On the Semantics of Conditional Commitment

Shakil M. Khan
Department of Computer Science and Engineering
Toronto, ON, Canada M3J 1P3
skhan@cs.yorku.ca

Yves Lespérance
Department of Computer Science and Engineering
Toronto, ON, Canada M3J 1P3
lesperan@cs.yorku.ca

ABSTRACT

In this paper, we identify some problems with current formalizations of conditional commitments, i.e. commitments to achieve a goal if some condition becomes true. We present a solution to these problems. We also formalize two types of communicative actions that can be used by an agent to request another agent to achieve a goal or perform an action provided that some condition becomes true. Our account is set within ECASL [9], a framework for modeling communicating agents based on the situation calculus.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Theory, Design

Keywords

Agent communication, Semantics of requests, Conditional intentions, Conditional commitments, Logics for agent systems, Formal models of agency

1. INTRODUCTION

In recent years, the importance of agent communication in multiagent systems has been widely recognized. As a result, many researchers have developed communicative multiagent frameworks [3, 5, 7, 15, 18, 23, 25] and attempted to formalize various types of communicative actions in these frameworks. One important concept in these is the notion of *conditional commitment*. A conditional commitment is a commitment to achieve some goal if some condition becomes true (e.g. a commitment to ship some goods when payment of an agreed to amount arrives). Conditional requests are requests that seek to have the addressee acquire a conditional commitment. Any multiagent framework that deals with negotiation and cooperation ought to handle conditional commitments. Unfortunately, most definitions found in the literature (in [5, 28, 23, 1], for example) are inadequate: they either define conditional commitments as disjunctive goals, which makes the agents under-committed to the conditional goal, or define them as conjunctive goals, which renders the agent over-committed.

We will go over some examples to point out the problems associated with the disjunctive and the conjunctive accounts of conditional commitment. In these, we use the following modal operators: $\Diamond\phi$, i.e. ϕ eventually holds, Happens(α), i.e. the action α is performed next, ϕ Until ψ, i.e. eventually ψ becomes true, and as long as ψ is false, ϕ holds, and Before(ψ, ϕ), i.e. if ψ eventually becomes true, then ϕ becomes true before ψ does. The formal semantics of these operators are given in Section 2. In the following, we formulate these examples using internal/mental states semantics for communication acts, and use the terms 'intention' and 'commitment' interchangeably.

In the *disjunctive account*, a conditional commitment to achieve some goal provided that some condition holds is modeled as a commitment to achieve the goal if the condition holds, i.e. as a simple material implication. For example, consider an online marketplace domain. Suppose that there are two agents, a seller agent slr, and a buyer agent byr. If we use a disjunctive account, slr's conditional commitment to ship some goods to byr on the condition that byr pays can be modeled as follows:

$$\begin{aligned}&\text{CondInt}_{\text{dis}}(slr, GetPaid, \text{Happens}(shipGoods(slr, byr)))\\ &\doteq \text{Int}(slr, \neg\Diamond GetPaid \vee\\ &\qquad [\neg GetPaid \text{ Until}\\ &\qquad\quad (GetPaid \wedge \text{Happens}(shipGoods(slr, byr)))]).\end{aligned}$$

This says that slr's conditional commitment to ship the goods when byr pays amounts to slr having the intention that byr eventually pays and after that she ships the goods, if byr eventually pays (as mentioned earlier, the Until construct in the goal above implies that $\Diamond GetPaid$ and $\Diamond$-Happens($shipGoods(slr, byr)$)). One problem with this account of conditional intention is that there is a counter-intuitive way to satisfy the conditional intention, namely, the agent may commit to the triggering condition remaining false and deliberately perform some action that makes it remain false. Thus, in the example, to satisfy her conditional intention, slr may intentionally perform some action to stop byr from paying her, such as blocking debits from byr. In other words, there is nothing in this formalization of conditional intention that stops slr from intending not to

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

get paid and not to send the goods. However, this is counter-intuitive and a model of conditional commitment should not support this. Thus, with the disjunctive account of conditional commitment, the agent seems under-committed to the goal. Examples of accounts in the literature that formalize conditional commitments as disjunctive goals are [23] and [1].

In the *conjunctive account*, a conditional commitment to achieve a goal provided that a condition holds is modeled as a temporally ordered conjunctive commitment to the triggering condition and the conditional goal, where the triggering condition is achieved first. Although this model may seem appropriate in many cases, it often leads to problems. For example, suppose that *slr* has the conditional commitment to ship a replacement unit provided that *byr* reports and returns a defective good. If we use a conjunctive account, this can be modeled as follows:

$$\begin{aligned}&\text{CondInt}_{\text{con}}(slr, DefGoodRet,\\ &\qquad\qquad \text{Happens}(shipRepl(slr, byr))) \doteq\\ &\text{Int}(slr, \text{Before}(\text{Happens}(shipRepl(slr, byr)), DefGoodRet)\\ &\qquad \wedge \Diamond \text{Happens}(shipRepl(slr, byr))).\end{aligned}$$

This says that *slr*'s conditional commitment to ship a replacement unit provided that *byr* returns a defective good can be modeled as *slr*'s intention that *byr* returns a defective good before *slr* ships a replacement unit, and eventually *slr* ships a replacement unit. Note that, according to this definition, since *slr* has the intention that the defective product is returned before she ships the replacement unit, and that she eventually ships the replacement unit, it follows that *slr* has the intention that *byr* eventually returns a product, i.e. $\text{Int}(slr, \Diamond DefGoodRet)$. So *slr* may deliberately perform some action, such as shipping a defective good in the first place, to achieve this intention. Thus, the conjunctive account of conditional commitment results in over-committed agents. Both [5] and [28] formalize conditional commitments as conjunctive goals.

