Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Reasoning with Partial Satisfaction in Goal Models: an Effort-Based Approach

Massimo Cossentino[1], Luca Sabatucci[1] and John Mylopoulos[2]

[1]Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), Via Ugo La Malfa, 253, Palermo, 90146, Italy.

[2]Department, University of Toronto, 214 College St, City, ON M5T 3A1, Canada.

Rapporto Tecnico N.:
RT-ICAR-PA-24-01 **Marzo 2024**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Via Ugo La Malfa 153, 90146 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Reasoning with Partial Satisfaction in Goal Models: an Effort-Based Approach

Massimo Cossentino[1], Luca Sabatucci[1] and John Mylopoulos[2]

[1]Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), Via Ugo La Malfa, 253, Palermo, 90146, Italy.

[2]Department, University of Toronto, 214 College St, City, ON M5T 3A1, Canada.

Rapporto Tecnico N.:
RT-ICAR-PA-24-01 **Marzo 2024**

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo, Via Ugo La Malfa 153, 90146 Palermo.

Reasoning with Partial Satisfaction in Goal Models: an Effort-Based Approach

Massimo Cossentino[1*], Luca Sabatucci[1] and John Mylopoulos[2]

[1*]Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), Via Ugo La Malfa, 253, Palermo, 90146, Italy.
[2]Department, University of Toronto, 214 College St, City, ON M5T 3A1, Canada.

*Corresponding author(s). E-mail(s): massimo.cossentino@icar.cnr.it;
Contributing authors: luca.sabatucci@icar.cnr.it; jm@cs.toronto.edu;

Abstract

Goals are used in many research fields to describe the desired states of affairs a system has to bring about. Naturally, in real-life situations, systems often fail to satisfy their goals. It is then essential to measure the degree of failure in order to determine appropriate mitigations. Proposals for measuring failure in the literature use probabilistic, statistical and fuzzy notions to answer questions such as “How many times did the system fail to satisfy a given goal over a period of time?”. This work proposes a different metric, Effort-to-Satisfaction (E2S), which estimates the effort required to turn partial satisfaction into a full one. In our proposal, atomic goals are expressed as formulas in bounded linear temporal logic and can be combined into composite goals through AND/OR refinement operations. E2S may be calculated for atomic goals through domain-specific functions, while for composite goals, E2S is computed using an algorithm that generates alternative solution strategies as a set of goals that are to be satisfied at specific times for accomplishing the goal model satisfaction. The paper also includes an extended example to illustrate the practical application of the proposed approach to a Forest Fire Management scenario.

Keywords: Goal, Goal Satisfaction, Partial Goal Satisfaction, Partial Goal Satisfaction Metrics

1 Introduction

In Self-Adaptive Systems (SASs), Requirements Engineering (RE), AI Planning (AIP) and Multi-Agent Systems (MASs), goals describe desired-states-of-affairs a system is supposed to bring about. Of course, in real-life situations, systems often fail to satisfy their goals fully. For such cases, a useful form of analysis studied in the literature involves estimating the degree of satisfaction for a goal to determine appropriate compensation, manual or automatic. Indeed, reasoning about partial goal satisfaction constitutes a prerequisite for modern systems to rationalize and customize their activities in a world of uncertainties and limited resources. In fact, the trendy research area of smart systems (smart cities, smart homes, smart transportation, smart contracts, smart anything) often relies on software agents that monitor the execution of a plan for fulfilling system goals and compensating for failures to fulfill them. Evaluating the degree of goal satisfaction constitutes a prerequisite for reasoning about mitigating system behaviour [18]. In a previous work [30], we proposed an approach to adaptation triggered by goal failure that is based on run-time (re)planning by composing available services to fulfil system goals.

There is general agreement in the aforementioned research areas that goal satisfaction needs to transcend a Boolean, all-or-nothing perspective [43]. Indeed, systems do not always need to satisfy their goals in an absolute, clear-cut sense. Some authors [23] suggest using high-level goals as an instrument for exploring and comparing alternative plans by evaluating their impact on the degree of satisfaction of goals, measured by some metric. Here, the degree of satisfaction guides the selection of the 'most preferred' alternative. There are many examples of reasoning with goals where Boolean (all-or-nothing) satisfaction constitutes a major barrier to utilizing smart systems in realistic environments. When no plan fully satisfies operative goals, the system needs to identify priorities to be pursued before others [21]. The same philosophy holds in AI Planning, even if there are techniques that have proven truly valuable for service composition [40] and in cognitive agents, acting in a resource-bounded real world, able to adopt plans that only partially achieve their goals [33, 36, 39].

