A unified interaction-aware goal framework

Michael Winikoff[1] and **Mehdi Dastani**[2] and **M. Birna van Riemsdijk**[3]

Abstract. Goals are central to the design and implementation of intelligent software agents. Much of the literature on goals and reasoning about goals only deals with a limited set of goal types, typically achievement goals, and sometimes maintenance goals; and much of the work on interactions between goals only deals with achievement goals. We aim at extending a previously proposed unifying framework for goals with additional richer goal types, including a combined "achieve and maintain" goal type. We propose to provide an operationalization of these new goal types, proving that the operationalization meets desired properties.

1 Introduction

A widely-accepted approach to designing and realising agents is the *cognitive* approach, where agents are modeled in terms of mental concepts such as beliefs, goals, plans and intentions. Of the various concepts that have been used for cognitive agents, a key concept is *goals*. This is because agents are (by common definition) proactive, and goals are what allow agents to be proactive. It is also noteworthy that (the existence of explicitly represented) goals is one of the clearer differences between (proactive) agents and active objects.

Goals have been extensively studied (e.g. [6, 3, 1]). But, upon examining this literature, it becomes apparent that the goal types being considered are fairly narrow, often comprising just achievement and maintenance goals, and in some work, such as work that considers interactions between goals (e.g. [2, 4]), comprising only achievement goals.

To make this discussion more concrete, consider a personal assistant agent that manages a user's calendar and tasks. One task the agent may have is booking a meeting. This would typically be modelled as an achievement goal that aims to bring about a state where all required participants have the meeting in their calendar. However, in practice, diaries change, and we want to ensure that the meeting remains in participants' diaries, and that if a key participant become unable to attend, a new time will be negotiated. This is not captured by an achievement goal. Rather, it is better modelled by a combined "achieve then maintain" goal which achieves a certain condition, and then maintains it over a certain time period.

Another task that we might want the agent to undertake is to ensure that booking travel is not done until the budget is approved. Note that budget approval may be under the control (or perhaps just influence) of the agent, i.e. the agent may have plans for (attempting to) have the budget approved. Alternatively, it may be completely outside the agent's control, in which case the agent can just wait for it to happen and then enable the travel booking process.

[1] University of Otago, New Zealand, email: michael.winikoff@otago.ac.nz
[2] Utrecht University, The Netherlands, email: mehdi@cs.uu.nl
[3] Delft University of Technology, The Netherlands, email: m.b.vanriemsdijk@tudelft.nl

We propose a unifying framework for goals that deals with a wide range of goal types (not just achievement goals) and that models goal interactions. To our knowledge, these two strands of research have not yet been brought together, which would provide a comprehensive unified goal framework. Our motivation is practical: we aim to investigate goal types and their properties in such a way that the results can be fed back into (and implemented in) real agent platforms without requiring planning capabilities to deal with linear temporal logic (LTL) formulae. This is why we propose to define specific goal types and operationalise them.

We suggest to build on previous work [5] which presented a unifying framework for goals based on viewing goals as LTL formulae that described desired progressions. This previous work captured existing goal types, whereas we propose new goal types. Furthermore, [5] did not consider goal interactions.

2 New Goal Types

We extend the taxonomy of [5] with additional goal types. Specifically, the personal assistant agent example introduced new goal types. The first, booking a meeting and then maintaining participant availability can be formalised as a goal type using an LTL pattern as follows. Let pa be short for $participantsAvailable$, ms be short for $meetingScheduled$, and mo be short for $meetingOccurs$, then we represent the goal of booking a meeting as the following LTL formula[4]: $\Diamond(ms \wedge (pa \,\mathsf{U}\, mo))$.

Considering the second goal, not booking travel until the budget has been approved, this can be formalised as[5]: $(\neg bookTravel) \,\mathsf{U}\, budgetApproved$.

Abstracting from the specific goal instances to general goal types, we have defined goal types of the form $\Diamond(\phi_1 \wedge (\phi_2 \,\mathsf{U}\, \tau))$ (achieve ϕ_1 and maintain ϕ_2 until τ), and $(\neg\phi) \,\mathsf{U}\, \tau$ (refrain from ϕ until τ). We now generalise these goal types by considering a range of ways in which a property ϕ can be required to hold over a number of states.

