A Logic-Based Model of Intentions for Multi-Agent Subcontracting*

John Grant
Dept. of Computer and Information Sciences
and Department of Mathematics
Towson Univ.
Towson, MD 21252 USA
jgrant@towson.edu

Sarit Kraus
Dept. of Computer Science
Bar-Ilan Univ., Ramat-Gan, 52900 Israel
and Institute for Advanced Computer Studies
Univ. of Maryland, College Park, MD 20742
sarit@umiacs.umd.edu

Donald Perlis
Dept. of Computer Science
Univ. of Maryland,
College Park 20742, USA
perlis@cs.umd.edu

Abstract

We present a formalism for representing the intentions of agents engaged in cooperative planning and acting. We focus on cases where one agent alone cannot accomplish a complex task and must subcontract with other agents. Evolving intentions over time during the planning and acting, and the conditions under which an agent can adopt and maintain an intention, are central. In particular, the time taken to plan and to subcontract are modeled explicitly in the logic. This explicit time-representation is used to account for the time it takes an agent to adopt an intention. We use a syntactic approach presenting a formal logical calculus that can be regarded as a meta-logic that describes the reasoning and activities of the agents. We write some of the axioms of this meta-language and explain the minimal model semantics, in which one model, the intended model, represents the actual beliefs, intentions, and actions of the agents. We also prove several results showing that under the appropriate conditions the agents will act as expected.

Introduction

In this paper we argue that to properly understand computational models of intentions, especially in the context of multi-agent cooperative behavior, it is very useful to have a formal meta-theory of that behavior; and we present such a theory as well.

Logic plays at least two roles here. First, since much of what an agent must do is use its existing information to help it decide what to do, then the agent itself is employing processes for drawing conclusions, and is then using its own internal logic. Second, our own analysis of an agent's behavior can be made precise by a meta-logic in which we describe – and prove theorems about – the agent. For instance, an agent may know that another agent can perform action a, and that if a is performed then B will be true. The first agent may then conclude that, in order to make B true, it should ask the second agent to do a. This conclusion is in effect a use of modus ponens on the part of the first agent. If we have designed the agent so that it will always use Modus Ponens in such situations, then we can formulate that behavior as a meta-axiom and use it to prove general results about the agent. Whether B will become true in the above depends, in part, on whether the second agent cooperates with the first. In this paper we assume all agents to be cooperative; giving this requirement a formal characterization involves the notion of intention. We require an agent to adopt a potential intention to perform an action if it is asked to do so. This then becomes a (real) intention if other conditions, to be described below, are met as well.

Our work differs from others in its use of a meta-theory of intentions and of an evolving notion of time that allows agents to reason about genuinely-approaching deadlines. Thus we view the reasoning of agents as ongoing processes rather than as fixed sets of conclusions. This reasoning involves rapidly evolving sets of beliefs and intentions, and is governed in part by formal rules of inference. We model beliefs and intentions in a formal logical calculus that can be regarded as a meta-logic describing an actual on-board agent logic that evolves in (and takes account of) that same passage of time. This is a style of formal representation that departs from traditional temporal logics in that there is an internally evolving notion of time that the logic must keep track of.

Our approach utilizes a strongly sorted calculus, distinguishing the application language, time, and various syntactic sorts. This allows for useful and intuitively natural characterizations of such agents' reasoning abilities. We provide formal tools for representing agent intentions and methods for reasoning with and about such intentions, especially in the context of cooperative actions. We focus on cases where one agent alone cannot accomplish a complex task and must subcontract with other agents.

One of the scenarios that we have considered for an example is that of a software agent personal assistant. Suppose that you need plans to get to the airport to catch your flight later today but are busy with other things. So you ask this assistant, named PAT, to make arrangements to get you to the airport on time. PAT may not be fully expert in web searches, for instance, but it knows what to do: contact whatever expert agents are needed to assist it in the details, so that you can catch your flight. PAT breaks down this assignment into several parts: determine likely means of transport, how to arrange them, etc. Among PAT's concerns are temporal ones: it needs to find out information and arrange for the transportation within a reasonable period of time. This

*This research was supported by NSF under grant IIS-9907482.

means not only knowing how soon you need to be at the airport, but also how quickly the various subtasks can be done, including ones done by auxiliary agents such as specialized web search agents and telephone agents. In the next sections we will use this example for illustration purposes.

