mid \gamma?$$

The complex process $(\gamma_1;\gamma_2)$ denotes the sub-process γ_1 followed by γ_2, the process $(\gamma_1 \cup \gamma_2)$ either γ_1 or γ_2 non-deterministically, while γ^* denotes zero or more iterations of process γ. A state test operator "?" allows sequential composition to follow only if successful. A *null* process represents no execution, while an *abort* process results in a failed state. We extend the program logic of PDL so as to express multi-agent interactions. The semantics of the additional constructors are specified in table 1, and are based on a Kripke model denoted by *M = (W,* R_γ*, V)* [5]. We add types for agents and roles. We assume throughout that each atomic formula *p*, agent and instance of an atomic process ϖ can be denoted by a distinct identifying term. Set notation is used to manipulate sets of agents and interaction states. The formula *none_of(B)* holds if none of the formulae in the set *B* are true. The formula *one_of(B)* holds if only one of the formulae in the set *B* is true.

An agent or a group of agents, $\mathcal{A}$, may execute atomic actions or complex processes, γ. So the term $\mathcal{A}.\gamma$ can be read as $\mathcal{A}$ executes process γ, as for example in *r:retailer.display* means *retailer r* executes the *display* process, but the agent role may be omitted. Using set notation, we can denote a joint process between two parties as $\{$*r,c*$\}$.*shopping*. The formula $A(\mathcal{A})$ allows state A to have an agent or a group, $\mathcal{A}$, as parameters. The semantics of $M, w \models (\gamma_1 :: \gamma_2)$ states that all the worlds obtained through execution of process γ_1 are elements of the set of worlds possible through performing γ_2. For example,
EbayAuction::EnglishAuction means that all the rules in the English auction apply to the Ebay auction.

4.2 CNP in Extended PDL and Statecharts

Figure 2 shows the CNP in an extended form of statecharts and each state is fully defined in figure 4. The process $[X \backslash Y]$ means the process of replacing occurences of *X* with *Y* in the resulting state.

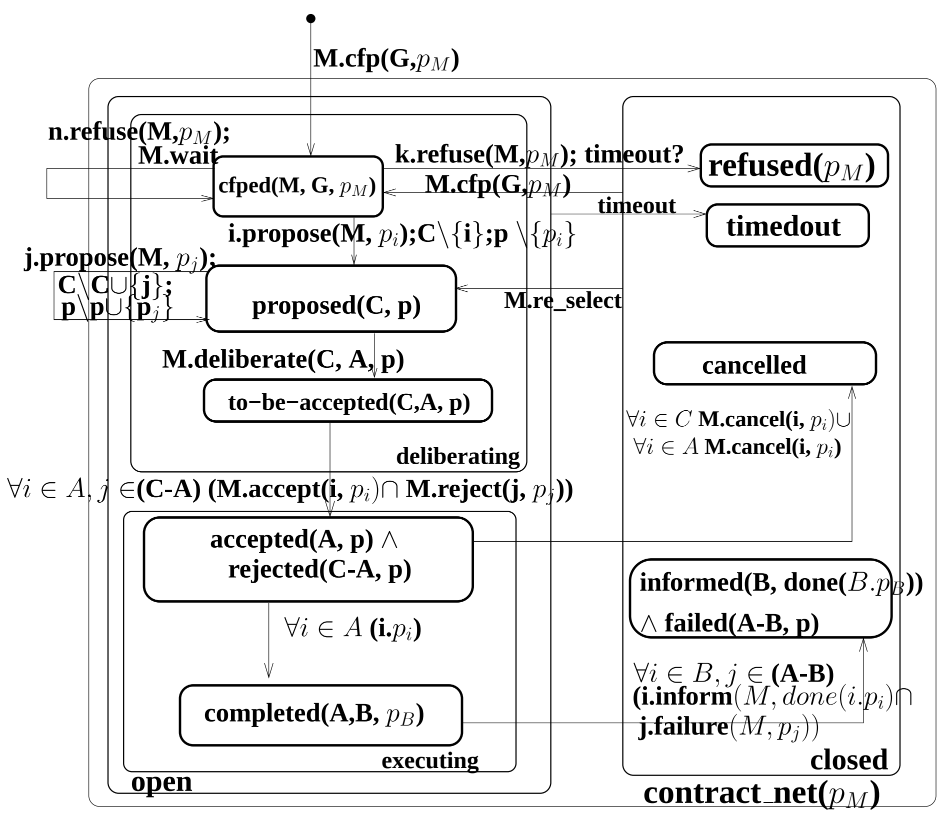

Figure 2: CNP in Extended Statechart Notation (see section 7 for interpretation of the states

$\neg$***contract_net***$\leftrightarrow$[***G.contract_net_process***]***closed*** **(1)**

contract_net $\leftrightarrow$ ***one-of*** ({ ***open , closed***}) **(2)**

closed $\leftrightarrow$ ***one-of***({***(failed***$\wedge$***informed), timedout, cancelled, refused***}) **(3)**

open $\leftrightarrow$ ***one-of***({***deliberating, executing***}) **(4)**

deliberating $\leftrightarrow$ ***one-of***({***cfped, proposed, to-be-accepted*** }) **(5)**

executing $\leftrightarrow$ ***one-of***({***(accepted***$\wedge$***rejected), completed*** }) **(6)**

$\neg$ ***contract_net***$\leftrightarrow$[***M:manager.cfp(G ,***p_M***)***]***cfped(M,G,***p_M***)*** **(7)**