Consider a multiple-state goal where a (non-temporal) property ϕ is required to hold over a number of states in the trace. The taxonomy of [5] only supports a single type of multiple-state goal. However, there are a number of ways in which a goal pattern can apply to a sequence of states. It can apply: (i) to all states: $\Box\phi$; (ii) at the start of the trace: $\phi \,\mathsf{U}\, \tau$, where τ is a formula that describes the state at which ϕ is no longer required to be true; (iii) at the end of the trace: $\Diamond(\tau \wedge \Box\phi)$, where τ is a "trigger" formula that describes the state at which ϕ begins to be required to be true; or (iv) in the middle of the trace: $\Diamond(\tau \wedge (\phi \,\mathsf{U}\, \tau'))$, where τ is the starting trigger and τ' the

[4] In LTL "$\Diamond\phi$" is "eventually ϕ", i.e. ϕ is true in the current state or in some future state; "$\Box\phi$" is "always ϕ", i.e. ϕ is true in this and in all future states; and "$\phi \,\mathsf{U}\, \psi$" is "ϕ until ψ", i.e. ψ is eventually true, and in all intervening states ϕ is true.

[5] This goal would be expected to be used in conjunction with a goal to book travel, $\Diamond bookTravel$.

ending trigger; or (v) it can apply to a number of sub-sequences of states: $\Box(\tau \rightarrow (\phi \mathsf{U} \tau'))$, where τ is a trigger that describes a state at which ϕ begins to be required to hold, and τ' describes the states at which ϕ is no longer required to hold.

To summarise, we define the following LTL patterns for multiple-state goal types (where ϕ and τ are formulae in propositional logic, i.e. non-temporal): $\Box\phi$, $\phi \mathsf{U} \tau$, $\Diamond(\tau \wedge \Box\phi)$, $\Diamond(\tau \wedge (\phi \mathsf{U} \tau'))$, and $\Box(\tau \rightarrow (\phi \mathsf{U} \tau'))$.

3 Realising the new goal types

In order to realise the new goal types in an operational setting, we assume that an agent configuration consists of a belief base, consisting of propositional formula, and two goal bases. The first goal base, called the *temporal goal base*, contains goals specified by temporal LTL formulae as discussed in previous section. The second goal base, called the *basic goal base*, consists of achieve goals of the form $A(\phi)$ (read as: ϕ should be achieved) and maintain goals of the form $M(\phi, \tau)$ (read as: ϕ should be maintained until τ) and $M(\phi, \bot)$ (read as: ϕ should be maintained forever). The formulae ϕ and τ are assumed to be propositional (non-temporal) formulae.

The operationalization of temporal goal types can then be defined in terms of operations on these bases. In this paper, we assume that the achieve and maintain goals have a correct operationalization. In particular, for the achieve goal we assume that if $A(\phi)$ is in the basic goal base, then the agent belief base will eventually entail ϕ, and for the maintain goals we assume that if $M(\phi, \tau)$ is in the basic goal base, then the agent belief base entails ϕ until τ is entailed by the belief base. Clearly, these assumptions are somewhat ideal, in that they assume that there is not any interference between goals that might prevent certain goals from being realisable. However, for the purposes of our work we want to show that the framework proposed realises goals correctly, if it is possible to do so. It is also worth noting that the assumption for $M(\phi, \tau)$ requires that ϕ hold immediately. However, other (weaker) assumptions are possible, for instance, allowing the belief base not to entail ϕ for a short while.

The operationalization of temporal goal types is accomplished by operational semantics, which indicate possible transitions between agent configurations due to temporal goal processing. These operational semantics defines how the agent pursues complex temporal goals in terms of the pursuit of basic achievement and maintenance goals. How the agent pursues basic achievement and maintenance goals is not defined here, and can be found elsewhere (e.g. [5]). We make assumptions about the pursuit of achievement and maintenance goals being done correctly, and then show that, given these assumptions, the semantics that we define correctly achieve complex temporal goals.