Given such motivation, we need to introduce a notion of partiality for goal satisfaction [39]. Let us consider a simple example of a goal for an emergency management application stating that if a fire breaks out in a populated area, emergency crews need to "evacuate and cordon off th area". Of course, addressing this goal depends on the size of the area, personnel and resources available, weather conditions, etc. It could be that isolating the whole area is not feasible, whereas securing some zones (schools, parks, stores, and homes, for instance) could make strategic sense. Or, if even this alternative is not feasible, it may be sufficient to secure and control some zones only during daylight hours.

This work aims to address this general problem statement by proposing a particular metric for estimating Effort-to-Satisfaction (E2S) for goals expressed as bounded Linear Temporal First Order Logic (bLTFOL) formulas. The proposed metric measures the effort required in time to satisfy a goal. For the Forest Fire Management example above, that effort might be measured by resources and person-power needed to achieve the evacuation&control goal fully. This metric differs from earlier proposals that adopt probabilistic/statistical concepts, where partial satisfaction means "Achieve goal G with probability p, or n per cent of the time" [16], or fuzzy concepts where the goal is a fuzzy one and satisfaction can be partial with respect to a fuzzy metric. The most critical aspect of any new proposal for a partial goal satisfaction metric involves measuring E2S for composite goals. The proposed approach consists of an algorithm that, given a goal model and a set of already satisfied goals within it, calculates the possible alternative solutions in terms of sets of goals to be satisfied in time, also considering the implications of temporal formulas.

The proposed E2S metric bears similarities to a classical proposal of incorporating design costs in the design process. This is the Manheim scheme for deciding between alternative design solutions [24] on the basis of value, also cited in [35]. The scheme suggests a process based on a cost-benefit analysis composed of two steps: the first step creates several alternative top-level plans, while the second attaches a value to each alternative plan. Design proceeds by detailing the plan of the highest value. Manheim applied the process to the construction of a highway. The E2S metric may be used similarly, except that it is applied to goal models rather than highway designs. As such, it consists of domain-specific estimation rules for the effort required for the satisfaction of leaf goals and aggregation estimation rules for higher-level goals.

The contributions of this work are as follows:

- We present a novel metric, E2S, for partial goal satisfaction grounded on the concept of effort to bring about the full satisfaction of a goal.
- We propose an algorithm for calculating E2S for composite goals, given information about the effort required to carry out domain-specific actions.
- We illustrate the application of the framework with a realistic example involving a Forest Fire Management scenario.
- We discuss how E2S can be usefully applied in self-adaptive systems, multi-agent systems, AI planning, requirements engineering.

A crucial advantage of the proposed approach is that it allows us to compare the satisfaction of goals defined in terms of non-commensurable variables. If a goal concerns the construction of one kilometer of a road while another the lumen provided by light poles installed on the road, E2S allows us to abstract from the specific units of measure (Km, Lumen) and compare the advancement in the construction of the road with its illumination. This becomes relevant in all the scenarios where compromises are to be accepted while pursuing some goal, for instance, because of the lack of funds, resources, or time. Suppose the head of a development team for a new software release has a set of requirements, but she knows she cannot implement all of them in time for a preset deadline. She must select among trade-offs between time, cost, and degree of satisfaction of the requirements. Such trade-offs amount to balancing the partial satisfaction of goals with the extra effort she needs to complete the job. Techniques for choosing which goal is preferred and the degree of partial satisfaction to aim for are beyond the scope of this paper. Nevertheless, any such technique chosen can benefit from the availability of a metric such as E2S.

The proposed metric is a step in the direction suggested by Simon in [35]. In fact, as he claims, Engineering needs to move from the logic of "recipes" to rigorous scientific methods, including metrics, and this would finally enable the growth of a "Science of Artificial". Indeed, the general paradigm in most engineering approaches consists of the following steps: (i) defining the goals that will solve the problem at hand, (ii) designing the blueprint of the solution artefact, and then: "Given a blueprint, to find the corresponding recipe" [35]. Conversely, the scientific paradigm is "Given the description of some natural phenomena, to find the differential equations for processes that will produce the phenomena [35]".

The rest of the paper is organized as follows: Section 2 reviews proposals for measuring partial goal satisfaction and literature contributions that measure goal achievement in terms of effort. Section 3 introduces preliminary definitions and concepts, also extending the theory of partial satisfaction to temporal operators, while Section 4 introduces the concept of effort. Section 5 defines the E2S metric and introduces an algorithm for calculating E2S of composite goals and an approach for calculating E2S of atomic goals. Section 6 provides an extensive example. Section 7 proposes a discussion of the advantages and limits of the approach, it also suggests some application domains for the proposed metrics. Conclusions are drawn in Section 8, which also includes some considerations about our foreseen future works.