In this paper, we propose a solution to these problems (the *under/over-commitment problems*, henceforth). Our solution involves using an additional constraint with the disjunctive account to eliminate the under-commitment problem. We use the Extended Cognitive Agent Specification Language (ECASL) [9] as our base formalism for this. Our account is formulated for internal/mental states semantics for communication acts. Nevertheless, the same issues arise for public/social-commitment semantics (as discussed in Section 5). Once again, in this paper, we will use the terms 'intention' and 'commitment' interchangeably.

Using our definition of conditional intention, we then prove that a conditional intention becomes an ordinary intention when the associated condition is true, and that agents correctly introspect about their conditional intentions. We also formalize two types of communicative actions that can be used by an agent to request another agent to achieve a goal or perform an action provided that some confdition becomes true. Then we prove that performing such a conditional request leads to a conditional intention under the right conditions.

The paper is organized as follows: in the next section, we outline the ECASL framework. In Section 3, we present our model of conditional commitment and discuss some of its properties. In Section 4, we present some communicative acts that allow agents to make requests that result in conditional commitments. In Section 5, we compare our approach to previous work on conditional commitments. Finally in Section 6, we summarize our results and discuss possible future work.

2. ECASL

The Extended Cognitive Agent Specification Language (ECASL) [9], an extension of CASL [22, 24], is a framework for specifying and verifying complex communicating multi-agent systems that incorporates a formal model of means-ends reasoning. In this section, we outline the part of ECASL that is needed for our formalization of conditional commitment.

In ECASL, agents are viewed as entities with mental states, i.e., knowledge and goals, and the specifier can define the behavior of the agents in terms of these mental states. ECASL combines a declarative action theory defined in the situation calculus with a rich programming/process language, ConGolog [4]. Domain dynamics and agents' mental states are specified declaratively in the theory, while the agents' behavior is specified procedurally in ConGolog.

In ECASL, a dynamic domain is represented using an action theory [17] formulated in the situation calculus [12], a (mostly) first order language for representing dynamically changing worlds in which all changes are the result of named actions. ECASL uses a theory D that includes the following set of axioms:

- action precondition axioms, one per action a characterizing $Poss(a, s)$,
- successor state axioms (SSA), one per fluent, that encode both effect and frame axioms and specify exactly when the fluent changes [16],
- initial state axioms describing what is true initially including the mental states of the agents,
- axioms identifying the agent of each action,
- unique name axioms for actions, and
- domain-independent foundational axioms describing the structure of situations [10].

Within ECASL, the behavior of agents is specified using the notation of the logic programming language ConGolog [4], the concurrent version of Golog [11]. A typical ConGolog program is composed of a sequence of procedure declarations, followed by a complex action. Complex actions can be composed using constructs that include primitive actions, waiting for a condition, sequence, nondeterministic branch, nondeterministic choice of arguments, conditional branching, while loop, procedure call, nondeterministic iteration, concurrent execution with and without priorities, and interrupts. To deal with multiagent processes, primitive actions in ECASL take the agent of the action as argument.

The semantics of the ConGolog process description language is defined in terms of *transitions*, in the style of structural operational semantics [14]. The overall semantics of a program is specified by the *Do* relation:

$$Do(\delta, s, s') \doteq \exists \delta' \cdot (Trans^*(\delta, s, \delta', s') \wedge Final(\delta', s')).$$

$Do(\delta, s, s')$ holds if and only if s' can be reached by performing a sequence of transitions starting with program δ in s, and the remaining program δ' may legally terminate

in s'. Here, $Trans^*$ is the reflexive transitive closure of the transition relation $Trans$.[1]

The situation calculus underlying ECASL is a branching time temporal logic, where each situation has a linear past and a branching future. In the framework, one can write both state formulas and path formulas. A state formula $\phi(s)$ takes a single situation as argument and is evaluated with respect to that situation. On the other hand, a path formula $\psi(s_1, s_2)$ takes two situations as arguments and is evaluated with respect to the interval (finite path) $[s_1, s_2]$. A state formula ϕ may contain a placeholder constant *now* that stands for the situation in which ϕ must hold. $\phi(s)$ is the formula that results from replacing *now* by s. Similarly, a path formula ψ may contain the placeholder constants *now* and *then* that stand for the situations that are the endpoints of the interval $[now, then]$ over which ψ must hold. $\psi(s_1, s_2)$ denotes ψ with s_1 substituted for *now* and s_2 substituted for *then*. Where the intended meaning is clear, we sometimes suppress the placeholder(s).

ECASL allows the specifier to model agents in terms of their mental states by including operators to specify agents' information (i.e., their knowledge), and motivation (i.e., their goals or intentions). We use state formulas within the scope of knowledge, and path formulas within the scope of intentions. Following [13, 20], ECASL models knowledge using a possible worlds account adapted to the situation calculus. $K(agt, s', s)$ is used to denote that in situation s, *agt* thinks that she could be in situation s'. s' is called a *K-alternative situation* for *agt* in s. Using K, the knowledge or belief of an agent, $\text{Know}(agt, \phi, s)$, is defined as $\forall s'(K(agt, s', s) \supset \phi(s'))$, i.e. *agt* knows ϕ in s if ϕ holds in all of *agt*'s K-accessible situations in s. In ECASL, K is constrained to be reflexive, transitive, and Euclidean in the initial situation to capture the fact that agents' knowledge is true, and that agents have positive and negative introspection. As shown in [20], these constraints then continue to hold after any sequence of actions since they are preserved by the successor state axiom for K.