Figure 3: Theory of CNP's diagram structure in extended PDL

cfped(M,G,p_M***)***$\leftrightarrow$***(***

[n:contractor.refuse(M,p_M***); M.wait]cfped(M,G,*** p_M***)***

$\vee$ ***[n:contractor.refuse(M,***p_M***); timeout?] refused(***p_M***)***

$\vee$ ***[i:contractor.propose(M,***p_i***);C*** $\setminus$ {***i***}***; p***$\setminus$ $\{p_i\}$***]***

(proposed(C, p) $\wedge \forall i(i \in C \leftrightarrow B_M done(i$***.propose(M.***$p_i$***))))*** **(8)**

proposed(C, p) $\leftrightarrow$ ***[M:manager.deliberate(C,A, p)]***

(to-be-accepted(C,A, p) $\wedge \forall i \in A, \forall j \notin A(B_M(i \in A) \wedge B_M(i \notin A))$***)***

$\vee$ ***[j:contractor.propose(M,***p_j***);C*** $\setminus C \cup \{j\}; p \setminus p \cup \{p_j\}$***]*** **(9)**

to-be-accepted(C,A,p)$\leftrightarrow$ ***[***$\forall i \in A, \forall j \in (C - A)$***(M:manager.accept(***$i, p_i$)$\cap$ ***M:manager.reject***(j, p_j)***)]***

(accepted(A, p) $\wedge \forall i \in A(B_i(accepted(i, p_i))))$

$\wedge$ ***(rejected(C-A, p)*** $\wedge \forall j \in (C - A).B_j(rejected(j, p_j)))$***)*** **(10)**

accepted(A, p)$\leftrightarrow$ ***[***$\forall i \in A(i.p_i)$***] (completed(A,B,p)*** $\wedge$ $(\forall i \in B(B_i succeeded(p_i))$

$\vee \forall j \in (A - B) B_j \neg succeeded(p_j)) \vee$***) [***$\forall i \in A(M : manager.cancel(i, p_i) \cup$

$\forall i \in C(M : manager.cancel(i, p_i)$***]cancelled*** **(11)**

completed(A,B, p) $\leftrightarrow$ $[\forall i \in B, \forall j \in (A - B)$ ***[i.inform(M, done(i.***p_i)) $\cap$ j***.failure(M,***p_j***]***

(informed(B, done($B.p_B$***))*** $\wedge$ ***failed(A-B, p))*** **(12)**

open$\leftrightarrow$***[timeout] timedout*** **(13)**

closed $\leftrightarrow$***[M:manager.cfp(G,***p_M***)***]***cfped(M,G,***p_M***)*** $\vee$ ***[M:manager.re_reselect]proposed(C, p)*** **(14)**

Figure 4: Definition of the states of the CNP in extended PDL

Figures 3 and 4 provide a corresponding logical theory of the protocol in extended PDL. Axioms (1)-(7) in figure 3 specify the relation between the states as given in the CNP diagram in figure 2. Axioms (8)-(14) both define the states and state transitions of the CNP. In more detail, axiom (1) defines that a group of agents *G* adheres to the CNP in a process instance called *contract_net_process*. Double implication in the action-condition rules allows an agent to infer the history of an interaction. Axioms (2)-(6) define the relations between parent and sub-states, as seen in the hierarchy of figure 2. There are 5 other axioms (not shown here) for ensuring when a parent state is false, none of its sub-states are true.

A manager may initiate a contract net process into a *cfped* state by issuing a call for proposal to a group of contractors *G*, only if the interaction has not yet started (Axiom 7), leading to the *cfped* state. Contractors may refuse the manager's *cfp*, and if the manager receives only refusals by the deadline, the process terminates in a *refused* state (Axiom 8). Otherwise, some contractors may refuse whilst others (proposers) send a proposal, leading to *proposed(C, p)* where *C* is the set of proposers and *p* the set of proposals, each associated to its proposer. Further proposals from other proposers are added to the sets *C* and *p*. The expression $\forall i(i \in C \leftrightarrow B_M i$*.done(i.propose(M,*$p_i$*)))* means that the manager's beliefs include the identity of all proposers in *C* (Axiom 8).

Then the contractor deliberates to record those proposals it will accept as set *A*. The condition $\forall i \in A, j \notin A(B_M(i \in A)) \wedge (B_M(i \notin A))$ ensures that the manager's beliefs include accepted and rejected proposers (axiom 9). The manager concurrently sends an *accept* message to all chosen proposers in *A*, and rejections to those in *(C-A)* (axiom 10). The manager can also cancel the *cfp* or reject all proposals through a *cancel* message.

The action $\forall i \in A(i.p_i)$ and the state *completed(A,B,p)* express the fact that all accepted contractors in *A* execute their proposals and if successful are implicitly part of the set *B*. Each contractor privately believes whether it has succeeded or not in its execution (axiom 11). Finally, the contract net process terminates after all contractors in *B* have informed the manager of success, and the rest in *(A-B)* of failure (axiom 12). In an *open* state, a timeout can occur at any point (axiom 13), while from the *closed* state the manager can re-issue a call for proposal (axiom 14).

