A Logical Framework for Prioritized Goal Change

Shakil M. Khan
Dept. of Computer Science and Engineering
York University
Toronto, ON, Canada
skhan@cse.yorku.ca

Yves Lespérance
Dept. of Computer Science and Engineering
York University
Toronto, ON, Canada
lesperan@cse.yorku.ca

ABSTRACT

Most previous logical accounts of goals do not deal with prioritized goals and goal dynamics properly. Many are restricted to achievement goals. In this paper, we develop a logical account of goal change that addresses these deficiencies. In our account, we do not drop lower priority goals permanently when they become inconsistent with other goals and the agent's knowledge; rather, we make such goals inactive. We ensure that the agent's chosen goals/intentions are consistent with each other and the agent's knowledge. When the world changes, the agent recomputes her chosen goals and some inactive goals may become active again. This ensures that our agent maximizes her utility. We prove that the proposed account has desirable properties. We also discuss previous work on postulates for goal revision.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal Logic*

General Terms

Theory, Languages

Keywords

Prioritized goals, goal change, intention, logic of agency

1. INTRODUCTION

There has been much work on modeling agents' mental states, beliefs, goals, and intentions, and how they interact and lead to rational decisions about action. As well, there has been a lot of work on modeling belief change. But the dynamics of motivational attitudes has received much less attention. Most formal models of goal and goal change [3, 14, 17, 10, 24, 23] assume that all goals are equally important and many only deal with achievement goals. Moreover, most of these frameworks do not guarantee that an agent's goals will properly evolve when an action/event occurs, when the agent's beliefs/knowledge changes, or when a goal is adopted or dropped (one exception to this is the model of prioritized goals in [21]). Dealing with these issues is important for developing effective models of rational agency. It is also important for work on BDI agent programming languages, where handling declarative goals is an active research topic [28, 1].

In this paper, we present a formal model of prioritized goals and their dynamics that addresses some of these issues. In our framework, an agent can have multiple goals at different priority levels, possibly inconsistent with each other. We define intentions as the maximal set of highest priority goals that is consistent given the agent's knowledge. Our model of goals supports the specification of general temporally extended goals, not just achievement goals.

We start with a (possibly inconsistent) initial set of *prioritized goals* or desires that are totally ordered according to priority, and specify how these goals evolve when actions/events occur and the agent's knowledge changes. We define the agent's *chosen goals* or intentions in terms of this goal hierarchy. Our agents maximize their utility; they will abandon a chosen goal ϕ if an opportunity to commit to a higher priority but inconsistent with ϕ goal arises. To this end, we keep all prioritized goals in the goal base unless they are explicitly dropped. At every step, we compute an optimal set of chosen goals given the hierarchy of prioritized goals, preferring higher priority goals such that chosen goals are consistent with each other and with the agent's knowledge. Thus at any given time, some goals in the hierarchy are active, i.e. chosen, while others are inactive. Some of these inactive goals may later become active, e.g. if a higher priority active goal that is currently blocking an inactive goal becomes impossible.

Our formalization of prioritized goals ensures that the agent always tries to maximize her utility, and as such displays an idealized form of rationality. In the Section 5, we discuss how this relates to Bratman's [2] theory of practical reasoning. We use an action theory based on the situation calculus [12] along with our formalization of paths in the situation calculus as our base formalism.

In the next section, we outline our base framework. In Section 3, we formalize *paths* in the situation calculus to support modeling goals. In Section 4, we present our model of prioritized goals. In Section 5 and 6, we present our formalization of goal dynamics and discuss some of its properties. Then in the last section, we summarize our results, discuss previous work in this area, and point to possible future work.

Cite as: A Logical Framework for Prioritized Goal Change, Shakil M. Khan and Yves Lespérance, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 283-290

2. ACTION AND KNOWLEDGE

Our base framework for modeling goal change is the situation calculus [12] as formalized in [16]. In this framework, a possible state of the domain is represented by a situation. There is a set of initial situations corresponding to the ways the agent believes the domain might be initially, i.e. situations in which no actions have yet occurred. Init(s) means that s is an initial situation. The actual initial state is represented by a special constant S_0. There is a distinguished binary function symbol do where $do(a, s)$ denotes the successor situation to s resulting from performing the action a. Thus the situations can be viewed as a set of trees, where the root of each tree is an initial situation and the arcs represent actions. Relations (and functions) whose truth values vary from situation to situation, are called relational (functional, resp.) fluents, and are denoted by predicate (function, resp.) symbols taking a situation term as their last argument. There is a special predicate Poss(a, s) used to state that action a is executable in situation s.

Our framework uses a theory D_{basic} that includes the following set of axioms:[1] (1) action precondition axioms, one per action a characterizing Poss(a, s), (2) successor state axioms (SSA), one per fluent, that succinctly encode both effect and frame axioms and specify exactly when the fluent changes [16], (3) initial state axioms describing what is true initially including the mental states of the agents, (4) unique name axioms for actions, and (5) domain-independent foundational axioms describing the structure of situations [11].

Following [13, 20], we model knowledge using a possible worlds account adapted to the situation calculus. $K(s', s)$ is used to denote that in situation s, the agent thinks that she could be in situation s'. Using K, the knowledge of an agent is defined as:[2] $\text{Know}(\Phi, s) \stackrel{\text{def}}{=} \forall s'.\ K(s', s) \supset \Phi(s')$, i.e. the agent knows Φ in s if Φ holds in all of her K-accessible situations in s. 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. In our framework, the dynamics of knowledge is specified using a SSA for K that supports knowledge expansion as a result of sensing actions. As shown in [20], the constraints on K then continue to hold after any sequence of actions since they are preserved by the SSA for K. We also assume that all actions are public, i.e. whenever an action (including exogenous events) occurs, the agent learns that it has happened. Note that, we work with knowledge rather than belief. Although much of our formalization should extend to the latter, we leave this for future work.