Languages for Reasoning By and About Agents

In our framework we find it useful to introduce three related languages. The first language, L_S, is the language of the subject matter, that is, the subject that the agents reason about: e.g. web searches. The second language, L_A, is the "agent" language. This includes the subject matter language, but also has additional symbols allowing assertions about what some particular agent may believe at a given moment; this allows an agent to reason about another agent, for instance in deciding whether to ask another agent to perform some subtask based on what that other agent believes. (We assume that there is a public database with – possibly fallible – information about agent abilities.) The third language, L_M, is the meta-language used by us – as observers – to reason about the agents.

The meta-language, L_M, must be powerful enough to characterize agent behavior quite generally. For instance, in our meta-language we can write the statement that an agent always applies Modus Ponens, when that is possible. This, taken as an axiom, would assert that the agents are programmed to apply Modus Ponens; that is an aspect of their behavior. Thus, while the agent language, L_A, must allow for the expression of the formulas *A*, *A implies B*, and *B*, still an agent does not in general have the meta-rule such as "I apply Modus Ponens whenever I can," even if that happens to be true (and expressed in the meta-language).

In other approaches to the issue of agent reasoning, often logical omniscience is assumed. We think that our approach, where rules are applied over time is a more realistic way to model agent beliefs.

Agent Beliefs and Intentions

Because of our emphasis on changing knowledge over time, agents need to know the present time. Also, in our framework agents have introspection capabilities. By this we mean that one of the agent's beliefs may be that it believed or did not believe something at a different time. There are two types of introspection: positive and negative. An example of positive introspection is if an agent believed some fact (represented by a formula) at time t, it will then believe at time $t+1$ that it believed that fact at time t. This way an agent can reason about its own or another agent's beliefs over time. Suppose now that at time t the agent is considering but does not believe a particular possible fact A (that is, A is not in its database). Then at the next time value, $t+1$, it will believe (know) by negative introspection that it did not believe A at time t.

Another important capability of an agent is the inheritance of beliefs. The idea is that if an agent believes some fact A and does not believe something contradictory to it (that is, $notA$), it will continue to believe A. There are exceptions to inheritance. Things that are changing, such as the location of a moving object, should not be inherited. A fact involving the *Now* predicate is not inherited, because if the agent believes that the time now is t, it should not continue this belief at time $t+1$.

We have found it useful also to add axioms concerning the cooperation of agents. So for example if an agent "tells" another agent a fact, the second agent will add that fact to its database of beliefs. This assumes that agents are trustworthy. Agents may also be helpful to other agents by answering each other's questions. Note that of course there can be nontrustworthy agents, and that this is a major topic of research (see a survey in (Kraus 2001)). But here we focus on fully cooperative (and nonforgetful) agents.

The exact definition of intentions may be necessary for cooperation. For example, an agent needs to know how to interpret a belief that its partner has adopted an intention to do the given action: does this mean the action will indeed be performed by that partner if the partner is able? Thus, it is important to formally define (a) what are the minimal requirements that are needed for an agent to adopt an intention; (b) how an agent's intentions change over time (and what are the occasions when an agent will drop an intention); (c) what are the reasoning processes that the agent will perform when it adopts an intention; and (d) what are the actions that an agent will perform when it adopts an intention.

We define a notion of "potential" intention that means roughly that the agent has been given a task and is determining whether it can do it (perhaps with help). We reserve "intention" for the situation in which the agent has determined that it can do the task (perhaps with help) and in that case it will indeed attempt to carry out the task, unless in the meantime it has dropped that intention because (a) it is told to drop it; or (b) it is given another task that conflicts with the first one and which is assigned higher preference; or (c) the world changes and it finds that it can no longer do the task (perhaps because a subcontracting agent cannot do its part).

In the context of collaboration, it is also important to decide whether an agent can have intentions that another agent will perform an action. In our model, an agent can't intend directly that another agent will do an action; however, its plan for doing a, that is motivated by the intention to do a, may include the intentions of other agents.

