Towards Reasoning with Partial Goal Satisfaction in Intelligent Agents

M. Birna van Riemsdijk[1] and Neil Yorke-Smith[2,3]

[1] EEMCS, Delft University of Technology, Delft, The Netherlands.
m.b.vanriemsdijk@tudelft.nl
[2] SRI International, Menlo Park, CA, USA.
[3] American University of Beirut, Lebanon. nysmith@aub.edu.lb

Abstract. A model of agency that supposes goals are either achieved fully or not achieved at all can be a poor approximation of scenarios arising from the real world. In real domains of application, goals are achieved over time. At any point, a goal has reached a certain level of satisfaction, from nothing to full (completely achieved). This paper presents a framework for representing partial goal satisfaction in an intelligent agent. The richer representation enables agents to reason about *partial* satisfaction of the goals they are pursuing or that they are considering. In contrast to prior work on partial satisfaction in the agents literature which investigates partiality from a logical perspective, we propose a higher-level framework based on metric functions that represent, among other things, the progress that has been made towards achieving a goal. We present an example to illustrate the kinds of reasoning enabled on the basis of our framework for partial goal satisfaction.

Categories and subject descriptors: I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents*

General terms: Design; Theory

Keywords: goal reasoning, partial satisfaction, agent programming

1 Introduction and Motivation

This work starts from the observation that existing cognitive agent programming frameworks (e.g., [30, 4, 10]), i.e., programming frameworks in which agents are endowed with high-level mental attitudes such as beliefs and goals, take a 'boolean' perspective on goals: unless achieved completely, the agents have failed to achieve them. Following Zhou et al. [32], we argue that many scenarios would benefit from a more flexible framework in which agents can reason about *partial goal satisfaction*. As others have recognized, it is important that agents can be programmed with this reasoning ability, because often it is not possible for an agent to achieve a goal completely, in the context of all its commitments situated in the resource-bounded real world. A notion of partiality allows to express that only part of the goal is achieved, and it facilitates, among other possibilities, changing goals such that only a part has to be achieved.

While prior work proposes a logic-based characterization of partiality, in this paper we aim for a general framework for partial goal satisfaction that also allows quantitative notions of partiality. In particular, we propose a framework based on metric functions that represent, among other things, the progress that has been made towards achieving a goal. Agents rescuing civilians from a dangerous area, for example, may have cleared none, some, or all of the area. Progress may be expressed in terms of different kinds of metrics, such as utility, or in terms of a logical characterization. This richer representation enables an agent or group of agents to reason about partial satisfaction of the goals they are pursuing or that they are considering. The more sophisticated behaviour that can result not only reflects the behaviour expected in real scenarios, but can enable a greater total level of goal achievement. For example, an agent might realize that it cannot completely clear a sub-area and inform teammates of the situation; in turn, they adjust their behaviour appropriately, e.g., by coming to assist.

This paper aims to further establish partial goal satisfaction as an important topic of research, and to provide a step towards a metric-based approach that also allows for quantitative notions of partial achievement. This represents a step towards enhancing the capabilities of cognitive agent programming frameworks. Accordingly, we develop an abstract framework for partial goal satisfaction and discuss the concept using an example scenario. We identify *progress appraisal* (the capability of an agent to assess how far along it is in achieving a goal [6]) and *goal adaptation* (the modification of a goal [17, 23, 32]) as the basic types of reasoning that the framework should support. We illustrate how reasoning using partial goal satisfaction can be embedded into a concrete computational framework for the metrics.

The foremost challenge of the work in this paper is conceptual: we identify the main ingredients we believe should be part of an abstract framework for partial goal satisfaction. Through this, we lay foundations for future work, which will address the important technical challenges that have to be faced to concretize the framework and render it suitable for programming a cognitive agent.

2 Background and Related Work

Goal representation. In cognitive agent programming, the concept of a goal has received increasing attention in the past years. Different goal types have been distinguished (see, e.g., [29, 2] for a discussion), including achievement goals and maintenance goals. The former, which have received the most attention in the literature, form the focus of this paper. In the literature, goals have been viewed as declarative and thus as properties of states (i.e., goals-to-be); we take the same perspective [30, 4, 10].

Achievement goals in logic-based cognitive agent programming languages are often represented as a logical formula, expressing a property of the state of the multi-agent system that the agent should try to achieve [30, 31, 4, 29, 10]. The