5 Beliefs and Intentions in a Group

Given the formal representation of the CNP in extended PDL, we now give meaning to the actions and states in the protocol in terms of the beliefs and intentions of the group. To do this, we use the modal operators for beliefs and intentions [2] [14]. Specifically, an axiomatic system for belief may be defined in terms of axioms for consistency and introspection [2]. We assume that each agent in a group has such a system of beliefs. The formula $B_i\alpha$ is read as agent *i* believes α, $I_i\alpha$ is read as agent *i* intends to do α, and $E_G\alpha$ is read as everyone in a group of agents, *G*, believes α, where α itself may express an agent's beliefs and intentions. The joint beliefs ($B_{i_1}\alpha_1 \wedge \ldots B_{i_n}\alpha_n$) of a group are the sentences that are consequences of the union of the individual beliefs of the agents. We also re-use a modified version of the FIPA SL *done* operator [4]. Here, *done(i.a)* (*done(i,a)* in FIPA SL) means that agent *i* has performed action *a*. Pre-conditions *p* for doing *a* can be expressed as part of *a* (e.g. *p?;a*).

As mentioned in section 2, we ignore the Gricean condition involving uncertain beliefs. We also specify intention axioms and other axioms for sincerity and trust that hold in our environment. These axioms are independent of the ACL and the protocol, but apply to reasoning about an interacting group of agents. Thus, although the semantics or the ACLs may differ, these core axioms should nevertheless hold in all agent interactions.

5.1 Belief and Intention Axioms

We assume the tradition axioms K, D, 4, 5 for belief (but not normality). They express closure under implication, consistency, and positive and negative introspection. Together they make iterated belief redundant, i.e. $B_iB_i\alpha \leftrightarrow B_i\alpha$. In addition to a KD5 axiom for belief, we have axioms for intentions and beliefs about intentions. Like positive introspection for belief and knowledge, we assume an agent's iterated intentions collapse to a single intention (A1): $I_i I_i \alpha \leftrightarrow I_i \alpha$.

Axiom (A2) states that an agent *i* is aware of its intentions and, intends what it believes it wants to intend: $B_i I_i \alpha \leftrightarrow I_i \alpha$.

We can have a stronger system where an agent is aware of what it does not intend to do, which is some kind of negative introspection (A3): $\neg I_i \alpha \rightarrow B_i \neg I_i \alpha$.

An agent is rational, that is it does not intend what it believes it does not intend (A4): $B_i \neg I_i \alpha \rightarrow \neg I_i \alpha$.

Intentions for negative intentions also collapse to not intending α (A5): $I_i \neg I_i \alpha \rightarrow \neg I_i \alpha$.

An agent has control over its beliefs. If it intends to believe α, then it intends it (A6): $I_i B_i \alpha \rightarrow I_i \alpha$. This also implies $I_i B_i \neg \alpha \rightarrow I_i \neg \alpha$.

From axioms (A1) and (A2), we obtain the following axiom, which can be further simplified to: $I_i \alpha$ (A7): $B_i I_i B_i I_i \alpha \leftrightarrow B_i I_i \alpha$.

Definitions for common intentions may be formulated in the same way as common knowledge and common beliefs. Axioms (A1) to (A6), except for axiom (A3) hold in our framework, where α can itself include belief and intention modalities.

5.2 Assumptions

In this section, we formulate the axioms holding in our model. These are the foundations of our reasoning and proofs and combined infer a sincere and trustworthy behaviour, and that messages are successfully delivered. All of these assumptions are implicit in FIPA ACL. Although they may seem to require an ideal environment, our goal in this paper is to provide a simple and well-founded semantics that work in such a social context. Untrustworthy environments and relaxing these are avenues for future work. Let *s* (sender) and *r* (receiver) represent two different agents interacting with one another.

Sincerity Axioms: $I_s B_r B_s \alpha \rightarrow B_s \alpha$.
$I_s B_r I_s \alpha \rightarrow I_s \alpha$.

Agents are sincere and the sender does not intend the receiver to believe what it does not believe itself.

Trust Axiom: $B_r I_s B_r \alpha \rightarrow B_r \alpha$.

This states that if *r* believes that agent *s* sent a message to agent *r* that α holds, then *r* believes α. Receivers trust the sender. If α is a proposition or a belief formula, then *r* also believes *s* believes α; that is, $B_r I_s B_s \alpha \rightarrow B_r \alpha \wedge B_r B_s \alpha$ (Trust Axiom 2).

Cooperative Axiom: $B_r I_s I_r \alpha \rightarrow (I_r B_s I_r \alpha \vee I_r B_s \neg I_r \alpha)$.

The agents are co-operative. Thus on receiving a message, an agent replies, even if it is a refusal or a rejection.

Similar to the FP and RE of a CA, there are preconditions and postconditions when sending a message, that are independent of the meaning of the CA. Our precondition states that a sender intends for the message to be sent. For example, the CA *s.inform(r,p)* means that the sender *s* informs the receiver *r* that *p* holds. The FIPA semantics define the FP of this CA as the sender believing *p*. However, we also need to specify that the sender intends to send *r* the *inform* CA about *p*. The FP for sending all the sender's beliefs will hold. Let *done(m)* denote a sent message *m* containing a CA. Let *FP(m)* denote the FP of the CA and *RE(m)* denote the RE of the CA being sent. The axioms that apply to message exchange are:

Transport Precondition: $I_s done(m)$. Before sending a message, sender *s* intends to send it.

Transport Postcondition: $B_r done(m)$. The receiver received the message.

Message Sending: $I_s done(m) \rightarrow FP(m)$.

Message Receipt: $B_r done(m) \rightarrow RE(m)$.

If the sender intends to send a message *m*, then the FP of the message (its communicative act) should hold and likewise for the receiver to believe the RE of the message on receiving it. Given the above system for beliefs and intentions, we specify the semantics of the speech-acts and states in the CNP in terms of the intentions of a message sender and the receiver's beliefs.