3. PATHS IN THE SITUATION CALCULUS

To support modeling temporally extended goals, we introduce a new sort of *paths*, with (possibly sub/super-scripted) variables p ranging over paths. A path is essentially an infinite sequence of situations, where each situation along the path can be reached by performing some *executable* action in the preceding situation. We introduce a predicate OnPath(p, s), meaning that the situation s is on path p. Also, Starts(p, s) means that s is the starting situation of path p. A path p starts with s iff s is the earliest situation on p:[3]

Axiom 1.

$$\text{Starts}(p, s) \equiv \text{OnPath}(p, s) \wedge \forall s'.\ \text{OnPath}(p, s') \supset s \leq s'.$$

In the standard situation calculus, paths are implicitly there, and a path can be viewed as a pair (s, F) consisting of a situation s representing the starting situation of the path, and a function F from situations to actions (called *Action Selection Functions* (ASF) or strategies in [23]), such that from the starting situation s, F defines an infinite sequence of situations by specifying an action for every situation starting from s. Thus, one way of axiomatizing paths is by making them correspond to such pairs (s, F):

Axiom 2.

$$\begin{aligned}&\forall p.\ \text{Starts}(p, s) \supset (\exists F.\ \text{Executable}(F, s)\\ &\qquad \wedge \forall s'.\ \text{OnPath}(p, s') \equiv \text{OnPathASF}(F, s, s')),\\ &\forall F, s.\ \text{Executable}(F, s) \supset \exists p.\ \text{Starts}(p, s)\\ &\qquad \wedge \forall s'.\ \text{OnPathASF}(F, s, s') \equiv \text{OnPath}(p, s').\end{aligned}$$

This says that for every path there is an executable ASF that produces exactly the sequence of situations on the path from its starting situation. Also, for every executable ASF and situation, there is a path that corresponds to the sequence of situations produced by the ASF starting from that situation.

$$\begin{aligned}&\text{OnPathASF}(F, s, s') \stackrel{\text{def}}{=}\\ &\qquad s \leq s' \wedge \forall a, s^*.\ s < do(a, s^*) \leq s' \supset F(s^*) = a,\\ &\text{Executable}(F, s) \stackrel{\text{def}}{=}\\ &\qquad \forall s'.\ \text{OnPathASF}(F, s, s') \supset \text{Poss}(F(s'), s').\end{aligned}$$

Here, OnPathASF(F, s, s') [18] means that the situation sequence defined by (s, F) includes the situation s'. Also, the situation sequence encoded by a strategy F and a starting situation s is executable iff for all situations s' on this sequence, the action selected by F in s' is executable in s'.

We will use both state and path formulae. A state formula $\Phi(s)$ is a formula that has a free situation variable s in it, whereas a path formula $\phi(p)$ is one that has a free path variable p.[4] State formulae are used in the context of knowledge while path formulae are used in that of goals. Note that, by incorporating infinite paths in our framework, we can evaluate goals over these and handle arbitrary temporally extended goals; thus, unlike some other situation calculus based accounts where goal formulae are evaluated w.r.t. finite paths (e.g. [21]), we can handle for example unbounded maintenance goals. Also, while our account is restricted to infinite paths, one could argue that situations where no action is possible are artificial.

We next define some useful constructs. A state formula Φ *eventually holds* over the path p if Φ holds in some situation that is on p, i.e. $\Diamond\Phi(p) \stackrel{\text{def}}{=} \exists s'.\ \text{OnPath}(p, s') \wedge \Phi(s')$. Other Temporal Logic operators can be defined similarly, e.g. always Φ: $\Box\Phi(p)$.

[1] We will be quantifying over formulae, and thus assume D_{basic} includes axioms for encoding of formulae as first order terms, as in [22].

[2] Φ is a state formula that can contain a situation variable, *now*, in the place of situation terms. We often suppress *now* when the intent is clear from the context.

[3] In the following, $s < s'$ means that s' can be reached from s by performing a sequence of executable actions. $s \leq s'$ is an abbreviation for $s < s' \vee s = s'$.

[4] As with state formulae, we often suppress the path variable p in a path formula $\phi(p)$ when the intent is clear from the context.

An agent *knows* in s that ϕ has become *inevitable* if ϕ holds over all paths that starts with a K-accessible situation in s, i.e. $\text{KInevitable}(\phi, s) \stackrel{\text{def}}{=} \forall p.\ \text{Starts}(p, s') \wedge K(s', s) \supset \phi(p)$. An agent knows in s that ϕ is impossible if she knows that $\neg\phi$ is inevitable in s, i.e. $\text{KImpossible}(\phi, s) \stackrel{\text{def}}{=} \text{KInevitable}(\neg\phi, s)$.

Thirdly, we define what it means for a path p' to be a suffix of another path p w.r.t. a situation s:

$$\text{Suffix}(p', p, s) \stackrel{\text{def}}{=} \text{OnPath}(p, s) \wedge \text{Starts}(p', s) \wedge \forall s'.\ s' \geq s \supset \text{OnPath}(p, s') \equiv \text{OnPath}(p', s').$$

Fourthly, $\text{SameHist}(s_1, s_2)$ means that the situations s_1 and s_2 share the same history of actions, but perhaps starting from different initial situations:

Axiom 3.

$$\text{SameHist}(s_1, s_2) \equiv (\text{Init}(s_1) \wedge \text{Init}(s_2)) \vee (\exists a, s_1', s_2'.\ s_1 = do(a, s_1') \wedge s_2 = do(a, s_2') \wedge \text{SameHist}(s_1', s_2')).$$

Finally, we say that ϕ has become *inevitable* in s if ϕ holds over all paths that starts with a situation that has the same history as s: $\text{Inevitable}(\phi, s) \stackrel{\text{def}}{=} \forall p, s'.\ \text{Starts}(p, s') \wedge \text{SameHist}(s', s) \supset \phi(p)$.

4. PRIORITIZED GOALS

Most work on formalizing goals only deals with static goal semantics and not their dynamics. In this section, we formalize goals or desires with different priorities which we call *prioritized goals* (p-goals, henceforth). These p-goals are not required to be mutually consistent and need not be actively pursued by the agent. In terms of these, we define the consistent set of *chosen goals* or intentions (c-goals, henceforth) that the agent is committed to. In Section 5, we formalize goal dynamics by providing a SSA for p-goals. The agent's c-goals are automatically updated when her p-goals change.

Not all of the agent's goals are equally important to her. Thus, it is useful to support a priority ordering over goals. This information can be used to decide which of the agent's c-goals should no longer be actively pursued in case they become mutually inconsistent. We specify each p-goal by its own accessibility relation/fluent G. A path p is G-accessible at priority level n in situation s (denoted by $G(p, n, s)$) iff the goal of the agent at level n is satisfied over this path and if it starts with a situation that has the same action history as s. The latter requirement ensures that the agent's p-goal-accessible paths reflect the actions that have been performed so far. A smaller n represents higher priority, and the highest priority level is 0. Thus here we assume that the set of p-goals are totally ordered according to priority (given a priority level, the agent can have only one goal at that level, possibly a complex one, e.g. a conjunctive goal). We say that an agent has the p-goal that ϕ at level n in situation s iff ϕ holds over all paths that are G-accessible at n in s:

$$\text{PGoal}(\phi, n, s) \stackrel{\text{def}}{=} \forall p.\ G(p, n, s) \supset \phi(p).$$

To be able to refer to all the p-goals of the agent at some given priority level, we also define *only p-goals*.

$$\text{OPGoal}(\phi, n, s) \stackrel{\text{def}}{=} \text{PGoal}(\phi, n, s) \wedge \forall p.\ \phi(p) \supset G(p, n, s).$$

An agent has the only p-goal that ϕ at level n in situation s iff ϕ is a p-goal at n in s, and any path over which ϕ holds is G-accessible at n in s.