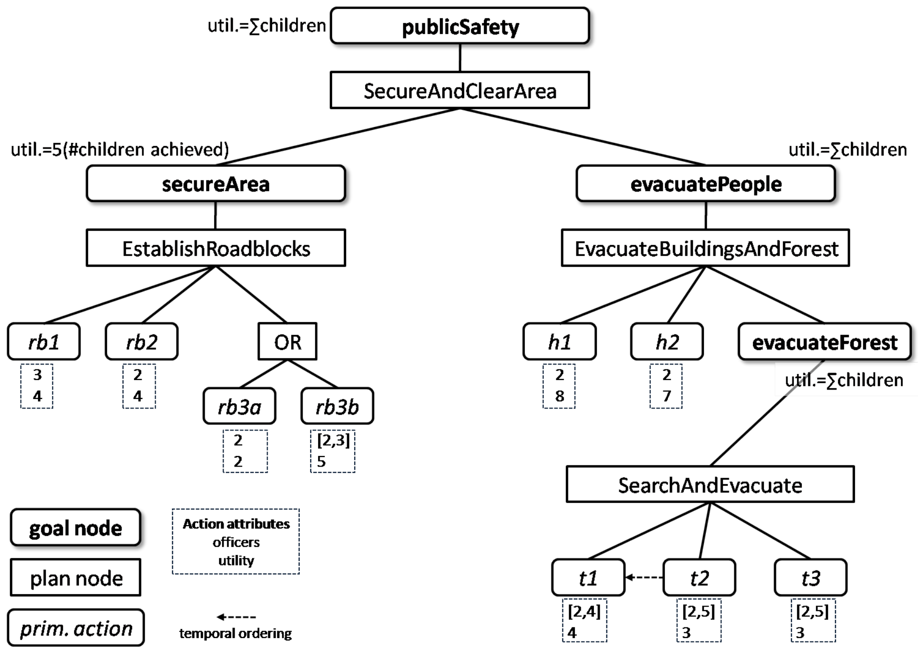

Fig. 1. Goal-plan tree for the scenario.

agent considers the goal to be achieved, if it believes a state has been reached in which the formula is satisfied, according to the semantics of the logic employed. This logic-based approach can induce a binary way of thinking about goals, in which the goal is either achieved or not achieved. While we do not reject that point of view, we suggest in this paper that a framework in which levels of goal satisfaction can be represented enables several useful kinds of reasoning.

Partial achievement. The concept of partial achievement of a goal appears in limited extents in the literature. Whereas goals in agent frameworks and programming languages are not customarily defined to allow for partial satisfaction, philosophically, Holton argues for the existence of "partial intentions" [13], a concept spanning both desires and goals.

In the foundational work of Rao and Georgeff [20] an intention (goal) is dropped if it is achieved, not desired, or now believed by the agent to be impossible. Singh [25] drops an goal if another more important task arises. In these and works that followed them, goal achievement remains a boolean concept.

Haddawy and Hanks made an early study [9], in which a function from propositions to a real number represents the degree of satisfaction of a goal. Indeed, various authors have associated goals with a utility or preference, in the agents literature (e.g., [14, 11], among others) and in the AI planning literature (e.g., [5]), although usually for the purpose of deciding which goals to prioritize or which subset to pursue, or which plan or action to select.

Zhou and Chen adopt instead a logical approach, defining a semantics for partial implication of desirable propositions from a symbolic point of view [31]. Zhou et al. [32] investigate partial goal satisfaction on the basis of this logical semantics, viewing a goal as achieved when a (possibly disjunctive) proposition is achieved according to the logic. They examine in particular application of different notions of partial implication to goal modification in the context of belief change. Although recognizing its value, we do not approach partial satisfaction viewing goals as logical formulas to be achieved. We discuss the relationship between the approaches later.

While van der Hoek et al. [28] explore a related concept, in their logical analysis of BDI intention revision, we aim for more a fine-grained and broader concept. Morley et al. [18] investigate dynamic computation of resource estimates as a partially-complete goal is executed. Again, the representation of a generic concept of partial achievement is not the focus of their work.

Partial plans and goal/plan adaptation. There is a fair amount of work on reasoning with partial *plans*, for instance in plan formation or negotiation (e.g., [17, 8, 15]), as well as in the AI planning literature (e.g., [26]). In the area of multi-agent planning and negotiation, researchers have examined inter-agent communication (e.g., about problems in goal achievement). Kamar et al., for instance, investigate helpful assistance of teammates in pursuit of a plan that could be partially complete [15]. Goal adaptation has received less attention than the concept of goal or plan selection (e.g., [17]), or plan adaptation, the benefits of which are well established [19].

3 Example Scenario

We illustrate by means of an extended example the benefits that a framework for partial goal satisfaction may bring. The scenario is from the domain of crisis management. An accident has occurred in a chemical plant and hazardous chemicals have leaked into the area. The emergency response team must prevent anyone from entering the vicinity of the plant, and evacuate those who are currently in the area. A team of agents will execute a joint plan according to their training. Securing the area is done by setting up road blocks on the three main roads leading to the plant; the third road block can be installed in one of two different places. The two houses within a 3 km radius of the plant must be evacuated. The forest within the range of the chemical leak must be searched and any people brought to safety.

Fig. 1 depicts a *goal-plan tree* (GPT) [27, 3, 18] for the emergency response team in the scenario. A goal-plan tree consists of alternating layers of goal nodes and plan nodes. Goals are depicted in rounded boxes, and plans in square boxes. Goals descending from a plan node are conjunctive: all must be achieved for the plan to be successful. An OR node indicates disjunctive subgoals: achievement of any one renders the plan successful. Thus, the plan *EstablishRoadblocks* is successful when goals *rb1* and *rb2* and at least one of *rb3a* and *rb3b* are achieved.

Primitive actions (leaf goal nodes) are depicted in italicized rounded boxes. The numerical attributes on leaf nodes will be discussed later.

This scenario would benefit from agents being able to reason with partial goal satisfaction. A basic type of reasoning is *progress appraisal* [6]. Progress appraisal is the capability of an agent to assess how far along it is in achieving a goal, i.e., which part of a goal it has already achieved. In the scenario, for example, it may be important for the commander to keep headquarters up-to-date on her progress in setting up the road blocks.