ECASL supports knowledge expansion as a result of sensing actions [20] and some *informing* communicative actions. Here, we restrict our discussion to knowledge expansion as a result of *inform* actions. The preconditions of *inform* are as follows:

$$Poss(inform(inf, agt, \phi), s) \equiv \text{Know}(inf, \phi, s) \\ \wedge \neg\text{Know}(inf, \text{Know}(agt, \phi, now), s).$$

In other words, the agent *inf* can inform *agt* that ϕ, iff *inf* knows that ϕ currently holds, and does not believe that *agt* currently knows that ϕ. The successor state axiom (SSA) for K can be defined as follows:

$$K(agt, s^*, do(a, s)) \equiv \\ \exists s'.\ K(agt, s', s) \wedge s^* = do(a, s') \wedge Poss(a, s').$$

This says that after an action happens, every agent learns that it was possible and has happened. Moreover, if the action involves someone informing *agt* that ϕ holds, then *agt* knows this afterwards. This follows from the fact that it is a precondition of $inform(inf, agt, \phi)$ that *inf* knows that ϕ, that what is known must be true (i.e. K is reflexive), and that the SSA for K requires the agent to know that $Poss(a, s)$ after a happens in s. Note that this axiom only handles knowledge expansion, not revision.

ECASL also incorporates goal expansion and a limited form of goal contraction. Goals or intentions are modeled using an accessibility relation W over possible situations. The W-accessible situations for an agent are the ones where she thinks that all her goals are satisfied. W-accessible situations may include situations that the agent thinks are impossible (i.e. that do not have a predecessor that is K-related to the current situation), unlike Cohen and Levesque's [2] G-accessible worlds. But intentions are defined in terms of the more primitive W and K relations so that the intention accessible situations are W-accessible situations that are also compatible with what the agent knows, in the sense that there is a K-accessible situation in their history. This guarantees that agents' intentions are realistic, that is, agents can only intend things that they believe are possible. Thus we have:

$$\text{Int}(agt, \psi, s) \doteq \forall s', s^*. \\ [W(agt, s^*, s) \wedge K(agt, s', s) \wedge s' \leq s^*] \supset \psi(s', s^*).$$

This means that the intentions of an agent in s are those formulas that are true for all intervals between situations s' and s^* where the situations s^* are W-accessible from s and have a K-accessible situation s' in their past. Intentions are future oriented, and any goal formula will be evaluated with respect to a finite path defined by a pair of situations, a current situation *now* and an ending situation *then*. This formalization of goals can deal with both achievement goals and maintenance goals. An achievement goal ϕ is said to be eventually satisfied if ϕ holds in some situation between *now* and *then*, i.e., if $\Diamond(\phi, now, then)$, which is defined as $\exists s'.\ (now \leq s' \leq then \wedge \phi(s'))$.[2] In [21], Shapiro showed how positive and negative introspection of intentions can be modeled by placing some constraints on K and W. To make sure that agents' wishes and intentions are consistent, W is also constrained to be serial.

ECASL provides an intention transfer communication action, *request*, which is defined in terms of *inform*. This is somewhat similar to Herzig and Longin's account [7], where a request is defined as informing about one's intentions, and the requested goals are adopted via cooperation principles. The *request* action can be used by an agent to request another agent to achieve some state of affairs. Formally, we have:

$$request(req, agt, \phi) \doteq inform(req, agt, \text{Int}(req, \phi, now)).$$

The SSA for W which handles intention change in ECASL, has the same structure as a SSA for a domain dependent fluent. In the following, $W^+(agt, a, s^*, s)$ ($W^-(agt, a, s^*, s)$, resp.) denotes the conditions under which s^* is added to (dropped from, resp.) W as a result of the action a in s:

$$W(agt, s^*, do(a, s)) \equiv W^+(agt, a, s^*, s) \vee \\ (W(agt, s^*, s) \wedge \neg W^-(agt, a, s^*, s)).$$

An agent's intentions are expanded when it is requested

[1]Since we have predicates that take programs as arguments, we need to encode programs and formulas as first-order terms as in [4]. For notational simplicity, we suppress this encoding and use formulas and programs as terms directly.

[2]We sometimes use $\Diamond$ with a path formula ψ argument, in which case, we mean that ψ holds over some interval $[s, then]$ that starts at some situation s between *now* and *then*; see Table 1 for the formal definition.

something by another agent. After the $request(req, agt, \psi)$ action, *agt* adopts the goal that ψ, unless she has a conflicting goal or is not willing to serve *req* for ψ. Therefore, this action should cause *agt* to drop any paths in W where ψ does not hold. This is handled in W^-:

$$\begin{aligned} &W^-(agt, a, s^*, s) \doteq \\ &\quad [\exists req, \psi.\ a = request(req, agt, \psi) \\ &\quad \wedge Serves(agt, req, \psi, s) \wedge \neg \mathrm{Int}(agt, \neg\psi, s) \\ &\quad \wedge \exists s'.\ K(agt, s', s) \wedge s' \leq s^* \wedge \neg\psi(do(a, s'), s^*)]. \end{aligned}$$

A limited form of intention contraction is also handled in ECASL. Agents intentions are contracted as a result of a *cancelRequest* action. ECASL also incorporates a formal model of means-ends reasoning and commitment to rational plans to achieve intentions. See [9] for the details of these.