6 The Semantics of CNP Actions

We analyse the semantics of the most commonly used CAs, as given by FIPA in the SL language, and discuss the incorrectness of these semantics with respect to the intended meaning of the CA. As a remedy, we provide a simpler and more intuitive semantics for the CAs in table 2. Let *s* and *r* denote the sender and receiver respectively.

Table 2: **BIS Semantics for FIPA CAs**

CA	**FP**	**RE**
$s.inform(r,\phi)$	$B_s\phi$	$B_r\phi$
$s.propose(r,\gamma)$	$B_s(B_sI_rdone(s.\gamma) \rightarrow I_sdone(s.\gamma))$	$B_r(B_sI_rdone(s.\gamma) \rightarrow I_sdone(s.\gamma))$
$s.accept(r,\gamma)$	$B_s(B_rI_sdone(r.\gamma) \rightarrow I_rdone(r.\gamma))$, $B_sI_sdone(r.\gamma)$	$B_rI_sdone(r.\gamma)$
$s.reject(r,\gamma)$	$B_s(B_rI_sdone(r.\gamma) \rightarrow I_rdone(r.\gamma))$, $B_s\neg I_sdone(r.\gamma)$	$B_r\neg I_sdone(r.\gamma)$
$s.request(r,\gamma)$	$B_sI_sdone(r.\gamma)$	$B_rI_sdone(r.\gamma)$
$s.agree(r,\gamma)$	$B_sI_rdone(s.\gamma)$, $B_sI_sdone(s.\gamma)$	$B_rI_sdone(s.\gamma)$
$s.refuse(r,\gamma)$	$B_sI_rdone(s.\gamma)$, $B_s\neg I_sdone(s.\gamma)$	$B_r\neg I_sdone(s.\gamma)$
$s.cfp(r,\gamma)$	$B_sI_s(done(r.propose(s,\gamma)) \vee done(r.refuse(s,\gamma)))$	$B_rI_s(done(r.propose(s,\gamma)) \vee done(r.refuse(s,\gamma)))$

6.1 Speech Acts Semantics in BIS Semantics

Henceforth, we refer to the SL semantics as "FIPA semantics" and we refer to our proposed revised semantics as "BIS" (Belief Intention Semantics) semantics. As in FIPA SL, we give the preconditions (FP) and postconditions (RE) holding, respectively, before sending and after receiving a CA. In general, the FP of a CA includes the intention of the sender for conveying that CA and the RE of a CA includes the receiver's beliefs. To be compatible with the way FIPA semantics are expressed, using axiom (A2), we prefix the intentions of a sender with its beliefs about those intentions, (e.g. $B_i I_i B_j \alpha$ instead of $I_i B_j \alpha$). Table 2 presents our BIS semantics for most of the FIPA CAs. We discuss below the semantics of some of the salient CAs.

6.1.1 *s.inform(r,ϕ).*

s informs *r* that ϕ holds.

Fipa *inform*	FP:$B_s\phi \wedge \neg B_s(Bif_r\phi \vee Uif_r\phi)$ RE: $B_r\phi$
BIS *inform*	FP: $B_s\phi$ RE:$B_r\phi$

In the FIPA semantics, the FP includes the fact that the sender believes ϕ and the RE that the receiver believes ϕ. As mentioned in section 5.2, $B_s\phi$ is not strong enough since we need to represent the intention of *s* to send the message. We could express this as $B_s I_s B_r \phi$, the sender intends the receiver to believe ϕ. By the sincerity and the trust axioms, in the BIS semantics (as shown in table 2), the FP and RE respectively simplify to $B_s\phi$ and $B_r\phi$.

6.1.2 *s.propose(r,γ).*

s proposes *r* for *s* itself to do γ.

Fipa *propose*	FP: $B_s\alpha \wedge \neg B_s(Bif_r\alpha \vee Uif_r\alpha)$ RE: $B_r\alpha$ where, $\alpha = I_r done(<s,\gamma>) \rightarrow I_s done(<r,\gamma>)$
BIS *propose*	FP: $B_s(B_s I_r done(s.\gamma) \rightarrow I_s done(s.\gamma))$ RE: $B_r(B_s I_r done(s.\gamma) \rightarrow I_s done(s.\gamma))$

In the FIPA semantics FP, the sender *s* believes that if *r* intends *s* to do γ, then *s* will intend it. However, *s* may not know what *r* intends and therefore cannot consequently infer that it should intend to do γ. For *s* to be aware that *r* intends $done(s.\gamma)$, it must have received an accept to its proposal. As such, the FIPA semantics specifies *s* adopts an intention by being privy to the individual beliefs of *r*. In our BIS semantics, both the FP and the RE specify that *s* (the proposer) believes that *r* intends $done(s.\gamma)$, for *s* to adopt the same intention. Therefore, $B_s I_r done(s.\gamma)$ is the premise for *s* to adopt the intention to do γ.

6.1.3 *s.accept(r,γ).*

s sends an accept proposal to *r*.

Fipa	FP: $B_s\alpha \wedge \neg B_s(Bif_r\alpha \vee Uif_r\alpha)$
accept	RE: $B_r\alpha$, where $\alpha = I_s done(<r,\gamma>)$.
BIS	FP: $B_s(B_r I_s done(r.\gamma) \rightarrow I_r done(r.\gamma))$,
accept	$B_s I_s done(r.\gamma)$
	RE: $B_r I_s done(r.\gamma)$

The FIPA semantics for *accept*ing a proposal do not consider the context of sending an accept. As FP, *s* believes that it intends *r* to do γ. There is no notion in either the FP or in the RE, that *s* is accepting a proposal that *r* must have sent. These FP and RE could also hold in other speech-acts such as *tell* and does not distinguish an *accept-proposal* from them. In our BIS semantics, we also include that both sender and receiver are aware that *r* sent a proposal previously and it is up to *s* to accept it. Our FP and RE also specify the context of the CA, this being an acceptance, there was a proposal before. The other part is the choice of the sender to intend the receiver to do γ.