Another, more advanced, type of reasoning with partial goal satisfaction is *goal negotiation*, which has been identified as a key challenge for agents research [16]. Assume, for example, that the team does not have enough members to secure the area and evacuate the forest. The commander may engage in goal negotiation with headquarters, to try to adapt the *publicSafety* goal so only the part that is achievable for the team will have to be pursued. Note that the ability to do *goal adaptation* is thus necessary in order to engage in goal negotiation. The commander suggests to set up only road blocks 2 and 3. However, neglecting road block 1 is not an option according to headquarters, since people may (re-)enter the area, which would lead to a hazardous situation and further evacuation duties. The latter decision is based on an analysis of the importance of achieving the various subgoals. The commander agrees with headquarters that another team will be sent to set up road block 1. Both goal negotiation and adaptation thus require agents to reason about the parts of which a goal is composed.

These kinds of reasoning may occur not only before a goal is adopted, but also during pursuit of a goal. For example, the commander may notice that searching the forest is taking more time than expected, and the team will not be able to search the entire forest before darkness sets in. Rather than abandoning *evacuateForest* entirely because the goal cannot be achieved completely, the team can perform an inferior search of it and achieve it only partially. A decision of whether this is acceptable, or whether it would be better to abandon the forest altogether, depends on an analysis the gains made by achieving the goal only partially—which in this case might be substantial since any person brought to safety is an accomplishment.

This paper provides a high-level framework for partial goal satisfaction that allows a quantitative instantiation, aimed at enabling the kinds of reasoning such as discussed above. After introducing the framework, we mention several other kinds of reasoning that benefit from such a framework for partial satisfaction.

4 Partial Goal Satisfaction and Progress Appraisal

At the heart of conceptualizing partial goal satisfaction is identifying how to define partiality. For this, it is essential to define when a goal is *achieved* (satisfied completely): we cannot define partiality without knowing what complete satisfaction means. In pursuit of our interest in a quantitative framework, moreover, one needs a metric in terms of which (complete) satisfaction is expressed. This metric will be endowed with a partial ordering, to allow an agent to determine whether a goal is getting closer to completion. We call such a metric the *progress metric* of a goal, and denote it as a set A with partial order $\leq$.[4] A goal then specifies a minimum value $a_{min} \in A$ (called the *completion value*) that should be reached in order to consider the goal to have been completely satisfied. For example, the progress metric for the goal *evacuateForest* might be defined in terms of time, where complete satisfaction is achieved when the forest has been searched for two hours (until it gets dark); or the metric may be defined of a (boolean) proposition such as *isSearched(forest)*; or it may be defined in terms of the number of subgoals achieved (e.g., searching tracks 1, 2, and 3), where complete satisfaction means that all tracks have been searched, etc.

One may consider a wide range of domain-independent metrics, such as TIME, UTILITY, NUMBER OF SUBGOALS, besides domain-dependent metrics such as NUMBER OF ROAD BLOCKS, NUMBER OF PEOPLE BROUGHT TO SAFETY. Besides the metric chosen as the progress metric, the agent (or designer) might have interest in others: e.g., progress may be defined in terms of tracks searched, but time taken could be an additional relevant factor in the team's decisions.

As seen earlier, a fundamental reasoning concerning partial goal satisfaction is *progress appraisal* [6]. An agent should thus be able to determine in a given situation where it is with respect to a progress metric $(A, \leq)$. For example, if TIME is the metric, the agent needs to be able to determine how long it has spent so far. In the case of TIME, the computation from the current state to the time spent is relatively direct. The computation may be more involved for other metrics. In the case of UTILITY, for example, more computation might be needed to determine the current appraised value of utility in terms of other, measurable quantities (i.e., other metrics besides the progress metric). However, in all cases, an agent should be able to determine, given its beliefs about current state, at least an estimation of the value of the progress metric for a goal.

Formally, for a goal with progress metric $(A, \leq)$, we require an agent to have a *progress appraisal function* $\phi : S \to A$, where S is the set of states (i.e., world state and multi-agent system state). In addition, in order to allow determination of whether the completion value $a_{min} \in A$ is reachable given the current state, we normally require the agent to have a *progress upper bound function* $\hat{\phi} : S \times M \to A$ that takes a state $s \in S$, and the *means* $m \in M$ that will be used for pursuing the goal, and yields (an estimation of) the maximum value in A reachable from state s with means m. The upper bound will enable reasoning about the achievability of a goal.

In the abstract framework, we do not further detail the content of the set of possible means M. The content of M will depend on the domain and the concrete agent (programming) framework that is used. Typically, we envisage that M will contain a description of plans and/or resources that can be used to pursue the goal. Sect. 6 contains an example in which we use the goal-plan tree.

[4] Combinations of metrics might be considered, but for simplicity, here we assume quantities are defined in terms of a single metric.

The functions ϕ and $\hat{\phi}$ now allow us to define a *goal template*. The intuition is that each type of goal, such as *secureArea* or *evacuateForest*, has an associated template. On the basis of a goal template, goal instances can be created.

Definition 1 (goal template). *Consider a multi-agent system (MAS) for which the set of possible states is defined as S. Let A be a nonempty set with a partial ordering $\leq$ (the* progress metric*), and let M be a set representing means that can be used for achieving a goal. A* goal template *T is then defined as a tuple $\langle A, M, \phi : S \to A, \hat{\phi} : S \times M \to A \rangle$, where ϕ is the* progress appraisal function*, and $\hat{\phi}$ is the* progress upper bound function*.*

This notion of goal template may be simplified to consist of only A and ϕ, if $\hat{\phi}$ cannot be provided in a certain case, i.e., when no sensible upper bound can be specified for a goal. Alternatively, it may be extended in various ways. First, the goal template itself may be parameterized to account for variants of the template. For example, depending on the area that has to be secured, the number of road blocks that have to be set up will differ, and this may influence the definition of ϕ and $\hat{\phi}$. Second, one may want to define a goal template for a single goal based on different progress metrics, allowing the agent to choose a progress metric depending on circumstances. We can capture this most simply by having two separate goal templates. Formally relating these templates (for instance by making them siblings in a hierarchy of goal types) is an extension of our basic framework. For reasons of simplicity and space, we leave the pursuit of these extensions for future work.