Table 1 shows some abbreviations that will be used throughout the paper.

Table 1: Some Definitions of Temporal Operators

1. $\Diamond(\psi, now, then) \doteq \exists s'.\ now \leq s' \leq then \wedge \psi(s', then)$,
2. $\Box(\psi, now, then) \doteq \neg\Diamond(\neg\psi, now, then)$,
3. $[\phi \text{ Until } \psi](now, then) \doteq \exists s'.\ now \leq s' \leq then \wedge \psi(s', then) \wedge \forall s''.\ now \leq s'' < s' \supset \phi(s'')$,
4. $\mathrm{Before}(\psi, \phi, now, then) \doteq \exists s'.\ now \leq s' \leq then \wedge \psi(s', then) \supset \exists s''.\ now \leq s'' < s' \wedge \phi(s'')$,
5. $\mathrm{E}_\Diamond(\phi, now) \doteq \exists s.\ now \leq s \wedge \phi(s)$,
6. $\mathrm{A}_\Box(\phi, now) \doteq \neg\mathrm{E}_\Diamond(\neg\phi, now)$,
7. $\mathrm{Happens}(a, now, then) \doteq do(a, now) \leq then$,
8. $\mathrm{Happens}_G(\delta, now, then) \doteq \exists s'.\ s' \leq then \wedge Do(\delta, s', then)$.

3. CONDITIONAL COMMITMENTS

Having presented our framework, we now return to our discussion about conditional commitments. Informally, an agent *agt* has a conditional commitment or intention that ψ on the condition that ϕ if *agt* intends to achieve ψ as soon as the condition ϕ holds. In our specification, we assume that ϕ is a state formula, whereas ψ is a path formula and can represent any kind of goal (achievement, maintenance, etc.). In other words, the trigger condition ϕ of a conditional intention takes a single situation *now* as argument, unlike the goal formula ψ, which takes two situations *now* and *then* as arguments.[3] If one wishes to use an achievement goal ϕ' for ψ, one can use $\Diamond(\phi', now, then)$, i.e. eventually ϕ'.

So we now propose a formalization of conditional intentions that avoids the under/over-commitment problem:

$$\begin{aligned} &\mathrm{CondInt}(agt, \phi, \psi, s) \doteq \\ &\quad \mathrm{Int}(agt, DisjGoal(\phi, \psi, now, then) \\ &\qquad \wedge NoUnderComm(agt, \phi, now, then), s), \\ &DisjGoal(\phi, \psi, now, then) \doteq \\ &\quad [\neg\phi \text{ Until } (\phi \wedge \psi)](now, then) \vee \neg\Diamond(\phi, now, then), \\ &NoUnderComm(agt, \phi, now, then) \doteq \\ &\quad \Box([\mathrm{Int}(agt, \Box(\neg\phi, now, then), now) \supset \\ &\qquad \mathrm{Know}(agt, \mathrm{A}_\Box(\neg\phi, now), now)], now, then). \end{aligned}$$

That is, *agt* conditionally intends that ψ provided that ϕ, iff *agt* intends that the following conditions hold:

1. either (*a*) ϕ eventually holds, and ψ holds immediately from the time ϕ comes to hold, or (*b*) ϕ never holds, and
2. if in any situation *agt* intends that ϕ never comes to hold, she must also know in that situation that it can never become true.

Intuitively, this says that one way to fulfill an agent's conditional intention is to (1*a*) satisfy ψ after ϕ comes to hold, and a second way is that (1*b*) ϕ never comes to hold in the future. This part of our account is as in the disjunctive approach. However, we add to this that (2) the agent does not intend that ϕ never comes to hold unless she knows that it can never hold. Thus we require that if at some situation, *agt* intends that ϕ never comes true, it must be the case that she knows in that situation that ϕ can never become true, and she only intends this because it has become inevitable. So the additional constraint that $NoUnderComm(agt, \phi, now, then)$ ensures that *agt* will not do anything intentionally to make the triggering condition ϕ remain false. One might be tempted to define $NoUnderComm(agt, \phi, now, then)$ as $\Box(\neg\mathrm{Int}(agt, \Box(\neg\phi, now, then), now), now, then)$, i.e. *agt* never intends that ϕ never holds. However, since some event may make ϕ impossible to achieve, there is a possibility that *agt* may come to intend that ϕ always be false, if this becomes inevitable. The only case in which *agt* intends that ϕ always be false is when she knows that it can never become true.

Consider once again our online marketplace example given in Section 1 for the disjunctive account. Using this definition of conditional commitment, a seller *slr*'s intention to send the goods when a buyer *byr* pays, $\mathrm{CondInt}(slr, GetPaid(byr, slr), \mathrm{Happens}(shipGoods(slr, byr), now, then), s)$ can be formalized as follows:

$$\begin{aligned} &\mathrm{Int}(slr, DisjGoal(GetPaid(byr, slr), \\ &\qquad \mathrm{Happens}(shipGoods(slr, byr), now, then), \\ &\quad now, then) \\ &\wedge NoUnderComm(slr, GetPaid(byr, slr), now, then), s). \end{aligned}$$

slr's intention can be further expanded to:

$$\begin{aligned} &\mathrm{Int}(slr, [GetPaidAndThenSendGoods(byr, slr, now, then) \\ &\qquad \vee \neg\Diamond(GetPaid(byr, slr), now, then)] \wedge \\ &\quad [\Box((\mathrm{Int}(slr, \Box\neg GetPaid(byr, slr), now) \supset \\ &\qquad \mathrm{Know}(slr, \mathrm{A}_\Box(\neg GetPaid(byr, slr), now), now)), \\ &\quad now, then)], s), \end{aligned}$$