We use the term "*recipe* for action a" (Pollack 1990) to refer to a specification of a set of actions, the doing of which constitutes performance of a. The subsidiary actions $b1, b2, ..$ in the recipe for action a, which are also referred to as subacts or subactions of a, may either be *basic actions* or complex actions. Basic actions are done by primitive acts by agents that can do them. Thus, the general situation is illustrated in the tree of Figure 1. Here an agent is given task $a1$, which has a recipe shown in the tree, i.e. to do $a1$, it is sufficient to do $b1$,$b2$,$b3$ (in that order) and $b1$ can be done by doing $c1$ and $c2$, etc. The leaves are the "basic" actions If an agent cannot do all the actions for a recipe, it then may subcontract some of them to other agents.

Sorted Language for Agent Intention

In this section we introduce the main predicates that we use in our work in the meta-language to characterize agent inten-

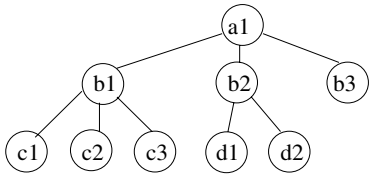

Figure 1: An example of a recipe tree.

tions. For each predicate we explain the use of the different attributes. We use a sorted language for comprehensibility and convenience so that, for example, agent names and times (for both of which we use positive integers) cannot be confused. Since the language is sorted, we use a convention for variables of different sort, namely t,i,j,k for time; m,n for agent names, a,b,c for actions, and r for recipes. As needed, e is used for the null element.

We start with the two predicates for intention: $PotInt$ and Int, as well as AskedToDo (ATD). Basically, $PotInt$ represents a potential intention for an agent asked to perform an action. Under certain conditions $PotInt$ will become Int.

The context keeps track of the tree structure of the recipe used so that when an agent has a potential intention or an intention for an action, the context of the action, if there is one, is the parent node in the tree. For instance, in the example given in the Introduction, the context for $c2$ would be $b1$ assuming that the potential intention or intention is for the same agent. For the root node, $a1$, the context is the null action e. Also, if $c2$ is the potential intention or intention of an agent requested to do $c2$ by another agent, the context is also the null action, but the requesting agent is also indicated.

$ATD(t, n, m, a, t')$ - At time t agent n asks agent m to do action a at time t'.

This predicate is used both by the agent's owner who asks an agent to do a task, as well as by other agents as they request one another to do various tasks for them: this is subcontracting. There are two times involved, the time of the asking and the time that the action needs to be done. This will also be the case for several other predicates. In the example given above, writing aat for "arrange airport transportation", we could write $ATD(14:30:00, e, PAT, aat, 17:00:00)$ to indicate that the owner (null agent) asked PAT (in the axioms we will assume for convenience that each agent is referred to by a number) at 2:30PM to arrange for airport transportation at 5PM.

$PotInt(t, m, n, b, a, t')$ -At time t agent m directly assisting agent n has the potential intention to do action b in the context of action a at time t'.

When PAT is asked to arrange the airport transportation, it adopts a potential intention to do this task at the next time period. Assuming that a time period is one second, we will have $PotInt(14:30:01, PAT, e, aat, e, 17:00:00)$ indicating that PAT will have a potential intention to arrange for airport transportation at 5PM. The task is done for the owner (hence the first e) and the arrange action is not a subaction in the context of another action (hence the second e).

$Int(t, m, n, b, a, t')$ - At time t agent m directly assisting agent n has the intention to do action b in the context of action a at time t'.

In our example, if PAT or its assistants will adopt all the needed intentions to arrange for airport transportation, say at time 14:35:00, then in the next time period it will have $Int(14:35:01, PAT, e, aat, e, 17:00:00)$.

We distinguish between basic level actions that can be performed without planning and complex actions that consist of several actions. These actions are specified in a recipe. The actions in a recipe may themselves be complex. $BL(a)$ indicates that action a is basic level. $Rec(a, r)$ means that for action a, r is the chosen recipe (not dealing with multiple recipes at this time). $Mem(r, b, i, j, k)$ means that in recipe r, b is the i'th subaction starting at relative time j and ending at relative time k (i.e. relative to the time at the beginning of the action). These predicates do not involve time in the sense of the previous predicates, because they refer to facts that are considered true for every time period.

In the next group we deal with other relevant concepts about agents: their beliefs, abilities to do actions, and means of communication with other agents. $Bel(t, n, f)$ means that at time t agent n believes the statement expressed by formula f. $CanDo(t, n, a)$ indicates that at time t agent n can do (or at least start) action a. $Tell(t, n, m, f)$ means that at time t agent n tells agent m the statement expressed by the formula f. The formulas are the formulas of the agent language, L_A. These are the formulas that are meaningful to the agents. Each such formula has a name (its quoted version) in the meta-language L_M.