As noted, in order to simplify definition and computation of ϕ and $\hat{\phi}$, these functions may yield *estimated* values for progress appraisal and the upper bound. In environments that are not fully observable or that are open or dynamic, the agent may not be able to compute precisely the functions. However, an agent must be mindful of the potential adverse effects of estimation. In over-estimation of $\hat{\phi}$ or under-estimation of ϕ, the agent would try to achieve a goal even though it may be impossible to fully satisfy it, or it is already completely satisfied. On the other hand, in under-estimation of $\hat{\phi}$ or over-estimation of ϕ the agent would stop too soon. Thus, while ϕ and $\hat{\phi}$ may yield estimated values, intuitively the agent should estimate the progress upper bound in a state $s \in S$ to be at least the current progress in that state. We call this *coherency* of a goal template, and formally define it as $\forall s \in S, m \in M : \hat{\phi}(s, m) \geq \phi(s)$.

To illustrate Def. 1, consider the goal *secureArea* of the example scenario. In the scenario, the main resource (leaving aside time) is the number of police officers $P = \{0, \ldots, 10\}$. We base the progress metric for the goal on the NUMBER OF SUBGOALS ACHIEVED.

Example 1. The goal template for *secureArea* is: $T_{sa} = \langle \mathbb{R}, P, \phi_{sa}, \hat{\phi}_{sa} \rangle$. Thus, the progress metric is $A = \mathbb{R}$ with its standard $\leq$ ordering. Arbitrarily, we define $\phi_{sa}(s)$ to be 20 if all subgoals have been fully achieved in s (assuming the agent can determine this in each state s), which means that road blocks have been set up and at least one police officer guards each road block, 10 if all road blocks have been set up but not all of them have at least one officer, and 0 otherwise. Let the means include p, the number of officers allocated. We define $\hat{\phi}_{sa}(s, p)$ to be 20 iff the plan *EstablishRoadblocks* can be executed in s and it is executed with at least 6 police officers, i.e., $p \geq 6$, 10 if $1 \leq p < 6$ and the plan can be executed successfully, and 0 otherwise. Computation of the upper bound thus requires determining whether *EstablishRoadblocks* can be executed successfully. This may be done by checking simply the precondition of the plan, or by performing planning or lookahead (compare [3, 12]).

A goal template specifies the progress appraisal and progress upper bound functions. As already addressed above, we need to specify the *completion value* for a goal to specify when it is completely satisfied. In addition, the agent should determine the means that will be allocated for pursuing the goal. The completion value and means together form a goal instance.

Definition 2 (goal instance). *Let* $T = \langle A, M, \phi : S \rightarrow A, \hat{\phi} : S \times M \rightarrow A \rangle$ *be a goal template. A* goal instance *of* T *is specified as* $(a_{min}, m) : T$*, where* $a_{min} \in A$ *is the completion value, and* $m \in M$ *specifies the means that will be used for achieving the instance.*

Example 2. In the scenario, one goal instance of the goal template T_{sa} for *secureArea* is $g_{sa} = (20, \{0, \ldots, 6\}) : T_{sa}$, expressing that the commander would like to achieve a progress metric value of 20 with no more than six police officers.

Achievement. Using this notion of goal instance, we can easily define when a goal is *achieved* (completely satisfied) in a certain state $s \in S$, namely, when the appraised value of the progress metric in s is at least the completion value.

We will assume that each progress metric $(A, \leq)$ has a (totally ordered) bottom element $\perp_a \in A$ for which $\forall a \in A$ with $a \neq \perp_a$, we have $\perp_a < a$. The bottom element represents a 'zero' achievement level. When a goal instance g is created, it may start partially completed, i.e., $\phi(s) > \perp_A$ where s is the state in which g is created. For example, the road block on road 1 may already be in place, when an instance of *secureArea* is created, because the road was closed for construction.

We can now formally define goal achievement. In addition, we use the progress upper bound function and the means of the goal instance to define a kind of *goal consistency*. In logic-based frameworks for goals [30, 31, 10], an inconsistent goal is not reachable by definition. Achievement in our framework for partial goal satisfaction is similar. We say that a goal instance is *achieved* in a state s if the maximum attainable value of the progress metric from s, given the means of the goal instance, is at least the completion value.

Definition 3 (goal achievement and achievability). *Let* T *be a goal template, let* $(a_{min}, m) : T$ *be an instance of* T*, and let* $s \in S$ *be the current state. The goal instance* $(a_{min}, m) : T$ *is* completely unachieved *iff* $\phi(s) = \perp_A$*,* (completely) achieved *(or* satisfied*) iff* $\phi(s) \geq a_{min}$*, and* partially achieved *otherwise, i.e., iff* $\perp_A < \phi(s) < a_{min}$*. The goal instance is* achievable w.r.t. m *(or simply* achievable*, where the context is clear) iff* $\hat{\phi}(s, m) \geq a_{min}$*.*

For example, the goal instance g_{sa} of Example 2 above is achieved if all road blocks have been set up and each remains guarded by at least one police officer (since in that case the achieved ϕ_{sa} value is 20). It is achievable in any state $s \in S$ since six police officers are allocated for achieving the instance, whence the progress upper bound is 20, equalling the completion value. If less than six officers were allocated, the goal instance would not be achievable since then the agent could maximally attain a ϕ_{sa} value of 10.