[3] We could also handle trigger conditions that are not state formulas. However, in these cases, since the trigger condition holds over a time interval, it is not always clear when exactly the triggering of the commitment to the conditional goal should occur. To avoid these complications, we stick to state formulas as triggers.

where,

$$GetPaidAndThenSendGoods(byr, slr, now, then) \doteq$$
$$[\neg GetPaid(byr, slr) \text{ Until}$$
$$(GetPaid(byr, slr) \wedge$$
$$\text{Happens}(shipGoods(slr, byr), now, then))]$$
$$(now, then).$$

From this, we can see that there are only two ways by which *slr* can satisfy this conditional intention: either at some future or current situation *byr* pays *slr* and then *slr* sends the goods to *byr*, or, *byr* never pays *slr*, and as long as *slr* does not know that *byr* will never pay her, she does not intend not to get paid. Since *slr* cannot intend not to get paid, she cannot deliberately perform anything (for example block payments from *byr*) to make the triggering condition remain false. If at a later situation, *slr* learns that it has become impossible for *byr* to ever pay her, *slr* will inevitably intend that *byr* never pays her, but otherwise she cannot acquire this intention. Thus, our formalization of conditional commitment does not suffer from the under-commitment problem.

Moreover, since we use the disjunctive approach, our account does not suffer from the over-commitment problem associated with the conjunctive approach. Consider the second example given in Section 1, where *slr* has the intention to ship a replacement unit when *byr* returns a defective good. Using our definition, this can be expanded to *slr*'s intention that either *byr* never returns a defective product, or *byr* returns a defective product and *slr* ships the replacement unit after that. Thus *slr* is not over-committed and will not perform something deliberately so that *byr* returns a product. The additional constraint that *slr* never intends that *byr* never return a product unless she knows that *byr* will never return a product does not lead to any over-commitment. Thus our formalization of conditional intention is also free from the over-commitment problem.

Note that our account allows the agent who has a conditional intention to intend not to know whether the condition holds. We could easily strengthen the definition to rule this out, but it is not clear that this is always appropriate.

Next, we show two simple properties of conditional intention. Assume that the domain theory D (as discussed in Section 2) includes our definition of conditional commitment given above. Then we have the following theorem that says that if an agent *agt* conditionally intends that ψ provided that ϕ in situation s, and if she knows that ϕ holds in s, then *agt* intends that ψ in s.

THEOREM 1.

$$D \models \text{CondInt}(agt, \phi, \psi, s) \wedge \text{Know}(agt, \phi, s) \supset$$
$$\text{Int}(agt, \psi, s).$$

So when the agent knows that the condition has become true, a conditional intention reduces to an ordinary intention. The second property states that agents are able to introspect their conditional intentions:

THEOREM 2.

$$D \models [\text{CondInt}(agt, \phi, \psi, s) \supset$$
$$\text{Know}(agt, \text{CondInt}(agt, \phi, \psi, now), s)] \wedge$$
$$[\neg\text{CondInt}(agt, \phi, \psi, s) \supset$$
$$\text{Know}(agt, \neg\text{CondInt}(agt, \phi, \psi, now), s)].$$

Thus, if an agent has a conditional intention (does not have a conditional intention, resp.) that ψ provided that ϕ, then she knows that she has (does not have, resp.) this conditional intention.

It would be interesting to prove additional results about conditional intentions, for instance, that a conditional intention persists as long as its condition is known to remain false and not known to have become impossible. We leave this for future work.

4. CONDITIONAL REQUESTS

We now discuss two communicative acts, *requestWhen* and *reqActWhen*, that can be used by an agent to request someone to achieve ψ or to execute a program δ respectively, on the condition that ϕ becomes true. Recall that, in ECASL the SSA for W determines whether an agent adopts a goal when requested; the requested goal is adopted by the requestee via cooperation principles encoded in the SSA for W. Thus, we model requests as informing about intentions, rather than as primitives. In the following, we use $CondIntCont(agt, \phi, \psi)$ as an abbreviation for the content of a conditional intention $DisjGoal(\phi, \psi, now, then) \wedge NoUnderComm(agt, \phi, now, then)$. Now, one simple way to model a requester *req*'s request to requestee *agt* to achieve ψ on the condition that ϕ is as follows:

$$requestWhen_{sim}(req, agt, \phi, \psi) \doteq$$
$$request(req, agt, CondIntCont(req, \phi, \psi)).$$

This says that, *req*'s conditional request to *agt* to achieve ψ provided that ϕ amounts to *req*'s request to *agt* to fulfill the content $CondIntCont(req, \phi, \psi)$ of her own conditional intention. Using the definition of *request*, this conditional request amounts to *req* informing *agt* that she currently intends to achieve ψ provided that ϕ. However, note that the content $CondIntCont(req, \phi, \psi)$ of this conditional intention includes mental attitudes that refer to *req*, rather than *agt*. Since the SSA for W does not automatically replace the agent parameters of mental state operators used in a goal formula, if we model conditional requests as above, given appropriate conditions (i.e., when *agt* agrees to serve *req* on $CondIntCont(req, \phi, \psi)$ and does not currently have the intention that $\neg CondIntCont(req, \phi, \psi)$), *agt* will adopt the intention that $CondIntCont(req, \phi, \psi)$, but not that $CondIntCont(agt, \phi, \psi)$. Thus she will not have the conditional intention to achieve ψ provided that ϕ after the conditional request is performed, and this simple definition is not quite correct.