A domain theory for our framework D includes the axioms of a theory D_{basic} as in the previous section, the axiomatization of paths, i.e. axioms 1-3, domain dependent initial goal axioms (see below), the domain independent axioms 4-6 and the definitions that appear throughout this paper. We allow the agent to have infinitely many goals. We expect the modeler to include some specification of what paths are G accessible at the various levels initially. We call these axioms *initial goal axioms*. In many cases, the user will want to specify a finite set of initial p-goals. This can be done by providing a set of axioms as in the example below. But in general, an agent can have a countably infinite set of p-goals, e.g. an agent that has the p-goal at level n to know what the n-th prime number is for all n. The agent's set of p-goals can even be specified incompletely, e.g. the theory might not specify what the p-goals at some level are initially.

We use the following as a running example. We have an agent who initially has the following three p-goals: $\phi_0 = \Box\text{BeRich}$, $\phi_1 = \Diamond\text{GetPhD}$, and $\phi_2 = \Box\text{BeHappy}$ at level 0, 1, and 2, respectively. This domain can be specified using the following two initial goal axioms:

(a) $\text{Init}(s) \supset$
$$((G(p, 0, s) \equiv \text{Starts}(p, s') \wedge \text{Init}(s') \wedge \phi_0(p)) \wedge (G(p, 1, s) \equiv \text{Starts}(p, s') \wedge \text{Init}(s') \wedge \phi_1(p)) \wedge (G(p, 2, s) \equiv \text{Starts}(p, s') \wedge \text{Init}(s') \wedge \phi_2(p))),$$

(b) $\forall n, p, s.\ \text{Init}(s) \wedge n \geq 3 \supset$
$$(G(p, n, s) \equiv \text{Starts}(p, s') \wedge \text{Init}(s')).$$

(a) specifies the p-goals ϕ_0, ϕ_1, ϕ_2 (from highest to lowest priority) of the agent in the initial situations, and makes $G(p, n, s)$ true for every path p that starts with an initial situation and over which ϕ_n holds, for $n = 0, 1, 2$; each of them defines a set of initial goal paths for a given priority level, and must be consistent. (b) makes $G(p, n, s)$ true for every path p that starts with an initial situation for $n \geq 3$. Thus at levels $n \geq 3$, the agent has the trivial p-goal that she be in an initial situation. Assume that while initially the agent knows that all of her p-goals are individually achievable, she knows that her p-goal $\Diamond\text{GetPhD}$ is inconsistent with her highest priority p-goal $\Box\text{BeRich}$ as well as with her p-goal $\Box\text{BeHappy}$, while the latter are consistent with each other. It can be shown that in our example, we have $D \models \text{OPGoal}(\phi_i(p) \wedge \text{Starts}(p, s) \wedge \text{Init}(s), i, S_0)$, for $i = 0, 1, 2$. Also, for any $n \geq 3$, we have $D \models \text{OPGoal}(\text{Starts}(p, s) \wedge \text{Init}(s), n, S_0)$.

While p-goals or desires are allowed to be known to be impossible to achieve, an agent's c-goals or intentions must be realistic. Not all of the G-accessible paths are realistic in the sense that they start with a K-accessible situation. To filter these out, we define *realistic* p-goal accessible paths:

$$G_R(p, n, s) \stackrel{\text{def}}{=} G(p, n, s) \wedge \text{Starts}(p, s') \wedge K(s', s).$$

Thus G_R prunes out the paths from G that are known to be impossible, and since we define c-goals in terms of realistic p-goals, this ensures that c-goals are realistic. We say that an agent has the *realistic p-goal* that ϕ at level n in situation s iff ϕ holds over all paths that are G_R-accessible at n in s:

$$\text{RPGoal}(\phi, n, s) \stackrel{\text{def}}{=} \forall p.\ G_R(p, n, s) \supset \phi(p).$$

Using realistic p-goals, we next define c-goals. The idea of how we specify c-goal-accessible paths is as follows: the set of G_R-accessibility relations represents a set of prioritized temporal propositions that are candidates for the agent's

c-goals. Given G_R, in each situation we want to compute the agent's c-goals such that it is the *maximal consistent* set of higher priority realistic p-goals. We do this iteratively starting with the set of all realistic paths (i.e. paths that start with a K-accessible situation). At each iteration we compute the intersection of this set with the next highest priority set of G_R-accessible paths. If the intersection is not empty, we thus obtain a new chosen set of paths at level i. We call a p-goal chosen by this process an *active* p-goal. If on the other hand the intersection is empty, then it must be the case that the p-goal represented by this level is either in conflict with another active higher priority p-goal/a combination of two or more active higher priority p-goals, or is known to be impossible. In that case, that p-goal is ignored (i.e. marked as inactive), and the chosen set of paths at level i is the same as at level $i-1$. Axiom 4 specifies this intersection:[5]

AXIOM 4.

$$
\begin{array}{l}
G_{\cap}(p,n,s) \equiv \\
\quad \textbf{if } (n=0) \textbf{ then} \\
\quad\quad \textbf{if } \exists p'.\ G_R(p',n,s) \textbf{ then } G_R(p,n,s) \\
\quad\quad \textbf{else } \mathrm{Starts}(p,s') \wedge K(s',s) \\
\quad \textbf{else} \\
\quad\quad \textbf{if } \exists p'.(G_R(p',n,s) \wedge G_{\cap}(p',n-1,s)) \\
\quad\quad\quad \textbf{then } (G_R(p,n,s) \wedge G_{\cap}(p,n-1,s)) \\
\quad\quad \textbf{else } G_{\cap}(p,n-1,s).
\end{array}
$$

Using this, we define what it means for an agent to have a c-goal at some level n:

$$\mathrm{CGoal}(\phi,n,s) \stackrel{\mathrm{def}}{=} \forall p.\ G_{\cap}(p,n,s) \supset \phi(p),$$

i.e. an agent has the c-goal at level n that ϕ if ϕ holds over all paths that are in the prioritized intersection of the set of G_R-accessible paths up to level n.