4.1 Binary Goal Achievement

We now discuss how our framework relates to logic-based frameworks for (achievement) goals. In the latter, as noted in Sect. 2, the success condition of a goal is usually defined as a logical formula s, which is achieved in a state $s \in S$ if the agent believes s to hold in that state. We show how our definition for partial goal achievement can be instantiated such that it yields the usual binary definition of goal. We abstract from means M.

Definition 4 (binary goal instance). *Let ψ be a logical formula, for which the truth value can be determined in the current MAS state s (denoted as $s \models \psi$ iff ψ is entailed). Let $A = \{\text{false}, \text{true}\}$ with* true $>$ false*. Let $M = \{\epsilon\}$ where ϵ is a dummy element. Let $\phi(s) =$* true *if $s \models \psi$ and* false *otherwise, and let $\hat{\phi}(s, \epsilon) =$* true *if $\psi \not\models \bot$ and* false *otherwise. Let $T_{bin(\psi)} = \langle A, M, \phi, \hat{\phi} \rangle$. Then we define a* binary goal instance $\psi = (\text{true}, \epsilon) : T_{bin(\psi)}$.

Proposition 1 (correspondence). *The instantiation of the partial goal framework as specified in Def. 4, corresponds to the binary definition of goal (Sect. 2) with respect to achievement and consistency.*

Proof. We have to show that achievement and consistency hold in the binary definition of goal, iff achievement and achievability hold in the instantiated partial definition of goal. The goal ψ is achieved in the partial case in some state s iff $\phi(s) \geq a_{min}$, i.e., iff $\phi(s) \geq$ true, i.e., iff $\phi(s) =$ true, i.e., if $s \models \psi$. This is exactly the definition of achievement in the binary case. The goal ψ is achievable in the partial case iff $\hat{\phi}(s, \epsilon) \geq a_{min}$, i.e., iff $\hat{\phi}(s, \epsilon) =$ true, which is precisely the case iff ψ is consistent. □

We envisage an instantiation of our framework with the logic-based characterization of partiality of Zhou et al. [32], where in particular the ordering relation of the progress metric will have to be defined. That is, consider a semantics of partial implication and an alphabet of atoms. Intuitively, we must specify a metric on a set such as propositions over the alphabet, that gives rise to a partial order of the propositions w.r.t. the semantics of implication. Making this instantiation precise will be future research. Indeed, the role of partial implication in connection with subgoals and plans—which we account for in our framework through the computation of metrics in the GPT—has already been noted as a research topic [32].

The instantiation of our framework with a binary goal definition emphasizes that progress metrics need not be numeric. However, if ϕ_g is numeric, the agent can compute how far it is in achieving a goal as a ratio with the completion value. That is, if T is a goal template with progress appraisal function $\phi : S \rightarrow A$ and $g = (a_{min}, m) : T$ is a goal instance of T, and if quotients in A are defined (e.g., if $A = \mathbb{R}$), then a measure of progress of goal instance g when the agent is in state s is the ratio $\frac{\phi(s)}{a_{min}}$. This metric of % COMPLETE corresponds to the intuitive notion of progress as the percentage of the completion value attained.

5 Goal Adaptation

The previous section outlined an abstract framework for partial goal satisfaction. We have taken progress appraisal as the most basic form of reasoning that such a framework should support. In the motivating scenario, we argued that the framework should support more advanced kinds of reasoning, such as goal negotiation. In this section, we highlight a type of reasoning that we suggest underlies many of these more advanced kinds of reasoning, namely *reasoning about goal adaptation*. Given a goal instance $g = (a_{min}, m) : T$ where T is a goal template, we define *goal adaptation* as modifying a_{min} or m (or both). Note that modifying the plan for g is included in the scope of modifying m.

The reasoning question is how to determine which goals to adapt and how to adapt them. While this is a question that we cannot fully answer here, we analyze the kinds of adaptation and possible reasons for adapting. One important factor that may influence the decision on how, and particularly when, to adapt is the evolution of the agent's beliefs. This aspect is a focus of prior works [22, 32]. Another important factor is the consideration of a *cost/benefit analysis*. We develop our basic framework to support this kind of reasoning.

5.1 Reasons for and Uses of Adaptation

We begin by distinguishing *internal* and *external* reasons for goal adaptation. By internal reasons for we mean those that arise from issues with respect to the goal itself, while external reasons are those that arise from other factors.

More specifically, we see a lack of achievability as a main internal reason for goal adaptation. If a goal instance g is not achievable, it means that its completion value cannot be attained from the current state with the means that are currently allocated. The options without a concept of partial satisfaction are to drop/abort g, to attempt a different plan for g (if possible), to suspend g until it becomes achievable (for example, waiting for more officers to arrive), or to abort or suspend another goal in favour of g. In our framework, the goal instance can be *adapted* to make it achievable by lowering the completion value, which we call *goal weakening*, as well as by the alternative of choosing different means that allows the achievement of the current completion value, e.g., by investing additional resources. Depending on the circumstances, the latter may not always be possible. For example, if the goal is to evacuate people from their houses but it is physically not possible to get to these houses, e.g., because of flooding, it does not matter whether the officers devote more time or personnel.

