Handling Prioritized Goals and Subgoals in a Logical Account of Goal Change

(Extended Abstract)

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

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

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, subgoals, intention, logic of agency

1. INTRODUCTION

Most existing formal models of goals [2, 3] assume that all goals are equally important and many only deal with achievement goals. Moreover, they do not guarantee that an agent's goals will properly evolve when an action/event occurs, e.g. when the agent's beliefs/knowledge changes or a goal is adopted or dropped. Also, most of these frameworks do not model the dependencies between goals and the subgoals and plans adopted to achieve these goals – subgoals adopted to bring about a goal should be dropped when the parent goal becomes impossible, is achieved, or is dropped. Dealing with these issues is important for developing effective models of rational agency and BDI agent programming languages.

Here, we outline a formal model of prioritized goals and their dynamics that addresses 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 formalization of goal dynamics ensures that the agent strives to maximize her utility. Our model of goals supports the specification of general temporally extended goals, not just achievement goals, and also handles subgoals and their dynamics.

Our base framework for modeling goal change is the situation calculus as formalized in [4]. We model knowledge using a possible worlds account adapted to the situation calculus [5]. To support modeling temporally extended goals, we introduce a new sort of *paths*, which are essentially infinite sequences of situations.

2. OUR FORMALIZATION OF GOALS

2.1 Prioritized Goals

We formalize goals or desires with different priorities; we call these *prioritized goals* or *p-goals*. These p-goals are not required to be mutually consistent and need not be actively pursued by the agent. Using these, we then define the consistent set of *chosen goals* or intentions (*c-goals*, henceforth) that the agent is committed to.

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 if all the goals of the agent at level n are 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 G-accessible paths reflect the actions that have been performed so far. A smaller n represents higher priority. Thus in this framework, we assume that the set of p-goals are totally ordered according to priority. 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. We assume that the agent has a finite number k of initial p-goals.

Using p-goals, we next define c-goals. While p-goals or desires may be known to be impossible, an agent's c-goals or intentions must be realistic. Thus we define *realistic* p-goal accessible paths, G_R: p is G_R-accessible at level n in s if it is G-accessible at n in s and starts with a situation that is knowledge-accessible in s.

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 starts with a knowledge-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 one 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 $i-1$. We repeat this until we reach $i = k$. C-goal accessible paths G_C are the result of this intersection after all priority levels have been considered. We say that the agent has the c-goal that ϕ in s if ϕ holds over all of her G_C-accessible paths in s.

Consider the following example: we have an agent who initially has the p-goals $\Box$BeRich, $\Diamond$GetPhD, and $\Box$BeHappy, in order of priority. While all of her p-goals are individually achievable initially, her p-goal $\Diamond$GetPhD is inconsistent with her highest priority p-goal $\Box$BeRich as well as with $\Box$BeHappy, while the latter are consistent with each other. It follows that initially our agent has the c-goals that $\Box$BeRich and $\Box$BeHappy, but not $\Diamond$GetPhD.

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 because they are

Cite as: Handling Prioritized Goals and Subgoals in a Logical Account of Goal Change (Short Paper), Shakil M. Khan and Yves Lespérance, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

known to be inevitable or as a consequence of two or more active p-goals at different priority levels, we define *primary* c-goals (or PrimCGoal). We can show that initially our agent has the primary c-goals that $\Box$BeRich and $\Box$BeHappy, but not their conjunction.

2.2 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. For the latter, we introduce two actions, $adopt(\phi)$ and $drop(\phi)$. We specify the dynamics of p-goals as follows (the agent's c-goals are automatically updated when her p-goals change). Firstly, to handle the occurrence of a non-adopt/drop action a, we progress all G-accessible paths to reflect the fact that this action has occured. Any path where the next action performed is not a is eliminated from the respective G accessibility level.

Secondly, to handle adoption of a p-goal ϕ, we add ϕ to the agent's goal hierarchy. We assume that the newly adopted p-goal ϕ has the lowest priority. Thus, in addition to progressing the G-accessible paths at all levels as above, we eliminate the paths over which ϕ does not hold from the k-th G-accessibility level, and the agent acquires the p-goal that ϕ at level k.

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.

Returning to our example, assume that after action *goBankrupt* happens, the p-goal $\Box$BeRich becomes impossible. Then the agent's c-goals become $\Diamond$GetPhD. The p-goal $\Box$BeHappy becomes inactive because it is inconsistent with $\Diamond$GetPhD which has higher priority. Note that, while it might be reasonable to drop a p-goal that is in conflict with another higher priority active p-goal (e.g. $\Diamond$GetPhD which is inconsistent with $\Box$BeRich ininitially), here we keep such p-goals around. The reason for this is that the agent might later learn that the higher priority p-goal has become impossible to bring about (e.g. $\Box$BeRich after *goBankrupt* occurs), and then might want to pursue the inactive lower priority p-goal (e.g. $\Diamond$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.