We define c-goals in terms of c-goals at level n:

$$\mathrm{CGoal}(\phi,s) \stackrel{\mathrm{def}}{=} \forall n.\ \mathrm{CGoal}(\phi,n,s),$$

i.e., the agent has the c-goal that ϕ if for any level n, ϕ is a c-goal at n.

In our example, the agent's realistic p-goals are $\Box$BeRich, $\Diamond$GetPhD, and $\Box$BeHappy in order of priority. The $G_{\cap}$-accessible paths at level 0 in S_0 are the ones that start with a K-accessible situation and where $\Box$BeRich holds. The $G_{\cap}$-accessible paths at level 1 in S_0 are the same as at level 0, since there are no realistic path over which both $\Diamond$GetPhD and $\Box$BeRich hold. Finally, the $G_{\cap}$-accessible paths at level 2 in S_0 are those that start with a K-accessible situation and over which $\Box$BeRich $\wedge$ $\Box$BeHappy holds. Also, it can be shown that initially our example agent has the c-goals that $\Box$BeRich and $\Box$BeHappy, but not $\Diamond$GetPhD.

Note that by our definition of c-goals, the agent can have a c-goal that ϕ in situation s for various reasons: 1) ϕ is known to be inevitable in s; 2) ϕ is an active p-goal at some level n in s; 3) ϕ is a consequence of two or more active p-goals at different levels in s. To be able to refer to c-goals for which the agent has a primitive motivation, i.e. c-goals that result from a single active p-goal at some priority level n, in contrast to those that hold as a consequence of two or more active p-goals at different priority levels, we define *primary* c-goals:

$$\mathrm{PrimCGoal}(\phi,s) \stackrel{\mathrm{def}}{=}$$
$$\exists n.\ \mathrm{PGoal}(\phi,n,s) \wedge \exists p.\ G(p,n,s) \wedge G_{\cap}(p,n,s).$$

That is, an agent has the primary c-goal that ϕ in situation s, if ϕ is a p-goal at some level n in s, and if there is a G-accessible path p at n in s that is also in the prioritized intersection of G_R-accessible paths up to n in s. The last two conjuncts are required to ensure that n is an active level. Thus if an agent has a primary c-goal that ϕ, then she also has the c-goal that ϕ, but not necessarily vice-versa. It can be shown that initially our example agent has the primary c-goals that $\Box$BeRich and $\Box$BeHappy, but not their conjunction. To some extent, this shows that primary c-goals are not closed under logical consequence.

5. GOAL DYNAMICS

An agent's goals change when her knowledge changes as a result of the occurrence of an action (including exogenous events), or when she adopts or drops a goal. We formalize this by specifying how p-goals change. C-goals are then computed using (realistic) p-goals in every new situation as above.

We introduce two actions for adopting and dropping a p-goal, $adopt(\phi,n)$ and $drop(\phi)$. The action precondition axioms for these are as follows:

AXIOM 5.

$$\mathrm{Poss}(adopt(\phi,n),s) \equiv \neg\exists n'.\ \mathrm{PGoal}(\phi,n',s),$$
$$\mathrm{Poss}(drop(\phi),s) \equiv \exists n.\ \mathrm{PGoal}(\phi,n,s).$$

That is, an agent can adopt (drop) the p-goal that ϕ at level n, if she does not (does, resp.) already have ϕ as her p-goal at some level.

In the following, we specify the dynamics of p-goals by giving the SSA for G and discuss each case, one at a time:

AXIOM 6 (SSA FOR G).

$$
\begin{array}{l}
G(p,n,do(a,s)) \equiv \\
\quad \forall \phi,m.\ (a \neq adopt(\phi,m) \wedge a \neq drop(\phi) \wedge \\
\qquad\qquad \mathrm{Progressed}(p,n,a,s)) \\
\quad \vee \exists \phi,m.\ (a = adopt(\phi,m) \wedge \mathrm{Adopted}(p,n,m,a,s,\phi)) \\
\quad \vee \exists \phi.\ (a = drop(\phi) \wedge \mathrm{Dropped}(p,n,a,s,\phi)).
\end{array}
$$

The overall idea of the SSA for G is as follows. First of all, to handle the occurrence of a non-adopt/drop (i.e. regular) action a, we progress all G-accessible paths to reflect the fact that this action has just happened; this is done using the $\mathrm{Progressed}(p,n,a,s)$ construct, which replaces each G-accessible path p' with starting situation s', by its suffix p provided that it starts with $do(a,s')$:

$$\mathrm{Progressed}(p,n,a,s) \stackrel{\mathrm{def}}{=}$$
$$\exists p',s'.\ G(p',n,s) \wedge \mathrm{Starts}(p',s') \wedge \mathrm{Suffix}(p,p',do(a,s')).$$

Any path over which the next action performed is not a is eliminated from the respective G-accessibility level.

Secondly, to handle adoption of a p-goal ϕ at level m, we add a new proposition containing the p-goal to the agent's goal hierarchy at m by modifying the G-relation accordingly. The G-accessible paths at all levels above m are progressed as above. The G-accessible paths at level m are the ones that

[5] **if** ϕ **then** δ_1 **else** δ_2 is an abbreviation for $(\phi \supset \delta_1) \wedge (\neg\phi \supset \delta_2)$.

share the same history with $do(a, s)$ and over which ϕ holds. The G-accessible paths at all levels below m are the ones that can be obtained by progressing the level immediately above it. Thus the agent acquires the p-goal that ϕ at level m, and all the p-goals with priority m or less in s are pushed down one level in the hierarchy.

$$\begin{aligned}&\text{Adopted}(p, n, m, a, s, \phi) \stackrel{\text{def}}{=}\\ &\quad \textbf{if } (n < m) \textbf{ then } \text{Progressed}(p, n, a, s)\\ &\quad \textbf{else if } (n = m) \textbf{ then } \exists s'.\ \text{Starts}(p, s')\\ &\qquad\qquad \wedge \text{SameHist}(s', do(a, s)) \wedge \phi(p)\\ &\quad \textbf{else } \text{Progressed}(p, n-1, a, s).\end{aligned}$$

Finally, to handle the dropping of a p-goal ϕ, we replace the propositions that imply the dropped goal in the agent's goal hierarchy by the trivial proposition that the history of actions in the current situation has occurred. Thus, in addition to progressing all G-accessible paths as above, we add back all paths that share the same history with $do(a, s)$ to the existing G-accessibility levels where the agent has the p-goal that ϕ.