Finally we write the predicates that deal with the agents actually doing the actions, the initialization and completions of an action. $Ini(t, m, n, a)$ means that at time t agent m assisting agent n initiates action a. $Done(t, a)$ means that at time t action a has just been done.

Axioms of the Metalanguage

We present a theory $\mathcal{T}$ over L_M which consists of the axiom schemas that describe the desired intentions and behaviour of a team of cooperative agents and how their intentions change over time. We do not present any of the basic axioms of beliefs, time etc. that are given in (Grant, Kraus, & Perlis 2000). We also do not present the entire set of 17 axioms for intentions, but rather only about half of them to illustrate important concepts. First we sketch the general scenario for the agent potential intentions, intentions, subcontracting and doing actions.

When an agent is asked to do an action a at a particular time, in the next time period it adopts a potential-intention to do it. If the agent believes that it can do the action a, then in the basic level case, in the next time period it will adopt an intention to do it. In the case of a complex action, it will look for a recipe and in the next time period will adopt potential intentions for all the first-level subactions of the recipe. If after adopting a potential intention the agent comes to believe that it can't do the action a, in the next time period it will ask another agent, that it believes can do a, to do a

If at a particular time period an agent has a potential intention to do a complex action, it will adopt in the next time period an intention to do the action if for each subaction in the

recipe it either adopted an intention to do it or it has found another agent with an intention to do it. This process of checking the needed intentions and beliefs and the adopting of an intention will be repeated while the agent still has the potential intention. Each request to perform an action has a time associated with it. If all the needed intentions have been adopted for an action, then at the associated time the agent will initiate it. The initiations will lead to the performance of all the basic actions in the recipe tree of the action at the appropriate times, assuming that the agents can actually do the actions.

We start with a group of axioms that involves the adoption of intentions by the agents. Then we will discuss the axioms involving the agents doing the actual actions. We will conclude by discussing axioms of inheritance. All the variables that are not written with existential quantification are assumed to be universal quantified. All the axioms have the same form of $\alpha \rightarrow \beta$ and typically the time of α is t and the time of β is $t+1$. This is because the axioms characterize the way the mental state of the agent and the state of the world change over time. This way we capture the property that agent reasoning, planning, and actions take time.

We present here three of the axioms of the adoption of intentions group. The first axiom deals with subcontracting an action to one of the agents that can do it. This happens when an agent has a potential intention to do an action, but does not believe that it can do it. Unless it will ask another agent, this potential intention will not become an intention. The antecedents of this axiom are that the agent has a potential intention to do an action and does not believe that it can do it and believes that agent m is the "first" agent that can do it and it has not previously asked m to do it. The consequence is that the agent will ask m to do the action.

$$\begin{aligned}&PotInt(t,n,n',b,a,t') \;\&\; \neg Bel(t,n,"CanDo(t',n,b)") \;\&\\ &Bel(t,n,"CanDo(t',m,b)") \;\&\\ &(Bel(t,n,"CanDo(t',m',b)") \;\rightarrow\; m \leq m') \;\&\\ &(t" < t \;\rightarrow\; \neg ATD(t",n,m,b,t')) \;\&\; t+1 < t'\\ &\qquad\qquad \rightarrow\; ATD(t+1,n,m,b,t')\end{aligned}$$

The second axiom states that when an agent gets an intention to do an action for another agent, it instantaneously tells this to the other agent.

$$Int(t,m,n,b,a,t') \;\rightarrow\; Tell(t,m,n,"Int(t,m,n,b,a,t')")$$

The third axiom shows how potential intention becomes intention. If an agent has a potential intention to do an action and believes that it can do the action (possibly with help) and if each subaction in the recipe is either intended by the agent or an assisting agent, then in the next time period the agent will have the intention to do the action. This means that if an agent adopts an intention for a complex action, all the basic level actions in the recipe tree must already have been intended by this or some assisting agents. We take into account the time it takes to go down (adopting potential intentions) and up (adopting intentions) in the recipe tree. Additional time is needed for communication between agents.