The same remarks as for *accept-proposal* apply to the FIPA semantics for *reject-proposal*. The BIS semantics for *reject* can be found in table 2

6.1.4 *s.cfp(r,γ).*

s sends a call for proposal to *r* to do γ. In the FIPA semantics, the FP for a call for proposal includes that both sender and receiver intend for the receiver to perform the request. However, these intentions are premature given that *r* has yet to propose and *s* to accept for *r* to do γ. It does not leave the possibility for refusal or rejection. The rest of the semantics for *cfp* is so complicated that its meaning is unclear.

In our semantics, a call for proposal from *s* to *r* is equivalent to a request from *s* to *r* for *r* to make a proposal to *s*. Thus *s.cfp(r, γ)* is equivalent to *s.request(r, r.propose(s,γ))*.

Using our BIS semantics for *s.request(r, γ)*, as shown in table 2, we can specify a call for proposal by *s* as having FP $B_s I_s(done(r.propose(s,\gamma)) \vee done(r.refuse(s,\gamma)))$. This means that *s* intends that *r* either sends a proposal or refuses to do γ (because may be *r* cannot do γ). In turn, the RE of a call for proposal is that *r* believes *s* intends *r* to make a proposal or to refuse.

6.1.5 *s.refuse(r,γ).*

s sends a refusal to *r* for *r* to do γ. Again for the refusal CA, the FIPA semantics are obscure. For example, the FP given by the FIPA semantics is:
$B_s\neg Feasible(<s,\gamma>) \wedge B_s(B_r Feasible(<s,\gamma>) \vee$
$U_r Feasible(<s,act>)) \wedge B_s\alpha \wedge \neg B_s(Bif_r\alpha \vee Uif_r\alpha)$
where $\alpha = \beta \wedge \neg done(<i,\gamma>) \wedge \neg I_s done(<s,\gamma>)$.
β is the reason for the refusal and γ is the action being refused. The predicate *Feasible* is unclear and the formula α is hard to understand. In the BIS semantics, the precondition for a refusal is that the sender *s* believes that *r* intends *s* to do γ and *s* does not

intend to do so. The RE is that the receiver then believes that the sender does not intend γ.

6.2 Internal Actions

There are two internal actions in the CNP protocol — *M.deliberate(C,A,p)* from a *proposed* state and *i.p* from an *accepted* state. The process *i.p* expresses that agent *i* executes *p* and it semantics are given in terms of the semantics for extended PDL. In the process $M.deliberate(C, A, Act)$, the set *C* contains those agents which sent a proposal, *A* contains the set of agents whose proposals *M* will accept and *A*ct is the set of proposals subscripted with their corresponding proposer. A manager internally performs the $M.deliberate(C, A, Act)$ process to select which proposals to accept. The semantics can be found in table 3. The precondition requires that *M* believes that the set *C* contains those agents which sent a proposal. The postcondition specifies that after a *deliberate* action, the set *A* contains the contractors whose proposals *M* will accept and the set *(C-A)*, those who will be sent rejections.

7 The Semantics of CNP States

The interaction states of an IP specify the beliefs of an agent or group of agents. Given this, the interaction states in the CNP can be grouped into three types: public, shared and individual. Public states are believed by all the agents, shared states are mutually believed by a particular subset of the group, and individual states are the beliefs of one agent that others are unaware of. Internal actions are assumed to succeed. In the CNP, state s_i is equivalent to $done(a_i)$. For example, the state *cfped(M, G, p)* can be written as *done(M.cpf(G,p))* and likewise for the other states. For the sake of readability, in figure 2 we prefer to name a state as the past tense of an action leading to it, instead of a parameterised *done*. Let a group of contractors be denoted by *G* and the manager by *M*. Let $E_G\alpha$ be read as everyone in a group of agents, *G*, believes α. We specify below the semantics of the interaction states of the CNP, which together with the semantics of the processes in the CNP, constitute the semantics of the protocol.

7.1 Public States

The public states in the CNP are *cfped*, *timedout*, *open* and *closed* (see figure 2). These states are believed by the manager and all the contractors in *G*. The semantics of the state *cfped(M, G, p)* is that everyone in the group *GM*, (where $GM = G \cup \{M\}$) believes *done(M.cpf(G,p))*. That is:
$E_{GM} \forall i \in G(done(M.request(i, i.propose(M, p))))$. This entails that everyone believes the FP and RE of a call for proposal:
$E_{GM} I_s(done(r.propose(s, p)) \vee done(r.refuse(s, p)))$.

Similarly the semantics of the other public states are specified in terms of the beliefs of the group *GM*.

7.2 Shared States

In the CNP, these shared states are mutually believed by the manager and a contractor. For example, only the manager and a contractor sending a proposal believe and mutually believe that this particular contractor has sent the manager a proposal. The shared states of the CNP are *proposed, accepted, rejected, cancelled, refused, informed and failed* (see figure 2). Their semantics are given in terms of the beliefs of the manager and a contractor. We explain the semantics of the *proposed* and *accepted* states. The semantics of the other shared states are given in figure 3.