$$\begin{aligned}&\text{Dropped}(p, n, a, s, \phi) \stackrel{\text{def}}{=} \textbf{if } \text{PGoal}(\phi, n, s)\\ &\qquad \textbf{then } \exists s'.\ \text{Starts}(p, s') \wedge \text{SameHist}(s', do(a, s))\\ &\qquad \textbf{else } \text{Progressed}(p, n, a, s).\end{aligned}$$

Returning to our example, recall that our agent has the c-goals/active p-goals in S_0 that $\Box$BeRich and $\Box$BeHappy, but not $\Diamond$GetPhD, since the latter is inconsistent with her higher priority p-goal $\Box$BeRich. Assume that, after the exogenous event/action *goBankrupt* happens in S_0, the p-goal $\Box$BeRich becomes impossible. Then in $S_1 = do(goBankrupt, S_0)$, the agent has the c-goal that $\Diamond$GetPhD, but not $\Box$BeRich nor $\Box$BeHappy; $\Box$BeRich is excluded from the set of c-goals since it has become impossible to achieve (i.e. unrealistic). Also, since her higher priority p-goal $\Diamond$GetPhD is inconsistent with her p-goal $\Box$BeHappy, the agent will make $\Box$BeHappy inactive.

Note that, while it might be reasonable to drop a p-goal (e.g. $\Diamond$GetPhD) that is in conflict with another higher priority active p-goal (e.g. $\Box$BeRich), in our framework we keep such p-goals around. The reason for this is that although $\Box$BeRich is currently inconsistent with $\Diamond$GetPhD, the agent might later learn that $\Box$BeRich has become impossible to bring about (e.g. after *goBankrupt* occurs), and then might want to pursue $\Diamond$GetPhD. Thus, it is useful to keep these inactive p-goals since this allows the agent to maximize her utility (that of her chosen goals) by taking advantage of such opportunities. As mentioned earlier, c-goals are our analogue to intentions. Recall that Bratman's [2] model of intentions limits the agent's practical reasoning – agents do not always optimize their utility and don't always reconsider all available options in order to allocate their reasoning effort wisely. In contrast to this, our c-goals are defined in terms of the p-goals, and at every step, we ensure that the agent's c-goals maximize her utility so that these are the set of highest priority goals that are consistent given the agent's knowledge. Thus, our notion of c-goals is not as persistent as Bratman's notion of intentions. For instance as mentioned above, after the action *goBankrupt* happens in S_0, the agent will lose the c-goal that $\Box$BeHappy, although she did not drop it and it did not become impossible or achieved. In this sense, our model is that of an idealized agent. There is a tradeoff between optimizing the agent's chosen set of prioritized goals and being committed to chosen goals. In our framework, chosen goals behave like intentions with an automatic filter-override mechanism [2] that forces the agent to drop her chosen goals when opportunities to commit to other higher priority goals arise. In the future, it would be interesting to develop a logical model that captures the pragmatics of intention reconsideration by supporting control over it.

6. PROPERTIES

We now show that our formalization has some desirable properties. Some of these (e.g. Proposition 1, 3(a), 4, 5) are analogues of the AGM postulates [8]. First we show that c-goals are consistent:

PROPOSITION 1 (CONSISTENCY).
$$D \models \forall s.\ \neg\text{CGoal}(\text{False}, s).$$

Thus, the agent cannot have both ϕ and $\neg\phi$ c-goals in a situation s. Even if all of the agent's p-goals become known to be impossible, the set of c-goal-accessible paths will be precisely those that starts with a K-accessible situation, and thus the agent will only choose the propositions that are known to be inevitable.

We also have the property of realism [3], i.e. if an agent knows that something has become inevitable, then she has this as a c-goal:

PROPOSITION 2 (REALISM).
$$D \models \forall \phi, s.\ \text{KInevitable}(\phi, s) \supset \text{CGoal}(\phi, s).$$

Note that this is not necessarily true for p-goals and primary c-goals – an agent may know that something has become inevitable and not have it as her p-goal/primary c-goal, which is intuitive. While the property of realism is often criticized [14, 15], one should view these inevitable goals as something that hold in the worlds that the agent intends to bring about, rather than something that the agent is actively pursuing.

A consequence of Proposition 1 and 2 is that an agent does not have a c-goal that is known to be impossible:

COROLLARY 1.
$$D \models \forall \phi, s.\ \text{CGoal}(\phi, s) \supset \neg\text{KImpossible}(\phi, s).$$

We next discuss some properties of the framework w.r.t. goal change. Proposition 3 says that (a) an agent acquires the p-goal that ϕ at level n after she adopts it at n, (b) that she acquires the primary c-goal (and thus the p-goal and c-goal) that ϕ after she adopts it at some level n in s, provided that she does not have the c-goal in s that $\neg\phi$ next, and (c) that she acquires the primary c-goal that ϕ after she adopts it at some level n in s provided that it is consistent with her c-goals up to level $n-1$; this holds even if she has the inconsistent c-goal at some level that $\neg\phi$ next, provided that she adopts ϕ at a higher priority than all such inconsistent goals.

PROPOSITION 3 (ADOPTION).

(a) $D \models \text{PGoal}(\phi, n, do(adopt(\phi, n), s))$,

(b) $D \models \neg\text{CGoal}(\neg\exists s', p'.\ \text{Starts}(s') \wedge$
$\qquad \text{Suffix}(p', do(adopt(\phi, n), s')) \wedge \phi(p'), s)$
$\qquad \supset \text{PrimCGoal}(\phi, do(adopt(\phi, n), s))$,

(c) $D \models \neg\text{CGoal}(\neg\exists s', p'.\ \text{Starts}(s') \wedge$
$\qquad \text{Suffix}(p', do(adopt(\phi, n), s')) \wedge \phi(p'), n-1, s)$
$\qquad \supset \text{PrimCGoal}(\phi, do(adopt(\phi, n), s))$.

It should be noted that (c) above is a specialization of (b) for dealing with prioritized goals. Recall that the agent's chosen goals act as a filter for adopting newer goals. (c) ensures that the agent takes into consideration the priorities of goals when adopting a new goal that is inconsistent with her current chosen goals.

We can also show that after dropping the p-goal that ϕ at n in s, an agent does not have the p-goal (and thus the primary c-goal) that the progression of ϕ at n, i.e. ProgOf($\phi, drop(\phi), s$), provided that ProgOf($\phi, drop(\phi), s$) is not inevitable in $do(drop(\phi), s)$.