$$\begin{aligned}&PotInt(t,n,n',b,a,t') \;\&\; Bel(t,n,"CanDo(t',n,b)") \;\&\\ &Rec(b,r) \;\&\; t+1 < t' \;\&\\ &Mem(c,r,i,j,k)\&(Int(t,n,n',c,b,t'+j) \vee\\ &\qquad\qquad \exists m Tell(t,m,n,"Int(t,m,n,c,e,t'+j)"))\\ &\qquad \rightarrow\; Int(t+1,n,n',b,a,t')\end{aligned}$$

Next we present three axioms involving agents doing actions. The first shows how the root of the recipe tree is initiated. When a basic level action is initiated by an agent, it gets done in one time period if the agent can do it. For a complex action, the agent must start by initiating the root of the recipe tree. Each node will have to be initiated in turn. Several agents may be involved in doing various subactions. We use the initiation of complex actions as a bookkeeping device to make sure that all the subactions get done at the proper time and in the right order.

The first axiom of this group is the initiation of the root action. If an agent has the intention for the root action starting at the next time step, it will initiate it.

$$Int(t,n,e,a,e,t+1) \;\rightarrow\; Ini(t+1,n,e,a)$$

If an action is the first subaction of a complex (sub)action, the agent doing the action will initiate it. A later subaction will be initiated by the agent or an assisting agent after all the previous subactions have been done. The second axiom shows that if the previous subaction was done at the right time and the agent doing the action has an assisting second agent intending to do the subaction, then this second agent will initiate the next subaction.

$$\begin{aligned}&Ini(t,m,n,a) \;\&\; Rec(a,r) \;\&\; Mem(r,b,i+1,j',k') \;\&\\ &Mem(r,c,i,j,k) \;\&\; Done(t+k,c) \;\&\\ &Int(t+j-1,m',m,c,e,t+j)\\ &\qquad\qquad \rightarrow\; Ini(t+j,m',m,c)\end{aligned}$$

The third axiom of this group considers the performance of a basic level action. If an agent initiates a basic level action at time t and can do it, then it will get done at time t+1.

$$\begin{aligned}&BL(a) \;\&\; Ini(t,m,n,a) \;\&\; CanDo(t,m,a)\\ &\qquad \rightarrow\; Done(t+1,a)\end{aligned}$$

A complex action is done when all its subactions are done.

The third group of axioms involves the inheritance of various mental states and activity formulas. We need inheritance because of our explicit representation of time. Each formula is associated with a specific time. Being true at a given time does not necessarily mean that the formula will continue to be true in the next time period. That is accomplished by the inheritance axioms that specify explicitly which formulas and under which conditions will be inherited from one time period to the other. Intentions are not inherited. Before continuing to hold an intention, an agent must check that all the conditions of having an intention are still valid, as specified in one of the axioms above. In addition to inheriting the potential intentions of a complex action, an agent also adopts potential intentions of the action's subactions (given below). Initiation of a given action is not inherited by itself, but rather by the initiation of its subactions in the proper order and the proper time.

$$\begin{aligned}&PotInt(t,m,n,b,a,t') \;\&\; Bel(t,m,"CanDo(t',m,b)") \;\&\\ &Rec(b,r) \;\&\; Mem(r,c,i,j,k) \;\&\; t+1 < t'\\ &\qquad \rightarrow\; PotInt(t+1,m,e,c,b,t'+j)\end{aligned}$$

Minimal Model Semantics

The axioms of the meta-theory can be shown to be consistent by constructing a model for them. In general, there will be many models for such a first-order theory that have only a tenuous relationship to the application of our interest. In fact, we are interested only in one model, the intended model that represents as its true formulas the actual beliefs, intentions, and actions of the agents as these change over time. The first restriction is to consider only Herbrand models, i.e. models that contain only elements that are ground terms of L_M. The second restriction is to consider only minimal models where we minimize each predicate for one time period at a time starting with 0 and then proceeding by induction on time. Because of the structure of the axioms, there will be only one such model. This construction is based on the construction of the unique perfect model of a locally stratified theory in logic programming. See (Grant, Kraus, & Perlis 2000) for details.

In order to have a sharp sense of how our agents behave, we simply define them to be processes that behave as in this unique minimal model. This has the desired effect of providing certain converses of the axioms as true statements in this model. In effect we are making a an assumption of complete information (akin to a closed-world assumption) about agent behavior; for example an agent comes to have a particular potential intention at time t, if and only if our axioms require that agent to have that potential intention at that time. This is a restriction of sorts; after all, much of commonsense reasoning involves situations that are usually taken to be too complex to fully axiomatize. But our view is to suppose that the complexities primarily arise from the external world and that the agent behaviors are completely characterized by the axioms, once the external inputs are given.