7.2.1 *proposed(C,p).*

Can be re-written as $\forall i \in C(done(i.propose(M, p_i))$. The beliefs between *M* and each *i* in *C* are given below. The FP and RE of the *propose* CA are:
$B_M(B_i I_M done(i.p_i) \rightarrow I_i done(i.p_i))$ and
$B_i(B_i I_M done(i.p_i) \rightarrow I_i done(i.p_i))$ leading to the shared belief between *i* and *M*:
$\forall i \in C(E_{\{M,i\}}(B_i I_M done(i.p_i) \rightarrow I_i done(i.p_i)))$.

7.2.2 *accepted(A, p).*

Every contractor *i* in the set *A* has been sent an acceptance message from *M*. This can be re-written as $\forall i \in A(done(M.accept(i, p_i))$.

From the FP and RE effect of *accept* CA, we have $\forall i \in A$:
$E_{\{M,i\}}(B_i I_M done(i.p_i) \rightarrow I_i done(i.p_i))$, *i* and *M* believe *i* previously sent a proposal to *M*.
$E_{\{M,i\}} I_M done(i.p_i)$ holds i.e. both *i* and *M* believe *M* has accepted *i*'s proposal, leading to the belief $E_{\{M,i\}} I_i done(i.p_i)$, both believe that *i* has adopted the intention to do γ.

7.3 Individual States

There are two individual states in the CNP process, *to-be-accepted* and *completed*, as the private state of the manager and a contractor respectively (see figure 2). The state *to-be-accepted(C,A, Act)* expresses the private belief of *M* that it will send an acceptance to all contractors in *C*. The following holds in the manager's beliefs for the *to-be-accepted(C,A, Act)* state:

- $B_M \forall i \in C(B_i I_M done(i.p_i) \rightarrow I_i done(i.p_i))$, *M* believes that all agents in *C* sent it a proposal.
- $B_M \forall i \in A(I_M done(i.p_i) \wedge I_M B_i I_M done(i.p_i))$. By axioms (A6) and (A1), this is equivalent to $B_M \forall i \in A(I_M done(i.p_i))$ *M* intends that all agents in *A* execute their proposal and *M* intends to let them know about its acceptance.
- $B_M \forall i \in (C - A)(\neg I_M done(i.p_i) \wedge I_M B_i \neg I_M done(i.p_i))$, likewise for the agents that *M* has decided to reject. Again by axioms (A6) and (A1), this is equivalent to $B_M \forall i \in (C - A)(\neg I_M done(i.p_i))$.

8 Proving Properties of the CNP

We validate our semantics for the CNP and its CAs by proving useful properties of the protocol. We do this by reasoning about the possible paths in the CNP. To this end, figure 5 shows an interpretation of the CNP from its start with a call for proposal to its completion with *refusals*, *cancellations* or *informs*. Figure 5 is derived from the CNP statechart in figure 2 and includes all possible paths in the execution of the CNP (apart from timeout). Let us refer to the states s_i in figure 2 as interaction states, and to the states S_i in figure 5 as execution states. An execution state, S_i, represents the beliefs of the manager and the contractors, given by the RE of the action leading to S_i, the FP of the next action, and the group beliefs in the interaction state s_i. The path S_0 to S_6 is the longest execution path in the CNP, and the beliefs holding at these states are given in table 3. For example, the action *cfp* leads to state S_1 in figure 5, and thus in table 3, the state S_1 gives the RE of *cfp*, the group's beliefs of the *cfped* state, and the FP of the next action *propose*. In table 3, let $GM = G \cup \{M\}$. Let α_i denote $B_i I_M done(i.p) \rightarrow I_i done(i.p)$ and let α_j denote $B_j I_M done(j.p) \rightarrow I_j done(j.p)$.

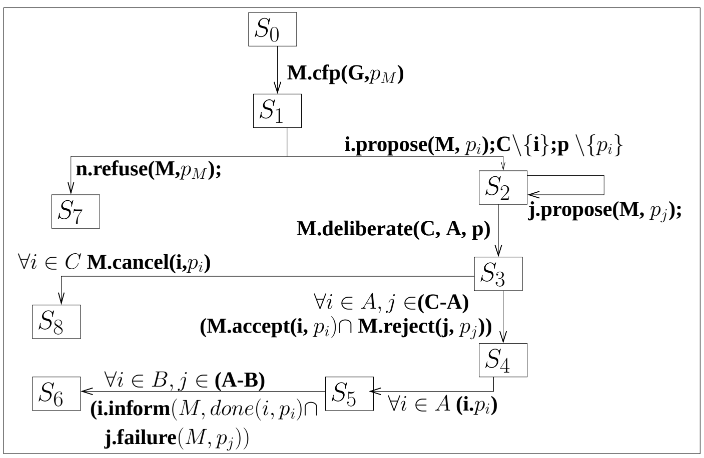

Figure 5: Flowchart showing CNP Interpretation

8.1 Termination

We prove that any interpretation of the CNP will terminate.