PROPOSITION 4 (DROP).

$$D \models \text{PGoal}(\phi, n, s) \wedge \neg\text{Inevitable}(\text{ProgOf}(\phi, drop(\phi), s), do(drop(\phi), s)) \supset \neg\text{PGoal}(\text{ProgOf}(\phi, drop(\phi), s), n, do(drop(\phi), s)),$$

where,

$$\text{ProgOf}(\phi, a, s) \stackrel{\text{def}}{=} \exists p', s'.\ \text{Starts}(p', s') \wedge \text{Suffix}(p', do(a, s')) \wedge \phi(p').$$

Note that, this does not hold for CGoal, as ϕ could still be a consequence of her remaining primary c-goals.

The next property states that adopting/dropping logically equivalent goals has the same result.

PROPOSITION 5 (EXTENSIONALITY).

$$D \models \phi_1 \equiv \phi_2 \supset \forall\psi.\ [\text{PrimCGoal}(\psi, do(adopt(\phi_1, n), s)) \equiv \text{PrimCGoal}(\psi, do(adopt(\phi_2, n), s))] \wedge \forall\psi.\ [\text{PrimCGoal}(\psi, do(drop(\phi_1), s)) \equiv \text{PrimCGoal}(\psi, do(drop(\phi_2), s))].$$

The next few properties concern the persistence of these motivational attitudes. First we have a persistence property for achievement realistic p-goals:

PROPOSITION 6 (PERSISTENCE OF ACHV.RPGOALS).

$$D \models \text{RPGoal}(\Diamond\Phi, n, s) \wedge \text{Know}(\neg\Phi, s) \wedge \forall\psi.\ a \neq drop(\psi) \supset \exists n'.\ \text{RPGoal}(\Diamond\Phi, n', do(a, s)).$$

This says that if an agent has a realistic p-goal that $\Diamond\Phi$ in s, then she will retain this realistic p-goal after some action a has been performed in s, provided that she knows that Φ has not yet been achieved, and a is not the action of dropping a p-goal. Note that, we do not need to ensure that $\Diamond\Phi$ is consistent with higher priority active p-goals, since the SSA for G does not automatically drop such incompatible p-goals from the goal hierarchy. Also, the level n where Φ is a p-goal may change, e.g. if the action performed is an adopt action with priority higher than or equal to n.

For achievement chosen goals we have the following:

PROPOSITION 7 (PERSISTENCE OF ACHV. CGOALS).

$$D \models \text{OPGoal}(\Diamond\Phi \wedge \exists s'.\ \text{Starts}(s') \wedge \text{SameHist}(s'), n, s) \wedge \text{CGoal}(\Diamond\Phi, s) \wedge \text{Know}(\neg\Phi, s) \wedge \forall\psi.\ a \neq drop(\psi) \wedge \forall\psi, m.\ \neg(a = adopt(\psi, m) \wedge m \leq n) \wedge \neg\text{CGoal}(\neg\Diamond\Phi, n-1, do(a, s)) \supset \text{CGoal}(\Diamond\Phi, n, do(a, s)).$$

Thus, in situation s, if an agent has the only p-goal at level n that $\Diamond\Phi$ and that the correct history of actions in s has been performed, and if $\Diamond\Phi$ is also a chosen goal in s (and thus she has the primary c-goal that $\Diamond\Phi$), then she will retain the c-goal that $\Diamond\Phi$ at level n after some action a has been performed in s, provided that: she knows that Φ has not yet been achieved, that a is not the action of dropping a p-goal, that a is not the action of adopting a p-goal at some higher priority level than n or at n, and that at level $n-1$ the agent does not have the c-goal that $\neg\Diamond\Phi$, i.e. $\Diamond\Phi$ is consistent with higher priority c-goals.

Note that, this property also follows if we replace the consequent with CGoal($\Diamond\Phi, do(a, s)$), and thus it deals with the persistence of c-goals. Note however that, it does not hold if we replace the OPGoal in the antecedent with PGoal; the reason for this is that the agent might have a p-goal at level n in s that ϕ and the c-goal in s that ϕ, but not have ϕ as a primary c-goal in s, e.g. n might be an inactive level because another p-goal at n has become impossible, and ϕ could be a c-goal in s because it is a consequence of two other primary c-goals. Thus even if $\neg\phi$ is not a c-goal after a has been performed in s, there is no guarantee that the level n will be active in $do(a, s)$ or that all the active p-goals that contributed to ϕ in s are still active.

7. DISCUSSION AND FUTURE WORK

In this paper, we presented a formalization of prioritized goals and their dynamics. Our formalization ensures that an agent's chosen goals are always consistent and that her goals properly evolve as a result of regular actions as well as of adopting and dropping goals. Although we made some simplifying assumptions, in this paper we have focused on developing an expressive framework that captures an idealized form of rationality without worrying about tractability. It would be desirable to study restricted fragments of the logic where reasoning is tractable. Also, before defining more limited forms of rationality, one should have a clear specification of what ideal rationality really is so that one understands what compromises are being made.

While in our account chosen goals are closed under logical consequence, primary c-goals are not. Thus, our formalization of primary c-goals is related to the non-normal modal formalizations of intentions found in the literature [10], and as such it does not suffer from the side-effect problem [3]. For instance, in our framework an agent can have the primary c-goal to get her teeth fixed and know that this always involves pain, but not have the primary c-goal to have pain.

Our framework can be extended to model subgoal adoption and the dependencies between goals and the subgoals and plans adopted to achieve them. The latter is important since subgoals and plans adopted to bring about a goal should be dropped when the parent goal becomes impossible, is achieved, or is dropped. One way of handling this is to ensure that the adoption of a subgoal ψ w.r.t. a parent goal ϕ adds a new p-goal that contains *both this subgoal and this parent goal*, i.e. $\psi \wedge \phi$. This ensures that when the parent goal is dropped, the subgoal is also dropped, since when we drop the parent goal ϕ, we drop all the p-goals at all G-accessibility levels that imply ϕ including $\psi \wedge \phi$.

Also, since we are using the situation calculus, we can easily represent procedural goals/plans, e.g. the goal to do a_1 and then a_2 can be written as: PGoal(Starts(p, s_1) $\wedge$ OnPath(p, s) $\wedge$ $s = do(a_2, do(a_1, s_1))$), $0, S_0$). Golog [7] can be used to represent complex plans/programs. So we can model the adoption of plans as subgoals.