By studying the minimal model under certain initial conditions and known facts about agent beliefs and abilities and the structure of recipes, we can prove various results. The proofs consist of tracing the steps of the agents over time in the model. We include three results here: two positive, one negative. The two theorems show that if there is "enough" time allowed for planning a complex action all of whose subactions a single agent can do (Theorem 1) or subcontract some to other agents (Theorem 2), then the agent will do it. The Proposition shows that if one basic level action in the recipe of a complex action cannot be done by any agent, then no agent will get the intention to do the complex action. There are some models of the meta-theory where this result fails, but it is true in the intended model.

Theorem 1 *Suppose an agent is asked at time t to do an action a starting at time $t+s$, and suppose that the recipe tree for a has v levels and takes k units of time, where $s > 2v+2$. Further suppose the agent believes that it can do each subaction (at the needed time) itself, and that it in fact can do them. Then the agent will complete the task at time $t+s+k$.*

Proof Sketch:

We show the key formulas (abbreviated leaving out agent names and time) that become true as time changes. We do not show the inherited formulas but add some comments about them at the end. We also do not show other true formulas, such as $BL(z1)$, that are assumed to hold for all times.

t $ATD(a)$

t+1 $PotInt(a)$

t+2 $PotInt(b1)$, $PotInt(b2)$, ... the children node of a

...

t+v+1 $PotInt(z1)$, $PotInt(z2)$, ... the nodes at the bottom level

t+v+2 $Int(z1)$, $Int(z2)$, ...

...

t+2v+1 $Int(b1)$, $Int(b2)$, ...

t+2v+2 $Int(a)$

...

t+s $Ini(a)$

t+s+1 $Ini(b1)$

... The times for the Ini and Done of the nodes depend on the tree structure

t+s+k $Done(a)$

We note that $PotInt(a)$ will actually hold starting at time t and ending at time $t+s-1$. In fact, in this case all the potential intentions and intentions will hold from their earliest time as indicated above (e.g. $t+2$ for $PotInt(b1)$, $t+v+2$ for $Int(z1)$) until $t+s'-1$, where the subaction is supposed to be done at time $t+s'$. The theorem below generalizes the earlier one to the case of multiple agents.

Theorem 2 *Suppose an agent is asked at time t to do an action a starting at time $t+s$, and suppose that the recipe tree for a has v levels and takes k units of time, where $s > 4v+4$. Further suppose the agent for each subaction either believes that it can do it or can successfully subcontract it to another agent that believes that it can do it (at the needed time), and in fact the intending agent can do the action or subaction. Then the agent, with the assistance of the subcontracting agents will complete the task at time $t+s+k$.*

In this case two time periods must be added at each level for possibly asking another agent to do a task and waiting for the agent to get a potential intention for it.

Proposition 1 *If there is a basic level action b in the recipe for a that no agent believes that it can do, then no agent will get an intention to do a.*

Related Work

Intentions in the context of SharedPlans were studied in (Grosz & Kraus 1996), but no formal semantics were given. Our starting point in this paper was the axioms presented by Grosz and Kraus but our requirements for an agent having an intention is much stronger than those presented in (Grosz & Kraus 1996). There, an agent may have an intention also when having a partial plan. For example, the agent may have only partial knowledge about a recipe, but a plan how to complete it; it may have only potential intentions toward subactions. In order for the agent to have an intention, we require that it have a full detailed plan to do the action and that

it has adopted the appropriate intentions and beliefs. These requirements enable us to formally prove various properties about agent intentions and actions.

Since the definition of Int in our model is much stronger than the $Int.To$ of SharedPlans, the $PotInt$ predicate of our model plays a more important role in the agent's reasoning than the $Pot.Int.To$ does in (Grosz & Kraus 1996). In (Grosz & Kraus 1996) $Pot.Int.To$ is used only as an intermediate operator until $Int.To$ is adopted. In our model the $PotInt$ is kept for the duration of the agent's need for the associated intention and is used during the planning and for continuous verification of the minimal requirements for having the intention.