We define our operational semantics in terms of two functions, $\mathsf{i}(\chi, \sigma)$ ("i" is short for "implement"), and $\mathsf{r}(\chi, \sigma)$ ("r" is short for "residual"). Both functions take a temporal LTL formula, χ, and the current beliefs of the agent, σ. The first function, i, returns a set (either singleton or empty) containing a basic goal (of the form $A(\phi)$ or $M(\phi, \tau)$), whereas r returns either $\emptyset$ or $\{\chi\}$. For example, $\mathsf{i}(\Diamond\phi, \sigma)$ is $\{A(\phi)\}$ if $\sigma \not\models \phi$, i.e. if the agent does not already believe that ϕ holds, then the goal $\Diamond\phi$ is implemented by replacing it with the basic goal $A(\phi)$. The two functions are defined appropriately to realise all of the goal types defined in the previous section. The semantics for the agent simply apply these functions to its temporal goal base to yield new basic goals, and then uses existing transition rules to realise the basic goals.

4 Adding Interaction-Awareness

When pursuing its goals, a rational agent should avoid its goals interfering with each other. For example, when a rational personal assistant agent has scheduled a day to be spent visiting a remote site, it should not schedule meetings on that day that require returning to the office. Ideally, a rational agent should also exploit synergies between its goals. For example, if two meetings are required at a remote site, it should schedule them next to each other.

The question is how to extend the generic goal framework with facilities for ensuring goal consistency and exploiting synergies. In the remainder of this section we briefly outline our approach for extending the LTL-based goal framework to provide these facilities.

Interactions between goals can be around resources or conditions. A given goal may have certain requirements such as: resources that are needed (for example, a room might be needed for a meeting); or certain conditions that might need to hold either before a goal can be pursued (for example, the departmental budget might need to be approved before any significant purchases can be made); or certain conditions might need to hold *during* the pursuit of a goal (termed "in-conditions"; for example, a certain room may not be used during a meeting). We model this by attaching to each goal a requirement set R. We then define various conditions that capture different forms of goal interaction. For example, goal G_1 interferes with the in-conditions of goal G_2 if both goals are active, c is an in-condition of G_2, and the effects of G_1 imply the negation of c.

In responding to a situation where there is interference between two goals we can do a number of things. Firstly, we can influence the selection of which goal to pursue so that the interfering goals are not pursued at the same time. This can be done by suspending a goal that is interfering with another goal. In extreme cases it may be necessary to drop a conflicting goal completely. In some situations conflict can be resolved by *adding* a goal, for example, if the conflict is that G_1 breaks a condition c which G_2 needs as a pre-condition, then an additional goal to restore c may be adopted in order to resolve the conflict.

REFERENCES

[1] Lars Braubach, Alexander Pokahr, Daniel Moldt, and Winfried Lamersdorf, 'Goal representation for BDI agent systems', in *Programming multiagent systems, second international workshop (ProMAS'04)*, volume 3346 of *LNAI*, 44–65, Springer, Berlin, (2005).

[2] Bradley J. Clement and Edmund H. Durfee, 'Theory for coordinating concurrent hierarchical planning agents using summary information', in *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 495–502, (1999).

[3] Mehdi Dastani, M. Birna van Riemsdijk, John-Jules Ch. Meyer, 'Goal Types in Agent Programming', in *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06)*, Italy, (2006).

[4] J. Thangarajah, M. Winikoff, L. Padgham, and K. Fischer, 'Avoiding resource conflicts in intelligent agents', in *Proceedings of the 15th European Conference on Artifical Intelligence (ECAI)*, ed., F. van Harmelen, Lyon, France, (2002).

[5] M. Birna van Riemsdijk, Mehdi Dastani, and Michael Winikoff, 'Goals in agent systems: A unifying framework', in *Autonomous Agents and Multi-Agent Systems (AAMAS)*, eds., Padgham, Parkes, Müller, and Parsons, pp. 713–720. IFAAMAS, (2008).

[6] Michael Winikoff, Lin Padgham, James Harland, and John Thangarajah, 'Declarative & procedural goals in intelligent agent systems', in *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Toulouse, France, (2002).