Several external reasons may lead to goal adaptation. First, a goal instance g may in itself be achievable, but (collective) unachievability of other goal instances may be a reason for adapting g. That is, in practice an agent has only limited resources and it has to choose how it will invest them to achieve a set of current and future goal instances [1, 27]. For example, the agent may decide that another goal instance is more important and needs resources, leading to adaptation of the means of g. In our framework, goal adaptation provides the agent with the option of *partially* suspending, replanning, or abandoning goals. Moreover, progress appraisal helps the agent determine which goals to adapt. For example, it does not seem sensible to drop a goal instance that has a plan that is almost completed and that yields zero utility unless it is completely satisfied.

There are further external reasons for goal adaptation. Second, a particular case is consideration of a new candidate goal instance g': the question of *goal adoption*. Partial satisfaction allows an agent to consider adapting an existing goal instance, or adopting the new instance g' in a weakened form. Third, an agent might be requested by another agent to increase the completion value of a goal instance, which we call *goal strengthening*. For example, the team leader may decide that more time should be spent searching the forest.

Together, progress appraisal and goal adaptation form a basis for higher-level reasoning tasks. We have already discussed goal negotiation (Sect. 3), goal adoption, and avoiding and resolving goal achievement inconsistencies. We now briefly discuss several other kinds of reasoning. First, in order to coordinate their actions, agents should *communicate* about how far they are in achieving certain goals [7, 17, 15]. Progress appraisal provides a principled approach. Second, an agent might realize it cannot achieve a goal completely. Allowing itself to weaken a goal, it can *delegate* part of the goal to other agents. Similarly, delegation may be another option for an agent finding it has achievement difficulties. Related, third, is *reasoning about other agents* and their ability to complete tasks. For example, one agent realizing that another agent is unlikely to fully complete its task(s), irrespective of whether the other agent has acknowledged this.

5.2 Cost/Benefit Analysis

When deciding which goals to adapt and how, we suggest that a cost/benefit analysis can be an important consideration (see also, e.g., [1, 21]). We have already noted the example of an agent pursuing a goal that yields zero utility unless completely satisfied, for which only a small additional amount of effort is required. On the other hand, if an agent has obtained much utility from a goal instance g, compared to that expected when the progress metric of g reaches the completion value, and if much more effort would have to be invested to fully achieve g, it may be sensible to stop pursuit of the goal if resources are needed elsewhere. These kinds of cost/benefit analyses to obtain an optimal division of resources over goals essentially form an optimization problem. While it is beyond the scope of this paper to investigate how optimization techniques can be applied in this context, we do analyze how our framework supports it.

In order to weight up costs and benefits, one needs to know how much it would cost to achieve a certain benefit. The benefit obtained through progress on a goal can be derived in our framework by means of a UTILITY metric $u_g : S \to U$, where U is a set.[4] Note that the progress metric of a goal template might be defined in terms of u_g (as is the case for *publicSafety*), in which case $\phi_g \equiv u_g$ and $A \equiv U$; or u_g might be a different metric (as in the case of *secureArea*).

The incremental benefit obtained when achieving a goal completely is the difference between the benefit at a state s^* for which $\phi(s^*) \geq a_{min}$ (i.e., upon completion of the goal) and the benefit in the current state s_{now}. That is, for a goal instance $(a_{min}, m) : T$, the incremental benefit is $\Delta u = u(s^*) - u(s_{now})$. Note that Δu can only be calculated in this way if differences are defined on U, which will be the case if U is numeric.

The cost associated with obtaining a_{min} can be computed by a COST function $\kappa : S \times M \times S \to C$, where C is a set and $\kappa(s, m, s') = c$ implies that the cost of going from state s with means m to state s' is estimated to be c. Then we can calculate the estimated minimal incremental cost to move from the current state s_{now} to a completion state s^* with means m as $\min_{s':\phi(s') \geq a_{min}} \kappa(s_{now}, m, s')$.

Supposing that the set that measures benefit, U, and the set that measures cost, C, are mutually comparable—for instance, if both are subsets of $\mathbb{R}$—then the estimates for utility achieved so far, utility expected upon completion, cost so far, and cost to completion can be compared.

6 Towards an Embedding within a Goal Framework

In this section, we illustrate how our metric-based framework for partial goal satisfaction can be applied to a concrete goal representation framework, namely the GPT as introduced earlier. This is a step towards rendering the capabilities within a cognitive agent programming framework. An attraction of the GPT is its representation of goals, subgoals, and plans—which is pertinent for reasoning about the means and the progress in execution of a goal—combined with the annotation of and aggregation of quantities on the tree nodes—which we will use for computation of metrics. Fig. 1 depicted a goal-plan tree for the evacuation scenario. The goal and action nodes correspond to goal instances in our framework; the tree structure gives the plan aspect of their means.

For the reasons just given, we posit that the concept of partially satisfied goals fits naturally into this kind of representation framework for goals. We augment annotations of tree nodes to include metrics about goal (and, where relevant, plan) satisfaction. In the simplest case, this comprises annotating each goal node with values from its progress metric A, as we will explain. The % COMPLETE metric allows normalization of the values.

Progress appraisal. Inference over the tree structure computes and updates metrics by propagation upwards from descendant nodes, in a similar fashion as resource estimates and other information are propagated [3, 24]. For example, the current value of % COMPLETE of a parent plan node may be aggregated from the values of its child goal nodes. Metrics are aggregated according to their nature and the type of the node. For example, by default, a conjunctive plan node will aggregate % COMPLETE as the arithmetic mean of the children's values, while a disjunctive plan node will aggregate it as the maximum of their values. Mechanisms for aggregation have been explored in the cited literature. Since the algorithms are already parameterizable according to the nature of the quantity (in our case, the metric) and the type of the node, we need not repeat them.