2.3 Properties

We can prove the following properties. Let D be our domain theory.

- CONSISTENCY: $D \models \forall s.\ \neg\text{CGoal}(\text{False}, s)$, i.e. c-goals are consistent, and the agent cannot have both ϕ and $\neg\phi$ as c-goals in s.
- REALISM: $D \models \text{KInevitable}(\phi, s) \supset \text{CGoal}(\phi, s)$, i.e. if an agent knows that something has become inevitable, then she has this as her c-goal [2]. This is not necessarily true for PGoal/PrimCGoal.
- ADOPTION: (a) $D \models \exists n.\ \text{PGoal}(\phi, n, do(adopt(\phi), s))$, i.e. an agent acquires the p-goal at some level n that ϕ after she adopts it, and (b) $D \models \neg\text{CGoal}(\neg\phi, s) \supset \text{PrimCGoal}(\phi, do(adopt(\phi), s))$, i.e. an agent acquires the primary c-goal (and c-goal) that ϕ after she adopts it in s, if she does not have the c-goal that $\neg\phi$ in s.
- DROP: $D \models \neg\exists n.\ \text{PGoal}(\phi, n, do(drop(\phi), s))$, i.e. after dropping the p-goal that ϕ in s, an agent does not have the p-goal that ϕ. Note that this does not hold for CGoal, as ϕ could still be a consequence of two or more of her remaining primary c-goals.
- PERSISTENCE OF ACHIEVEMENT PGOALS: $D \models \text{PGoal}(\Diamond\Phi, n, s) \wedge \text{Know}(\neg\Phi(now), s) \wedge \forall\psi.\ a \neq drop(\psi) \supset \text{PGoal}(\Diamond\Phi, n, do(a, s))$, i.e. if an agent has a p-goal that $\Diamond\Phi$ in s, then she will retain it after action a has been performed in s, if she knows that Φ has not yet been achieved, and a is not a $drop$ action.

We have also proved a similar property about achievement c-goal persistence; in addition to the above conditions, it requires that the agent's higher priority c-goals remain consistent with $\Diamond\Phi$ after a has been performed.

2.4 Handling Subgoals

We also handle subgoal adoption and model 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. We handle this as follows: adopting 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. $\psi \wedge \phi$, at a lower priority than the parent goal ϕ. 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, this means that dropping a subgoal does not necessarily drop the supergoal. Note that we can represent complex procedural subgoals using Golog [4].

3. DISCUSSION AND FUTURE WORK

While our chosen goals are closed under logical consequence, primary c-goals are not, and as such do not suffer from the side-effect problem [2]. It is interesting to look at the relationship between our c-goals and Bratman's notion of intention [1]. Recall that intentions limit the agent's practical reasoning – agents do not always reconsider all available options in order to allocate their reasoning effort wisely. In contrast, we ensure that the agent's c-goals maximize her utility. There is a tradeoff between optimizing the agent's chosen set of prioritized goals and being committed to chosen goals. Our c-goals are not as persistent as Bratman's intentions – our agent might loose a c-goal, e.g. $\Box$BeHappy after *goBankrupt* happens, although she did not drop it and it did not become impossible or achieved. In this sense, our c-goals behave like intentions with an automatic filter-override mechanism [1].

There have been a few proposals [7, 6] that deal with goal change. Shapiro and Brewka [6] modify the account of goal change in [7] to model prioritized goals and their dynamics. Their account is similar to ours; but, it has some unintuitive properties: the agent's chosen goals can change from a situation to the next simply because inconsistencies between goals at the same priority level are resolved differently (this can happen because goals are only partially ordered). 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, e.g. unbounded maintenance goals. Most approaches to agent programming languages with declarative goals are not based on a formal theory of agency, and to the best of our knowledge none deals with temporally extended goals or maintain the consistency of (chosen) goals.

4. REFERENCES

[1] M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.

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

[3] A. S. Rao and M. P. Georgeff. Modeling Rational Agents with a BDI-Architecture. In *Proc. of KR&R-91*, pages 473–484, 1991.

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

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

[6] S. Shapiro and G. Brewka. Dynamic Interactions Between Goals and Beliefs. In *Proc. of IJCAI-07*, pages 2625–2630, India, 2007.

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