Recently, there have been a few proposals that deal with goal change. Shapiro *et al.* [22] present a situation calculus based framework where an agent adopts a goal when she is requested to do so, and remains committed to this goal unless the requester cancels this request; a goal is retained even if the agent learns that it has become impossible, and in this case the agent's goals become inconsistent. Shapiro and Brewka [21] modify this framework to ensure that goals are dropped when they are believed to be impossible or when they are achieved. Their account is similar to ours in the sense that they also assume a priority ordering over the set of (in their case, requested) goals, and in every situation they compute chosen goals by computing a maximal consistent goal set that is also compatible with the agent's beliefs. In their framework, goals are only partially ordered and inconsistencies between goals at the same level (given goals at higher levels and knowledge) can be resolved differently in different models. In fact, the agent's chosen goals in $do(a, s)$ in a model may be quite different from her goals in s, although a did not make any of her goals in s impossible or inconsistent with higher priority goals, simply because the inconsistencies between goals at the same priority level are resolved differently in s and $do(a, s)$. This is rather unintuitive. Note that, while one might argue that a partial order over goals might be more general, allowing this means that additional control information is required to obtain a single goal state after the agent's goals change. In other words, the problem with a partial ordering is that it does not specify what a rational agent should do when two of her goals that have equal priority become inconsistent with each other. Also, we provide a more expressive formalization of prioritized goals – we model goals using infinite paths, and thus can model many types of goals that they cannot. Finally they model prioritized goals by treating the agent's p-goals as an arbitrary set of temporal formulae, and then defining the set of c-goals as a subset of the p-goals. However, our possible world semantics has some advantages over this: it clearly defines when goals are consistent with each other and with what is known. One can easily specify how goals change when an action a occurs, e.g. the goal to do a next and then do b becomes the goal to do b next, the goal that $\Diamond\Phi \vee \Diamond\Psi$ becomes the goal that $\Diamond\Psi$ if a makes achieving Φ impossible, etc.

To the best of our knowledge, the only set of goal change postulates that can be found in the literature is the one proposed by da Costa Pereira et al. in a series of papers on goal revision for rational agents (e.g., see [4, 5, 6]). In their framework, an agent's state S is a triple $\langle \sigma, \gamma, \mathcal{R}_D \rangle$ that consists of a belief-base σ and a desire-base γ (these are presumably achievement goals), each of which is a set of propositional formulae taken from an object language $\mathcal{L}$ containing the standard boolean connectives, and a desire adoption rule-base $\mathcal{R}_D$. The latter consists of PRS-like rules [9], which depending on the agent's current beliefs and desires, allow her to derive new desires, and are meant to serve as a justification for having a desire. Given a state S, a rule whose antecedent is entailed by the agent's current beliefs and desires is called an *active* rule. An agent's desires are updated both as a result of a new/revised belief b and of adoption of a new desire d. When the agent's beliefs are revised/updated, she removes from her desire-base any desire d for which there is no justification in the desire adoption rule-base, i.e. there is no active desire adoption rule in $\mathcal{R}_D$ that can be used to derive d. In addition, she adds the new desires that can be derived from her active desire adoption rules. Thus γ is closed under the application of rules from $\mathcal{R}_D$. When the agent adopts a new desire d, a new goal update rule with the antecedent that True is added to her rule-base, which in turns makes her add d to her desire-base. The authors then suppose that an intention/goal selection function $\mathcal{I}$ is provided, which given a belief-base and a desire-base, decides which of these desires the agent should actively pursue, i.e. intend.

Their notion of consistency of goals/desires appeals to a specification of consistency of plans for these goals. Consistency of plans is specified in terms of consistency in ordinary propositional logic, as opposed to using a proper formalization for actions and their preconditions and effects in a suitable temporal framework.

To model prioritized desires, they assume a preference relation $\succeq$ over desires in γ that is reflexive and transitive, which they extend to apply to sets of desires.

In the following, we give their postulates which constrain $\mathcal{I}$. Suppose that $\otimes$ is the desire-base γ revision operator, $\oplus$ is the desire adoption rule-base $\mathcal{R}_D$ update operator (that adds an unconditional rule to $\mathcal{R}_D$ when the agent adopts a new desire), and $S_d = \langle \sigma, \gamma \otimes d, \mathcal{R}_D \oplus d \rangle$ is the updated state resulting from the adoption of desire d in S. Then:

I_1 : For all S, $\mathcal{I}(S)$ is a feasible goal set, i.e. a consistent set of goals that are possible.

I_2 : For all S, if $\gamma' \subseteq \gamma$ is a feasible goal set, then $\mathcal{I}(S) \succeq \gamma'$, i.e. a rational agent always selects the most preferable intention set.

I_3 : If d is consistent with $\mathcal{I}(S)$, then $d \in \mathcal{I}(S_d)$.

I_4 : If d is inconsistent with $\mathcal{I}(S)$ and there is an intention i in $\mathcal{I}(S)$ that is conflicting with d and $i \succeq d$, then $\mathcal{I}(S_d) = \mathcal{I}(S)$.

I_5 : If d is inconsistent with $\mathcal{I}(S)$ and for all intentions i in $\mathcal{I}(S)$ that are conflicting with d, we have $d \succeq i$, then $d \in \mathcal{I}(S_d)$ and $i \notin \mathcal{I}(S_d)$.

As mentioned above, these postulates are based on notions of consistency of sets of desires and executability of desires that seems problematic. In our framework, we specify executability using a formal action theory (by action precondition axioms), and we interpret consistency among a set of (achievement) goals as the existence of a path starting with the current situation over which all of these goals hold. Given this interpretation, we think these postulates are in fact sound. A formal version of I_1 is shown to hold in our framework by Proposition 1 and Corollary 1. Note that I_2 seems problematic unless the ordering $\succeq$ over desires is total, which is the case for our framework. If a partial order is assumed, an agent might have several alternative sets of chosen goals, none of which is better than the others. We formalize I_3 in Proposition 3(b). Proposition 3(c) shows that I_5 is partially satisfied in our framework (we didn't prove that $i \notin \mathcal{I}(S_d)$). Finally, we believe that I_4 and I_5 are both satisfied in our framework. Proving this is left for future work.