For example, suppose that the manager agent *mgr* wants to conditionally request the seller *slr* in situation s to ship the goods when the buyer *byr* pays her. So *mgr* can do this by performing the following action in s:

$$requestWhen_{sim}(mgr, slr, GetPaid(byr, slr),$$
$$\text{Happens}(shipGoods(slr, byr), now, then)),$$

which can be expanded to:

$$request(mgr, slr,$$
$$CondIntCont(mgr, GetPaid(byr, slr),$$
$$\text{Happens}(shipGoods(slr, byr), now, then))).$$

After the request is performed, if *slr* agrees to serve *mgr* on $CondIntCont(mgr, GetPaid(byr, slr)$, Happens (*ship-*

$Goods(slr, byr), now, then))$, and does not intend that $\neg CondIntCont(mgr, GetPaid(byr, slr), \text{Happens}(shipGoods(slr, byr), now, then))$, the SSA for W will ensure that:

$$\text{Int}(slr, CondIntCont(mgr, GetPaid(byr, slr), \text{Happens}(shipGoods(slr, byr), now, then)), s_r),$$

which can be expanded to:

$$\text{Int}(slr, DisjGoal(GetPaid(byr, slr), \text{Happens}(shipGoods(slr, byr), now, then), now, then) \wedge NoUnderComm(mgr, GetPaid(byr, slr), now, then), s_r),$$

where s_r is the situation that results from performing the $requestWhen$ action in s. Now, using the definition of conditional intention, we can see that in s_r, slr does not have the conditional intention of sending the goods provided that byr pays her. The problem is with the mental state operators in the $NoUnderComm(mgr, \ldots)$ part of slr's intention: they say that mgr will not intend that the payment not occur unless she knows it can never occur. What we need is for this constraint to hold for slr.

To deal with this problem, we propose the following model of conditional requests:

$$requestWhen(req, agt, \phi, \psi) \doteq request(req, agt, CondIntCont(agt, \phi, \psi)).$$

This says that req's request to agt to conditionally achieve ψ provided that ϕ amounts to req's request to agt to fulfill the content of agt's conditional intention to achieve ψ provided that ϕ, i.e., $CondIntCont(agt, \phi, \psi)$. Using the definition of *request*, this can be further expanded to:

$$requestWhen(req, agt, \phi, \psi) \doteq inform(req, agt, \text{Int}(req, CondIntCont(agt, \phi, \psi), now)).$$

That is, req can request agt to achieve ψ on the condition that ϕ by informing agt that she intends that $CondIntCont(agt, \phi, \psi)$. Note that, the agent parameter of $CondIntCont(agt, \phi, \psi)$ is now the requestee agt, rather than the requester req. This guarantees that given that agt serves req and does not have the opposite intention, she will conditionally intend to achieve ψ provided that ϕ after req conditionally requests her this. Thus this formalization of conditional request does not suffer from the above mentioned problem.

We also define a special type of conditional request, namely, a request to perform an action when some condition holds:

$$reqActWhen(req, agt, \phi, \delta) \doteq requestWhen(req, agt, \phi, Happens_C(\delta, now, then)).$$

This states that req's conditional request to agt to execute the program δ provided that ϕ amounts to req's conditional request to agt to execute δ starting in the situation where ϕ holds.

Now consider what happens when mgr conditionally requests slr to ship the goods when byr pays her, that is, when mgr performs the $reqActWhen(mgr, slr, GetPaid(byr, slr), shipGoods(slr, byr))$ action. Given that slr agrees to serve mgr and does not have the opposite intention, the SSA for W will make slr adopt the following intention:

$$\text{Int}(slr, CondIntCont(slr, GetPaid(byr, slr), \text{Happen}_C(sendGoods(slr, byr), when, then)), s_r),$$

and thus, by the definition of conditional intention, she will conditionally intend to send the goods when byr pays her. Thus, our formalization of conditional requests allows the proper transfer of conditional intention from the requester to the requestee.

We next present a theorem that shows how agents' intentions are affected by the *requestWhen* action. Assume that the domain theory D includes our definition of these new communicative actions. We can show that:

THEOREM 3.

$$D \models [\neg\text{Int}(agt, \neg CondIntCont(agt, \phi, \psi), s) \wedge Serves(agt, req, CondIntCont(agt, \phi, \psi), s) \wedge Poss(requestWhen(req, agt, \phi, \psi), s)] \supset \text{CondInt}(agt, \phi, \psi, do(requestWhen(req, agt, \phi, \psi), s)).$$

This says that if in some situation s, an agent agt does not intend not to fulfill the content of a conditional intention to achieve ψ provided that ϕ, and if she serves another agent req on the content of this conditional commitment in s, then she will have the conditional intention to achieve ψ given that ϕ after req conditionally request her this in s, provided that the request is possible in s.

It would be useful to extend our framework with a communication act that allows a conditional commitment created as a result of a *requestWhen* to be cancelled. We believe that the existing ECASL *cancelRequest* action can be used to define such a conditional commitment cancelling act. We leave this for future work.