2 State of the Art

In AI, as well as in Requirements Engineering (RE) and many other research fields, partial goal satisfaction is a major concern because goals are often fully/partially conflicting. This is especially true for quality goals, such as security, usability and performance ones, where the satisfaction of one, say a security goal, generally contributes negatively to the satisfaction of others, say usability and performance ones. The NFR framework [7, 25], intended to model quality (aka non-functional requirements, includes relationships for partial/-full conflicts and synergies between two goals. The framework also supports a form of qualitative reasoning over goal models whereby once leaf-level goals are marked S (satisfied) or D (denied), higher level goals are also marked in terms of S, D, PS (partially satisfied) and PD (partially denied). This line of research was continued in [15] and [32] with a formalization of goal relationships and the development of reasoning tools that use SAT solvers to answer questions such as "Given a certain S/D marking for leaf goals, which root goals are satisfied?" (bottom up reasoning), and "Given a goal model, is there a marking of leaf goals that satisfy all root goals?" (top down reasoning). [15] includes a case where partial goal satisfaction is measured and reasoned with in probabilistic terms. Our proposed E2S metric offers a quantitative variant of the NFR reasoning framework, where the propagation rules from leaf to root goals are motivated by the notion of effort. However, the E2S proposal does not include a partial satisfaction/denial relationship between goals, while the NFR framework does not include the quantification of the partial goal satisfaction.

Letier and van Lamsweerde in [23] describe a technique for quantifying the impact of alternative system designs on goal satisfaction. A probabilistic layer enriches a goal model and is used to reason about goal satisfaction. The paper also deals with non-functional goals that are specified in probabilistic terms, and their satisfaction depends on domain-specific rules. A comprehensive comparison of goal-oriented satisfaction analysis techniques is presented by Horkoff and Yu in [17].

Feng Lin Li et al. [14] propose three metrics for talking about partial goal satisfaction, measuring respectively (1) universality, the percentage of goal instances that are satisfied; (2) gradability, the degree to which a goal is satisfied; (3) social agreement, the percentage of stakeholders who agree that a goal is satisfied. For a requirement such as "Ambulance shall be at the scene of an accident within 15mins of an emergency call", universality measures the percentage of emergency calls where the requirement is satisfied; gradability might measure how close to the scene an ambulance gets within 15mins, while social agreement measures the percentage of stakeholders who think that the goal is satisfied. The relationship between functional goal satisfaction and its quality attributes is also discussed by Chi Mai Nguyen et al. in [26], where a goal modelling language is proposed that makes explicit the notion of goal refinement and domain assumption. This allows for defining constraints and optimization goals over multiple objective functions. An SMT/OMT solver is used to reason with goal models expressed in the proposed language.

Partial goal satisfaction is a central concern for Self-Adaptive Systems (SASs), where the degree of satisfaction may determine the adaptation chosen by a self-adaptive system, see [2, 34, 43]. For example, for a meeting scheduling system, the adaptation mechanism monitors the number of failures in scheduling a meeting over time and determines what adaptation is appropriate. Adaptations can take different forms, sometimes, they consist of choosing an alternative path within a goal model for satisfying root-level goals. In other cases, goals are relaxed or altogether replaced by new goals.

Belcomo et al. [3, 4, 27, 28] propose a partially observable Markov Decision Process for supporting run-time reasoning about partial satisfaction of non-functional requirements (NFRs) and their tradeoffs within a dynamic environment. Dell'Anna et al.[10–12] present a norm/requirement revision framework guided by information retrieved from runtime execution data. Relaxation, strengthening and alteration are enacted, at run-time, based on a Bayesian network according to assumptions made in the requirements model.

Zhou et al. [42] propose the partial implication operator for First-Order Logic, intended to capture the partial satisfaction relationship between two propositional formulas: formula $x \wedge z$ partially implies $x \wedge y$ since x is part of $x \wedge y$ and it is a logical consequence of $x \wedge z$. The new operator changes the nature of implication in the First-Order Logic drastically. Van Riemsdijk and Yorke-Smith in [39], propose a higher-level framework based on metric functions representing the progress towards achieving a goal. Progress appraisal [13] is the capability of an agent to assess what parts of a goal have been achieved. The framework does not detail what partial satisfaction metrics may be used, though it mentions several alternatives.

In [18], Jureta et al. face the problem that a system may not be able to satisfy all functional and quality requirements and may need to consider solutions that partially satisfy some requirements. To identify such solutions, they suggest (1) Setting 'minimum acceptable values' for measurable qualities; (2) Relaxing a requirement to allow the system to fail it some percentage of the time, as with probabilistic goals and the universality metric of[14], (3) For ambiguous requirements, such as "Ambulance shall arrive quickly to the scene of an accident", use domain-specific heuristics.