There has been much work on agent programming languages with declarative goals where the dynamics of goals and intentions are modeled (e.g. [19, 27, 1] and the references therein). However, most of these are not based on a

formal theory of agency. To the best of our knowledge, none maintains the consistency of (chosen) goals, i.e. when adopting a plan to achieve a goal, these frameworks do not ensure that this plan is consistent with the agent's other concurrent goals/plans. For instance, the lookahead search operator Plan(P) proposed in the CAN-PLAN agent programming language [19], that searches for a complete execution of the plan P before performing it, is "local": the agent may adopt multiple search tasks, say Plan(P_1) and Plan(P_2), but the output of these Plan operators need not be consistent with each other or with the agent's other concurrent intentions/plans, as is acknowledged in [19]. Also, most of these agent programming languages do not deal with temporally extended goals, and as a result they often need to accommodate inconsistent goal-bases to allow the agent to achieve conflicting states at different time points (e.g. the default logic based framework in [26]); chosen goals are required to be consistent. In [25], the authors formalized two semantics for representing conflicting goals, using propositional and default logic; they argued that even logically consistent goals can be conflicting, e.g. when multiple goals/plans are chosen to fulfill the same (super)goal. Unlike us however, they do not address how an agent chooses the goals that she will actively pursue. In [18], the authors present a situation calculus based agent programming language where the agent executes a program while maximizing the achievement of a set of prioritized goals. However, they do not formalize goal dynamics.

One limitation of our account is that one could argue that our agent wastes resources trying to optimize her c-goals at every step. In the future, we would like to develop an account where the agent is strongly committed to her chosen goals, and where the filter override mechanism is only triggered under specific conditions. Also, it would be interesting to identify a set of postulates for goal change and examine how they differ from belief change postulates.

8. REFERENCES

[1] Bordini, R.H., Dastani, M., Dix, J., Fallah-Seghrouchni, A.E., eds.: Multi-Agent Programming: Languages, Platforms and Applications. Springer (2005)

[2] Bratman, M.E.: Intentions, Plans, and Practical Reason. Harvard Univ. Press, Cambridge, MA (1987)

[3] Cohen, P.R., Levesque, H.J.: Intention is Choice with Commitment. Artificial Intelligence **42**(2–3) (1990) 213–361

[4] da Costa Pereira, C., Tettamanzi, A.: A Belief-Desire Framework for Goal Revision. Eleventh Intl. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES) (2007) 164–171

[5] da Costa Pereira, C., Tettamanzi, A., Amgoud, L.: Goal Revision for a Rational Agent. Seventeenth European Conference on Artificial Intelligence (ECAI) (2006) 747–748

[6] da Costa Pereira, C., Tettamanzi, A.: Towards a framework for goal revision. Eighteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC), Namur, Belgium, (2006) 99–106

[7] De Giacomo, G., Lespérance, Y., Levesque, H.J.: ConGolog, a Concurrent Programming Language Based on the Situation Calculus. Artificial Intelligence **121** (2000) 109–169

[8] Gärdenfors, P. Knowledge in Flux. The MIT Press, Cambridge, MA (1988)

[9] Georgeff, M. P., Ingrand F.: Decision Making in an Embedded Reasoning System. Eleventh Intl. J. Conf. on Artificial Intelligence (IJCAI), Detroit (1989) 972–978

[10] Konolige, K., Pollack, M.E.: A Representationalist Theory of Intention. Thirteenth Intl. J. Conf. on Artificial Intelligence (IJCAI), Chambéry, France (1993) 390–395

[11] Levesque, H.J., Pirri, F., Reiter, R.: Foundations for a Calculus of Situations. Electronic Transactions of AI (ETAI) **2**(3–4) (1998) 159–178

[12] McCarthy, J., Hayes, P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. Machine Intelligence **4** (1969) 463–502

[13] Moore, R.C.: A Formal Theory of Knowledge and Action. In Hobbs, J.R., Moore, R.C., eds.: Formal Theories of the Commonsense World. Ablex (1985) 319–358

[14] Rao, A.S., Georgeff, M.P.: Modeling Rational Agents with a BDI-Architecture. Second Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR&R), San Mateo, CA (1991) 473–484

[15] Rao, A.S., Georgeff, M.P.: Asymmetry Thesis and Side-Effect Problems in Linear-Time and Branching-Time Intention Logics, Twelfth Intl. J. Conf on Artificial Intelligence (IJCAI), Sydney, Australia (1991) 498–504

[16] Reiter, R.: Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press, Cambridge, MA (2001)

[17] Sadek, M.D.: A Study in the Logic of Intention. Third Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR&R-92), Cambridge, MA (1992) 462–473

[18] Sardina, S., Shapiro, S.: Rational Action in Agent Programs with Prioritized Goals. Second Intl. J. Conf. on Autonomous Agents and Multi-Agent Sys. (AAMAS), Melbourne, Australia (2003) 417–424

[19] Sardina, S., de Silva, L., Padgham, L.: Hierarchical Planning in BDI Agent Programming Languages: A Formal Approach. Fifth Intl. J. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan (2006) 1001–1008

[20] Scherl, R., Levesque, H.: Knowledge, Action, and the Frame Problem. Artificial Intelligence **144**(1–2) (2003) 1–39

[21] Shapiro, S., Brewka, G.: Dynamic Interactions Between Goals and Beliefs. Twentieth Intl. J. Conf. on Artificial Intelligence (IJCAI), India (2007) 2625–2630

[22] Shapiro, S., Lespérance, Y., Levesque, H.J.: Goal Change in the Situation Calculus. J. of Logic and Computation **17**(5) (2007) 983–1018

[23] Shapiro, S., Lespérance, Y., Levesque, H.J.: Goals and Rational Action in the Situation Calculus - A Preliminary Report. Working Notes of the AAAI Fall Symp. on Rational Agency: Concepts, Theories, Models, and Applications, Cambridge, MA (1995) 117–122

[24] Singh, M.P.: Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications. Volume 799 of LNAI. Springer-Verlag, Germany (1994)

[25] van Riemsdijk, M.B., Dastani, M., Meyer, J.J.Ch.: Goals in Conflict : Semantic Foundations of Goals in Agent Programming. International Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS) **18**(3) (2009) 471–500

[26] van Riemsdijk, M.B., Dastani, M., Meyer, J.J.Ch.: Semantics of Declarative Goals in Agent Programming. Fourth Intl. J. Conf. on Autonomous Agents and Multiagent Sys. (AAMAS). (2005) 133–140

[27] van Riemsdijk, M.B., Dastani, M., Dignum, F., Meyer, J.J.Ch.: Dynamics of Declarative Goals in Agent Programming. Second Intl. Workshop on Declarative Agent Languages and Technologies (DALT). Volume 3476 of LNCS, NY, USA, Springer-Verlag (2004) 1–18

[28] Winikoff, M., Padgham, L., Harland, J., Thangarajah, J.: Declarative and Procedural Goals in Intelligent Agent Systems. Eighth Intl. Conf. on Principles and Knowledge Representation and Reasoning (KR&R), Toulouse, France (2002) 470–481