5. RELATED WORK

The under-commitment problem that we pointed out in Section 1 is related to another problem involving intentions discussed by Cohen and Levesque [2]. In that paper, they consider a robot who drops the intention of bringing a bottle of beer by breaking the last available bottle and thus making the intention impossible to achieve. Their solution was twofold: (1) they formalize intentions as persistent goals and (2) they assume that existing intentions act as a screen of admissibility over new intentions. In their framework, an agent's intentions persist until she knows that they have been achieved, or knows that it has become impossible to achieve them. Since the robot intends to bring a bottle of beer, she will not drop this goal until she achieves it or gets to know that it has become impossible to achieve. However, the robot can break the last available bottle to make her goal unachievable. But since an agent's current intentions provide a screen of admissibility for adopting new intentions, she cannot have these two conflicting intentions at the same time. Thus since she intends to bring a bottle, she cannot adopt the intention to break the only available bottle. Note that while this problem has similarities with the one addressed here, it does not involve conditional intentions.

In the literature, there has been some work on conditional intention. However, as mentioned earlier, all of the proposed treatments that we are aware of seem to suffer from the under- or over-commitment problems. Although it does not explictly address conditional intentions, the FIPA agent communication language specification [5] defines a type of communication act that leads to conditional intentions. In that framework, an agent can conditionally request another agent to execute an action when some condition holds. This

is modeled as follows: req's conditional request to agt to perform act when ϕ holds amounts to req informing agt that she has the intention that agt execute act and that ϕ be true just before that. Note that req's intention amounts to the conjunction that ϕ be true at some point and agt executes act right after that. Thus, this treatment of conditional intention can be viewed as a conjunctive account where the intention is to first achieve the triggering condition ϕ, and then to achieve the conditional goal. As discussed in Section 1, this leads to the over-commitment problem.

Yolum and Singh [28] present a different model of conditional commitment that relies on a social obligation-based semantics rather than a traditional one based on mental states. Their main concern was the study of communication protocols that accommodate exceptions and take advantage of opportunities. They model interaction protocols using *commitment machines* that supply a content to protocol states and actions in terms of the social commitments of the participating agents. In their formal semantics, which is only briefly described, they adopt a branching time temporal model. The semantics for commitments involves a modal accessibility relation for commitments C that relates a state of the protocol (i.e. a time-point) s, a debtor agent x, and a creditor agent y to a set of paths P. Intuitively, x is responsible to y for satisfying ϕ in state s iff ϕ holds at time-point s along all paths p that are C-accessible from (x, y, s). To model conditional commitment, they introduce a strict implication operator (denoted by $\rightsquigarrow$) that requires the consequent to hold when the antecedent holds. The strict implication is false when the antecedent is false. Their semantics says that $\phi \rightsquigarrow \phi'$ holds in a state s iff ϕ holds in s and for all s' that satisfy ϕ, every s'' that is similar to s' (i.e. $s'' \approx s'$) also satisfies ϕ'. What they mean by the similarity relation $\approx$ is not explained. Thus for them, a conditional commitment $C(x, y, \phi \rightsquigarrow \phi')$ holds in state s iff on all C-accessible paths p, ϕ holds at s, and whenever some s' satisfies ϕ, every s'' that is similar to s' satisfies ϕ'. Since they model conditional commitments using the $\rightsquigarrow$ operator, which behaves like a conjunction with some additional constraints, it appears that their formalization suffers from the over-commitment problem. It is also not clear how their formalization ensures that the goal is achieved *after* the condition along the paths.

Both [1] and [23] model conditional commitment as a disjunctive goal. In their social commitment and argument network based framework, Bentahar et al. [1] define conditional commitments as a simple implication. Their semantics of conditional commitment goes as follows: $M, s \models \text{CondInt}_{Ben}(agt_1, agt_2, \phi, \psi)$ iff $M, s \models \text{EF}^+\phi \Rightarrow M, s \models \text{ABC}(agt_1, agt_2, \psi)$, where s, E, F^+, and ABC denotes a time-point, there exists a path, sometime in the future, and absolute commitment, respectively. This says that agt_1 is committed to agt_2 to achieve ψ on the condition that ϕ means that agt_1 is unconditionally committed to agt_2 to achieve ψ if ϕ holds at some timepoint over some path in the future. Besides suffering from the under-commitment problem associated with disjunctive accounts, this seems to require commitment to the goal too early, before the condition becomes true.

In [23], Shapiro et al. describe a framework for specifying communicative multiagent systems using ConGolog [4] within the situation calculus, an early version of CASL. Since they were lacking a goal-revision mechanism at that point, they introduced a type of conditional request, the *requestUnless* action, in an attempt to avoid the need for goal-revision. $requestUnless(req, agt, \phi, \psi)$ means that req is requesting agt to adopt the goal that ψ unless ϕ is obtained. The execution of $requestUnless(req, agt, \phi, \psi)$ makes agt adopt the goal that $\phi \vee \psi$. This amounts to modeling conditional intentions as disjunctive goals, and hence the account suffers from the under-commitment problem.

6. CONCLUSION

In this paper, we identified some problems with many existing formalizations of conditional commitments. These seem to either have the agents over-committed, intending to achieve the condition under which the goal would have to be achieved, or under-committed, possibly intending that this condition remain false forever. We could not find any problem-free account in the literature. We presented a definition of conditional intentions that does not suffer from these problems. We then formalized two types of communicative actions that allow agents to make requests that lead to conditional commitments. We also proved some properties of conditional commitments and conditional requests. Finally, we discussed previous work on conditional commitments.