The computation is to be made *dynamically* as the current situation evolves [18, 15]. We assume agents can assess the progress of leaf nodes. For instance, the police officers should believe they know when they have finished clearing a house (and so achieve the utility depicted on each leaf node). Hence, there are two types of metric values attributed onto nodes. The first type are *static*, initial, *a priori* values before execution (as depicted in Fig. 1). These correspond to expected, estimated, or required values, such as the utility expected upon full satisfaction of a goal (i.e., $u(s^*)$), and the resources expected to achieve this. The second type of metric values are *dynamic* estimates computed during execution, such as the utility achieved so far from a goal. For the progress metric of each goal instance g, the static value corresponds to the completion value a_{min} of g, while the dynamic value corresponds to the appraised value $\phi_g(s_{now})$.[5]

6.1 Reasoning in the Example Scenario

The response team commander is given the goal *publicSafety*. The doctrinal plan, *SecureAndClearArea*, involves the two subgoals, *secureArea* and *evacuatePeople*; the two may be achieved concurrently, although the team must be mindful that the public may (re-)enter the incident area until it is secured.

Goal templates, metrics, and goal instances. Recall from Example 1 that the goal template for *secureArea* is $T_{sa} = \langle \mathbb{R}, P, \phi_{sa}, \hat{\phi_{sa}} \rangle$, where the progress metric for T_{sa} is the achievement of its subgoals. The UTILITY metric of T_{sa} can be seen from Fig. 1 to be $u_{sa} = 5 * (\#\text{ achieved subgoals})$. u_{sa} may be of interest as a measure of progress, even though this metric does not *define* the progress (according to police doctrine) nor therefore the completion of the goal.

By contrast to *secureArea*, the progress metric of the initial goal *publicSafety* in the scenario is defined in terms of utility. Its goal template is $T_{ps} = \langle \mathbb{R}, P, u_{\Sigma}, \hat{u_{\Sigma}} \rangle$ where $\phi_{ps} \equiv u_{\Sigma}$ specifies the cumulative utility from the subgoals in the current plan for a goal instance of T_{ps}. This progress metric is computed in the obvious manner by recursively transversing the subtree below the goal instance, summing up the current utility estimates for each goal node. Likewise, the progress upper bound function, $\hat{u_{\Sigma}}$, can be computed by a recursive descent through the GPT. An *a priori* estimate can be computed, based on the upper bounds of the static, *a priori* utility attributions on leaf nodes [27, 3,

[5] An agent may be capable of directly computing the value of a metric at a (non-leaf) node. In that case, if the reasoning is consistent and the static values on leaf notes are reliable estimates, then the directly-computed and aggregated values should agree. Where they do not, the agent may resolve the conflict according to which of the two computations it believes is most reliable.

24]. For example, an *a priori* upper bound on *EstablishRoadblocks*, relaxing resource considerations, is $4+4+\max(2,5)=13$. Tighter bounds can be obtained by considering resource limitations and the resulting goal interaction and plan scheduling [27, 24].

Goal adoption. The police commander and her team are tasked with the initial goal *publicSafety*; its goal instance is $(40,10):T_{ps}$. The team of 10, including the commander, has too few officers to meet the expected requirements for the full completion of the three roadblock actions (rb_i) and the two house-clearance actions (h_i), let alone the forest. That is, the goal instance is unachievable (i.e., $\hat{\phi_{ps}} < a_{min}$), as can be seen to be the case by examination of the GPT.

Negotiation, delegation, and requesting help. At first, the commander considers allocating six officers for *secureArea* and weakening the *evacuatePeople* goal by omitting the *evacuateForest* subgoal. This is unacceptable to incident control. After further negotiation, control agrees to send urgently a second team to perform *rb1*. The commander thus allocates four officers for *secureArea*. Hence, the goal instances are $(20,4):T_{sa}$ and $(25,6):T_{ep}$. Two officers will search each house; when done, they will join the forest search.

Appraisal and sharing information. As execution proceeds, updated metric values are computed on the leaf nodes of the GPT and aggregated to parent nodes. This provides a situational assessment for the commander. Searching house 2 is taking longer than anticipated. Should the two officers continue with *h2*, or join those searching the forest? Utility of 4 is estimated achieved from *h2* after 25 minutes have elapsed. The original estimate of UTILITY for completion of the goal was 7; but this was only an *a priori* estimate based on typical experience. The commander appraises that the rate of achieving utility is outweighed by the resources employed, and so calls off the officers from house 2.

This extract from the scenario illustrates the more sophisticated reasoning enabled by and founded on a metric-based notion of partial goal satisfaction that is embedded into a concrete computational framework for the metrics.

7 Conclusion and Next Steps

The contribution of this line of work stems from the recognition of the need for a concept of partial goal satisfaction in cognitive agent frameworks, manifest in terms of the proposal of an abstract framework for partial goal satisfaction that identifies the main necessary ingredients for reasoning based on partial goal satisfaction. Our objective is a representation of partial satisfaction integrated into a reasoning framework, and allowing for a quantitative instantiation, in order that cognitive agent programming frameworks might be enhanced. The benefit of the topic and our approach is more sophisticated reasoning about goals, impacting reasoning about selection, adoption, and pursuit; goal progress appraisal; goal interaction; and inter-agent communication and collaboration.