The SharedPlan model of collaborative planning uses the mental state model of plans (Pollack 1990). Bratman (Bratman 1987) also argues for a mental-state view of plans, emphasizing the importance of intentions to plans. He argues that intentions to do an action play three roles in rational action: having an intention to do an action constrains the other intentions an agent may adopt, focuses means-ends reasoning, and guides replanning. These roles are even more important for collaborative activity than for individual activity. In our model Int and $PotInt$ play these roles.

(Castelfranchi 1995) studies the notion of intention for describing and understanding the activities of groups and organizations in the context of improving the exchange between AI and social and management approaches to cooperative work. His motivation is different from our aim of developing a formal logic of beliefs and intentions.

There were several attempts to develop possible worlds semantics for intentions (Konolige & Pollack 1993; Cohen & Levesque 1990; Kraus, Sycara, & Evenchik 1998; Kumar *et al.* 2000; Wooldridge 2000). Some problems arise with these attempts such as that in most of them intentions are closed under Modus Ponens or under logical equivalence and that the relations between the action's recipe and the intention are not well defined. Using a syntactic approach provides more freedom in modeling the way agents' intentions change over time. See (Wooldridge 2000) for an excellent survey. An interesting dual treatment of agents that, like ours, has both an agent language ("first-person account") (Giacomo *et al.* 2002) and a meta language ("third-person account") (Lesperance 2001), uses the Golog family of languages based on the situation-calculus. Those papers (unlike our own) are more focussed on knowledge conditions than on intentions, and also do not take time-taken-to-plan into account.

Summary and Future Work

We presented a formal logical calculus that can be regarded as a meta-logic that describes the reasoning and activities of agents. The explicit representation of evolving time is an important feature of this approach. We dealt with the case where agents are assigned tasks for which a recipe is known. Recipes have a tree structure. An agent may subcontract some of the actions/subactions to other agents. Our emphasis is on developing a framework that models the beliefs, intentions, and actions of agents as they change over time. We present a syntactic approach and propose a minimal model semantics. Using this semantics, rather than possible world semantics, allows us to model agents activity more realistically and to prove several results to show that under the appropriate conditions the agents will act as desired.

We plan to extend this work in several ways. At present we have results only for strongly positive (agents always successfully subcontract actions/subactions and their beliefs about their activities are correct) and strongly negative (there is a subaction that no agent can do) cases. We will consider more complex situations. Also we have assumed that each task has only one recipe and so will deal with the case of multiple recipes. Additionally we will deal with agents doing subactions in parallel and situations where agents have SharedPlans (and not only subcontract actions).

References

Bratman, M. E. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Castelfranchi, C. 1995. Commitments: From individual intentions to groups and organizations. In *ICMAS 95*.

Cohen, P., and Levesque, H. 1990. Intention is choice with commitment. *Artificial Intelligence* 42:263–310.

Giacomo, G. D.; Lesperance, Y.; Levesque, H.; and Sardina, S. 2002. On the semantics of deliberation in indigolog—from theory to implementation. In *KR'02*.

Grant, J.; Kraus, S.; and Perlis, D. 2000. A logic for characterizing multiple bounded agents. *Autonomous Agents and Multi-Agent Systems Journal* 3(4):351–387.

Grosz, B. J., and Kraus, S. 1996. Collaborative plans for complex group activities. *AIJ* 86(2):269–357.

Konolige, K., and Pollack, M. E. 1993. A representationalist theory of intention. In *Proc. of IJCAI-93*, 390–395.

Kraus, S.; Sycara, K.; and Evenchik, A. 1998. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence* 104(1-2):1–69.

Kraus, S. 2001. *Strategic Negotiation in Multiagent Environments*. Cambridge, USA: MIT Press.

Kumar, S.; Huber, P.; McGee, D.; Cohen, P.; and Levesque, H. 2000. Semantics of agent communication languages for group interaction. In *AAAI 2000*, 42–47.

Lesperance, Y. 2001. On the epistemic feasibility of plans in multiagent systems specifications. In *ATAL'01*.

Pollack, M. E. 1990. Plans as complex mental attitudes. In Cohen, P.; Morgan, J.; and Pollack, M., eds., *Intentions in Communication*. Bradford Books, MIT Press.

Wooldridge, M. 2000. *Reasoning about Rational Agents*. The MIT Press.