Note that, our framework allows an agent with a conditional intention to not intend that the trigger condition eventually becomes true. However, it does not allow her to intend that the trigger condition never comes to hold, without also knowing that it can never become true. In other words, in our framework, an agent's conditional intention that ψ provided that ϕ is not consistent with her intention that $\Box\neg\phi$, unless she already knows that this must be the case. This might be problematic in some cases. For instance, in our example where a seller has a conditional intention to ship a replacement unit when a buyer returns a defective product, we might want to say that the seller has the intention that the buyer never returns a defective good. However, it is not possible for an agent to consistently have both of these intentions in our framework. One way to overcome this limitation might be to adopt a richer semantic model of intention, where one allows different degrees of preferability, similar to the levels of plausibility in traditional belief revision frameworks such as [6]. Such semantic models and the resulting logics are more expressive, but much more complex to specify and reason in.

The theory presented here is a part of our ongoing research on the semantics of speech acts and agent communication in the situation calculus. In [8], we present an extended version of this work where we model some simple communication protocols that deal with conditional requests. Much work remains. In the future, we would like to prove other properties of conditional commitments, for example, about the persistence and revision of such commitments. We also plan to formalize complex interaction protocols, such as the Contract Net protocol [27] and the Net Bill protocol [26], using our formalization of conditional intention. It would also be interesting to try to use this formalization to implement flexible communication agents as in [19] and to develop tools to support multiagent programming.

7. ACKNOWLEDGEMENTS

We thank Hector Levesque, Pinar Yolum, and the review-

ers for useful comments on this work.

8. REFERENCES

[1] J. Bentahar, B. Moulin, J.-J. C. Meyer, and B. Chaib-draa. A logical model of commitment and argument network for agent communication. In *Proc. of AAMAS-04*, pages 792–799, 2004.

[2] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–361, 1990.

[3] P. Cohen and H. Levesque. Rational interaction as the basis for communication. In P. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*, pages 221–255. MIT Press, Cambridge, Mass., 1990.

[4] G. De Giacomo, Y. Lespérance, and H. Levesque. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence*, 121:109–169, 2000.

[5] Foundations for Intelligent Physical Agents. FIPA communicative act library specification, document 37. 1997-2002.

[6] P. Gardenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, Massachusetts, 1988.

[7] A. Herzig and D. Longin. A logic of intention with cooperation principles and with assertive speech acts as communication primitives. In *Proc. of AAMAS-02*, pages 920–927, 2002.

[8] S. Khan. A situation calculus account of multiagent planning, speech acts, and communication, MSc Thesis (in preparation), 2005.

[9] S. Khan and Y. Lespérance. ECASL: A model of rational agency for communicating agents. In *Proc. of AAMAS-05*, pages 762–769. Utrecht, The Netherlands, July 2005.

[10] G. Lakemeyer and H. Levesque. AOL: A logic of acting, sensing, knowing, and only-knowing. In *Proc. of KR-98*, pages 316–327, 1998.

[11] H. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. Golog: A logic programming language for dynamic domains. *J. of Logic Programming*, 31:59–84, 1997.

[12] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[13] R. Moore. A formal theory of knowledge and action. *Formal Theories of the Commonsense World*, pages 319–358, 1985.

[14] G. Plotkin. A structural approach to operational semantics. Technical Report DAIMI-FN-19, Computer Science Dept., Aarhus University, Denmark, 1981.

[15] A. Rao and M. Georgeff. Modeling rational agents within a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proc. of KR&R-91*, pages 473–484, 1991.

[16] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in the Honor of John McCarthy*. Academic Press, 1991.

[17] R. Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[18] D. Sadek. Communication theory = rationality principles + communicative act models. In *Proc. of AAAI-94 Workshop on Planning for Interagent Comm.*, 1994.

[19] D. Sadek and P. Bretier. ARTIMIS: Natural dialogue meets rational agency. In *Proc. of IJCAI-97*, pages 1030–1035, 1997.

[20] R. Scherl and H. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2), 2003.

[21] S. Shapiro. *Specifying and Verifying Multiagent Systems Using CASL*. PhD thesis, Dept. of C.S., U. of Toronto, 2005.

[22] S. Shapiro and Y. Lespérance. Modeling multiagent systems with the Cognitive Agents Specification Language - a feature interaction resolution application. In C. Castelfranchi and Y. Lespérance, editors, *Intelligent Agents Vol. VII - Proc. of ATAL-00*, volume LNAI 1986, pages 244–259, 2001.

[23] S. Shapiro, Y. Lespérance, and H. Levesque. Specifying communicative multi-agent systems. *Agents and Multi-Agent Systems – Formalisms, Methodologies, and Applications*, LNAI 1441:1–14, 1998.

[24] S. Shapiro, Y. Lespérance, and H. Levesque. The Cognitive Agents Specification Language and verification environment for multiagent systems. In *Proc. of AAMAS-02*, pages 19–26, 2002.

[25] M. Singh. *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*. LNAI 799, 1994.

[26] M. Sirbu. Credits and debits on the internet. *Readings in Agents*, pages 299–305, 1998.

[27] R. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[28] P. Yolum and M. Singh. Commitment machines. In J.-J. C. Meyer and M. Tambe, editors, *Intelligent Agents VIII : 8th Intl. Workshop, ATAL-01*, volume LNAI 2333, pages 235–247, 2002.