Although we have indicated how our framework may be concretized in the context of GPTs, more work is needed to flesh out the details and investigate how advanced types of reasoning can be built on top of this basis and integrated into a programming framework. The modifications necessary to the semantics of a language such as GOAL [10] must be established and their correctness proved. To be investigated is how the various functions of our framework can be defined in concrete settings, and how existing work on, e.g., reasoning about resources can be used in this context. Also, while our framework provides the basis for reasoning about goal adaptation, it does not provide algorithms that allow the agent to decide how to adapt, weighing costs and benefits. This is an important area for future research, with just one relevant aspect being how to estimate cost and benefit projection into the future. Lastly, possible extensions are ripe for investigation, such as a logical instantiation with reasoning between goal outcomes, following Zhou et al. [32], inclusion of parameters in goal templates, and relation of templates in a hierarchy.

Acknowledgments

We thank David Martin for discussions and the reviewers for their helpful comments.

References

1. M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 14:349–355, 1988.
2. L. Braubach and A. Pokahr. Representing long-term and interest BDI goals. In *Proc. of ProMAS'09*, 2009.
3. B. J. Clement, E. H. Durfee, and A. C. Barrett. Abstract reasoning for planning and coordination. *JAIR*, 28:453–515, 2007.
4. M. Dastani. 2APL: A practical agent programming language. *JAAMAS*, 16(3):214–248, 2008.
5. M. B. Do, J. Benton, M. van den Briel, and S. Kambhampati. Planning with goal utility dependencies. In *Proc. of IJCAI'07*, 2007.
6. P. J. Feltovich, J. M. Bradshaw, W. J. Clancey, M. Johnson, and L. Bunch. Progress appraisal as a challenging element of coordination in human and machine joint activity. In *Proc. of ESAW'07*, 2007.
7. B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.
8. B. J. Grosz and L. Hunsberger. The dynamics of intention in collaborative activity. *Cognitive Systems Research*, 7(2–3):259–272, 2006.
9. P. Haddawy and S. Hanks. Representations for decision theoretic planning: Utility functions for deadline goals. In *Proc. of KR'92*, 1992.
10. K. V. Hindriks. Programming rational agents in GOAL. In *Multi-Agent Programming: Languages, Tools and Applications*. Springer, Berlin, 2009.
11. K. V. Hindriks, C. Jonker, and W. Pasman. Exploring heuristic action selection in agent programming. In *Proc. of ProMAS'08*, 2008.

12. K. V. Hindriks, W. van der Hoek, and M. B. van Riemsdijk. Agent programming with temporally extended goals. In *Proc. of AAMAS'09*, 2009.
13. R. Holton. Partial belief, partial intention. *Mind*, 117:27–58, 2008.
14. Z. Huang and J. Bell. Dynamic goal hierarchies. In *Proc. of the 1997 AAAI Spring Symp. on Qualitative Preferences in Deliberation and Practical Reasoning*, 1997.
15. E. Kamar, Y. Gal, and B. J. Grosz. Incorporating helpful behavior into collaborative planning. In *Proc. of AAMAS'09*, 2009.
16. G. Klein, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich. Ten challenges for making automation a "team player" in joint human-agent activity. *IEEE Intelligent Systems*, 19(6):91–95, 2004.
17. V. Lesser and et al. Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. *JAAMAS*, 9(1):87–143, 2004.
18. D. Morley, K. L. Myers, and N. Yorke-Smith. Continuous refinement of agent resource estimates. In *Proc. of AAMAS'06*, 2006.
19. B. Nebel and J. Koehler. Plan reuse versus plan generation: A theoretical and empirical analysis. *Artificial Intelligence*, 76(1–2):427–454, 1995.
20. A. S. Rao and M. P. Georgeff. Modeling agents within a BDI-architecture. In *Proc. of KR'91*, 1991.
21. M. Schut, M. Wooldridge, and S. Parsons. The theory and practice of intention reconsideration. *JETAI*, 16(4):261–293, 2004.
22. S. Shapiro and G. Brewka. Dynamic interactions between goals and beliefs. In *Proc. of IJCAI'07*, 2007.
23. S. Shapiro, Y. Lespérance, and H. J. Levesque. Goal change. In *Proc. of IJCAI'05*.
24. P. H. Shaw, B. Farwer, and R. H. Bordini. Theoretical and experimental results on the goal-plan tree problem. In *Proc. of AAMAS'08*, 2008.
25. M. P. Singh. A critical examination of use Cohen-Levesque theory of intentions. In *Proc. of ECAI-92*.
26. D. E. Smith. Choosing objectives in over-subscription planning. In *Proc. of ICAPS'04*, 2004.
27. J. Thangarajah, M. Winikoff, L. Padgham, and K. Fischer. Avoiding resource conflicts in intelligent agents. In *Proc. of ECAI-02*, 2002.
28. W. van der Hoek, W. Jamroga, and M. Wooldridge. Towards a theory of intention revision. *Synthese*, 155(2):265–290, 2007.
29. M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: A unifying framework. In *Proc. of AAMAS'08*, 2008.
30. M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah. Declarative and procedural goals in intelligent agent systems. In *Proc. of KR'02*, 2002.
31. Y. Zhou and X. Chen. Partial implication semantics for desirable propositions. In *Proc. of KR'04*, 2004.
32. Y. Zhou, L. van der Torre, and Y. Zhang. Partial goal satisfaction and goal change: Weak and strong partial implication, logical properties, complexity. In *Proc. of AAMAS'08*, 2008.