Theorem 1. *In a CNP between a manager M and a group of contractors G, the formula ($cfped(M, G, p_M) \rightarrow [Paths_{ALL}]closed$) holds, where the process $Paths_{ALL}$ expresses all complete paths of execution in our framework.*

Proof. We prove that the *closed* state is eventually reached from the *cfped* interaction state. The simplest path is a timeout, which terminates. Let the process ρ_i lead to the interaction state s_i and the execution state S_i. The CNP defines that process ρ_i may be

Table 3: **Execution Paths from State S_0 to S_6**

S_0	FP_{cfp}	$\forall i \in G(B_M I_M(done(i.propose(M,p)) \vee done(i.refuse(M,p))))$
S_1	RE_{cfp}	$\forall i \in G(B_i I_M(done(i.propose(M,p)) \vee done(i.refuse(M,p))))$
	$State_{cfped}$	$\forall i \in G(E_{GM} I_M(done(i.propose(M,p)) \vee done(i.refuse(M,p))))$
	$FP_{propose}$	$B_i \alpha_i$
S_2	$RE_{propose}$	$B_M \alpha_i$
	$State_{proposed}$	$\forall i \in C(E_{\{M,i\}} \alpha_i)$
	$Pre_{deliberate}$	$\forall i \in C(B_M \alpha_i)$
S_3	$Post_{deliberate}$	$A \subseteq C \wedge \forall i \in A(B_M I_M B_i I_M done_i)$, $\forall j \in (C-A)(B_M I_M B_j \neg I_M done_j)$
	to-be-accepted	$\forall i \in C(B_M \alpha_i), \forall i \in A(B_M(I_M done_i \wedge I_M B_i I_M done_i))$, $\forall j \in (C-A)(B_M(\neg I_M done(j, p_j) \wedge I_M B_j \neg I_M done_j))$
	$FP_{M.accept(i,p_i)}$	$B_M \alpha_i, B_M I_M done(i.p_i)$
	$FP_{M.reject(j,p_j)}$	$B_M \alpha_j, B_M \neg I_M done(j.p_j)$
S_4	$RE_{M.accept(i,p_i)}$	$B_i I_M done(i.p_i)$
	$RE_{M.reject(j,p_j)}$	$B_j \neg I_M done(j.p_j)$
	$State_{accepted}$	$E_{\{M,i\}} \alpha_i, \forall i \in A(E_{\{M,i\}} I_M done(i.p_i))$
	$State_{rejected}$	$\forall j \in$ (C-A)$(E_{\{M,j\}} \alpha_j \wedge E_{\{M,j\}} \neg I_M done_j, E_{\{M,j\}} \neg I_j done_j)$
	$Pre_{i.p}$	Beliefs of *accepted* state
S_5	$Post_{\forall i \in A(i.p)}$	Beliefs of *completed* state
	$State_{completed}$	$\forall i \in B(B_i(done(i.p_i) \wedge I_i B_M done_i)$ $\forall j \in$ (A-B)$\wedge B_j(\neg done_j \wedge \neg I_j done_j \wedge I_j B_M \neg done_j))$
	$FP_{i.inform}$	$B_i done(i.p_i)$
	$FP_{j.failure(M,p_j)}$	$B_j \neg I_j done_j$
S_6	$RE_{i.inform}$	$B_M done(i.p_i)$
	$RE_{j.failure(M,p_j)}$	$B_M(\neg done(j.p_j) \wedge \neg I_j done_j)$
	$State_{informed}$	$\forall i \in B(E_{\{M,i\}} done(i.p_i))$
	$State_{failed}$	$\forall j \in (A-B)(E_{\{M,j\}}(\neg done_j \wedge \neg I_j done_j))$

followed by the process ρ_{i+1}. Let the notations in table 3 be used (e.g. for α_i). Proving termination for all processes in $Paths_{ALL}$ implies proving that all paths in figure 5 terminate. Thus, we prove that all actions in the paths in figure 5 are feasible in their source state (i.e, the FP of all processes ρ_{i+1} may hold after action ρ_i and in interaction state s_i). The premise is that the CNP has been started with a call for proposals. The RE_{cfp} (RE of cpf) holds, stating that contractors should either reply with a refusal or a proposal. But this is what is required by the CNP to trigger the next state, so the process proceeds to the state *proposed* or *refused*, since the RE_{cfp} renders it possible. Since *refused* is a sub-state of *closed*, all paths from *refuse* terminate. So using table 3, we now prove that the paths following a *propose* action terminate. That is, for all actions ρ_i, after execution state S_1, the $FP_{\rho_{i+1}}$ is possible from the interaction state s_i and the RE_{ρ_i}. This can be seen in table 3 where the pre-condition of *deliberate* holds from the $RE_{propose}$.

The FP of the action $\forall i \in A(M.accept(i, p_i))$ holds since the state *to-be-accepted* includes the belief $\forall i \in C(B_M \alpha_i)$ and $\forall i \in A(B_M I_M done_i)$.

Similarly the FP of the *reject* action holds from the beliefs in the *to-be-accepted* state. Thus, both acceptance and rejection processes can occur. Then the pre-condition of the next action $i.p$ hold by virtue of being the beliefs of the resulting interaction state from an *inform* or *failure*.

From table 3 it can be seen that the FP of both *accept* and *reject* can be derived from the state *completed*, leading to the sub-states of *closed*. □

Corollary 1. *After a call for proposal, the FP of all actions may hold from the FP and RE of the previous action. That is,* $FP_{\rho_{i+1}}$ *from* FP_{ρ_i} *and the* RE_{ρ_i}

We can also show there are no deadlocks in the CNP interpretation. The corollary holds because when proving theorem 1, we proved $FP_{\rho_{i+1}}$ is possible from the interaction state s_i and the RE_{ρ_i}. From our semantics, interaction state s_i is itself defined from the FP and RE of the action ρ_i leading to it.

8.2 A Failed or Succeeded CNP

We can also show that a CNP always terminates with the beliefs of whether the CNP process has satisfied the goal of the interaction:

Theorem 2. *The interpretation of the CNP terminates with either the shared belief between a manager and a contractor i of either* $done_i$ *or* $\neg done_i$*, or the group beliefs of* $\neg done$*.*

Proof. By theorem 1, a CNP interpretation always terminates. $\neg done$ obviously holds in the *timedout* state. Terminal states are *refused, timedout, cancelled* and *(informed* $\wedge$ *failed)*. Section 7 and table 3 both show that $\neg done$ holds in states *refused* and *cancelled*. It can also be seen that in the *accepted* and *rejected* states, $E_{\{M,i\}} done_i$ and $E_{\{M,j\}} \neg done_j$ respectively hold. Thus, all accepted contractors *i* believe $done_i$ and all rejected contractors *j* believe $\neg done_j$, while the manager appropriately believes $done_i$ and $\neg done_j$. □

8.3 Consistent Joint Beliefs in a Group

Consistent joint belief about μ in group *G* entails that everyone in *G* believes μ and no-one believes $\neg\mu$. Below we show that for public and shared states, there are some beliefs that are consistent between the agents in the group (for public states) or sub-group (for shared states). Thus, there is a state of affairs which every agent in the group (or sub-group) believes.

Theorem 3. *In the CNP interpretation, for all public and shared states, there are some consistent joint beliefs in, respectively, the group or sub-group. That is, for all states public to G, and states shared between* G_{sub}*:*

$$\exists \mu (E_G \mu \wedge (\neg \exists i \in G(B_i \neg \mu))) \wedge$$
$$\exists \beta (E_{G_{sub}} \beta \wedge (\neg \exists i \in G_{sub}(B_i \neg \beta)))$$

Proof. In the public *cfped* state, it can be seen that:
$\forall i \in G(E_{GM} I_M(done(i.propose(M,p)) \vee done(i.refuse(M,p))))$.
Regarding shared states in the CNP, they have been formulated in table 3 in such a way to show the joint beliefs between the manager and each contractor *i*. For example, $\forall i \in B(E_{\{M,i\}} done(i.p_i))$ in the *informed* state. Since individual beliefs are consistent, then if everyone in that sub-group believes $done(i.p_i)$, no-one will believe the contrary. The same is true for the other shared states *accepted, failed, rejected* and *proposed*. □

9 Conclusions and Future Work

We believe there is a lack of consensus on a suitable ACL for agent interaction because of the bewildering array of approaches to formalising an ACL's semantics. Likewise, there is a similarly strong need for formal specification and verification of interaction protocols and their semantics. To highlight these needs and in attempting to satisfy them, this paper uses the contract net protocol as a non-trivial case study. In our framework, we formulate axioms for reasoning about an agent's beliefs about its intentions and present a simplified and revised semantics for the FIPA communicative acts that appear in the contract net protocol. We accompany these ACL semantics with those of the states and internal actions in the contract net protocol in order to obtain (for the first time) a complete semantics for that protocol. In so doing, we can prove properties when interpreting the protocol such as termination and consistency in joint beliefs. Even though the case study has raised several issues about ACL and IP semantics, it is still incomplete and we intend that future work analyses other interesting open issues. Future work includes relaxing the assumptions detailed in section 5.2 and analysing properties such as liveness, completeness, complexity and decidability. A denotational semantics for the speech-acts can be specified in addition to the given BDI semantics and these semantics may be combined with our previous work [9] in modeling agent interactions in imperfect communication environments for an analysis of the performance of agent interactions in realistic environments.

References

[1] J. L. Austin. *How to Do Things with Words.* Oxford University Press, 1962.

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

[3] Y. Finin, T.and Labrou and J. Mayfield. KQML as an agent communication language. In *Software Agents, Jeff Bradshaw (Ed)*, pages 267–273, 1995.

[4] Foundation for Intelligent Physical Agents, http://www.fipa.org. *FIPA Agent Communication Language Specification*.

[5] R. Goldblatt. *Logics of Time and Computation*. CSLI, 1987.

[6] F. Guerin and J. Pitt. Agent communication frameworks and verification. In *AAMAS 2002 Workshop on Agent Communication Languages, Bologna, LNCS volume 2650*, 2002.

[7] M. Nowostawski, M. Purvis, and S. Cranefield. A layered approach for modelling agent conversations. In *Proc. 2nd Int. Work. on Infrastructure for Agents, MAS, and Scalable MAS,*, 2001.

[8] S. Paurobally. *Rational Agents and the Processes and States of Negotiation*. PhD thesis, Imperial College, 2002.

[9] S. Paurobally, R. Cunningham, and N. R. Jennings. Ensuring consistency in joint beliefs of interacting agents. In *Proc. AAMAS*, pages 662–669, 2003.

[10] J. Pitt and A. Mamdani. Communication protocols in mas. In *Work. on Specifying and Implementing Conversation Policies*, pages 39–48, 1999.

[11] D. Sadek. A study in the logic of intentions. In *3rd Conf. on Principles of Knowledge Representation and Reasoning*, pages 462–473, 1992.

[12] J. R. Searle. *Speech acts: An essay in the philosophy of language*. Cambridge University Press, 1969.

[13] M. Singh. Agent communication languages: Rethinking the principles. *IEEE Computer*, 31(12):40–47, 1998.

[14] I. Smith, P. Cohen, J. Bradshaw, Greaves M., and H. Holmack. Designing conversation policies using joint intention theory. In *ICMAS*, pages 269–276, 1998.

[15] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1981.

[16] M. Wooldridge. Semantic issues in the verification of agent communication languages. *Journal of Autonomous Agents and MAS*, 3(1):9–31, 2000.

[17] P. Yolum and M. Singh. Flexible protocol specification and execution: Applying event calculus planning using commitments. In *Proc. AAMAS*, pages 527–534, 2002.