Almagor et al. [1] enrich LTL specification of goals with quantitative operators that allow reasoning about the quality of satisfaction. A similar approach is reported in [38] where Tumova et al. consider temporary violations of parts of the goal formula. Their proposal requires prioritization of some clauses of the goal formula, which is used to plan the strategy that includes fewer violations. As discussed in [23], analyzing the weights of the different clauses of a goal formula may be a hard task and requires a relevant degree of domain expertise. Similarly, in [21], Lahijanian et al. introduce a distance-to-satisfaction metric of the additional cost of a trajectory that is not optimal. The metric selects the path that minimizes the violation of the LTL goal formula when full satisfaction cannot be achieved. In [19, 20], Kim and Fainekos introduce methods for calculating distances between Buchi-automata representing goals and propose algorithms for revising these goals. Unfortunately, these algorithms are NP-hard and do not scale with goal size.

Another interesting research trend concerns goal adaptation techniques [22, 33, 39, 43] used when the desired state-of-affairs cannot be achieved. Vukovic et al. [40] introduce GoalMorph, a framework for goal transformation that constructs context-aware goals reformulating failed goals into problems that can be solved using an AI planner. In general, goal adaptation techniques are domain-specific and difficult to adopt in new domains.

Finally, Thangarajah et al. [36, 37] adopt an approach based on resource analysis to provide a BDI agent with a quantitative measure of the number of resources consumed in satisfying a goal. This approach is strictly domain-dependent and may be used only when there is a clear domain-specific link between goals and resources.

From the analysis of state of the art, we highlight the need for a novel framework for dealing with partial goal satisfaction that covers the following points:

- It supports temporal specification of goals, such as "While the fire is out of control, restrict access to populated areas in the vicinity"; thus defining precisely the relationship between time and partial satisfaction;
- It allows goals such as "Call the Fire Department and activate the 12 sirens in the neighbourhood" that combine atomic predicates with quantified predicate formulas for measuring partial satisfaction;
- It minimizes additional conceptual complexity for designers and runtime computational complexity.

In the following section 3, we define the grammar we use to specify our goals, and in the section 5 we introduce the concept of effort-to-satisfaction that we propose to use for estimating the degree of satisfaction of a goal.

3 Goals

Goals are expressed in bounded Linear Temporal First-Order Logic (bLTFOL), which is First-Order Logic with quantification over finite sets, enriched with Linear Temporal Logic operators defined over finite intervals.

Formally, a goal g is specified by an expression of the form:

$$\begin{aligned} < g > ::= \; & < \phi > | < g > \mathit{and} \ldots \mathit{and} < g > | \\ & | < g > \mathit{alt} \ldots \mathit{alt} < g > \\ < \phi > ::= \; & < \mathit{atom} > | \neg < \phi > | < \phi > \wedge < \phi > | \\ & < \phi > \vee < \phi > | \\ & \forall x/D[<\phi(x)>] \mid \exists x/D[<\phi(x)>], \\ & \mathbf{F}(<\phi(t)>, <i>) \mid \mathbf{G}(<\phi(t)>, <i>) \mid \\ & \mid \mathbf{U}(<\phi(t)>, <\phi(t)>, <i>) \end{aligned} \tag{3.1}$$

where $\mathbf{F}, \mathbf{G}, \mathbf{U}$ are respectively the LTL operators Finally, Globally and Until, ϕ is a logical formula in bLTFOL, i is a discrete, finite time interval, D is a finite set, and $\phi(x), \phi(t)$ indicate respectively a bLTFOL formula with free variable x or t.

$$< i > ::= [< t >, < t >],$$

where t is a time point,

$$< \mathit{atom} > ::= < \mathit{pred} > (< \mathit{param} >, \ldots, < \mathit{param} >),$$

where *pred* is a predicate name for an atomic goal and *param* is a constant or a variable.

Partial satisfaction of g, $P(g)$ is defined as a set of tuples:

$$P(g) = \{(\mathit{pred}_1(c_{11}, \ldots, c_{1n}), e_1), \ldots, (\mathit{pred}_m(c_{m1}, \ldots, c_{mn'}), e_m)\}$$

where $\mathit{pred}(c_{i1}, \ldots, c_{in})$ is an atomic goal and e_1 is its E2S deficit ($= 0$ for fully satisfied goals); also all the atomic goals in $P(g)$ are elements of a plan that satisfies g.

These rules mean that goal models are labelled, directed graphs where nodes are goals that are and/alt(ernative)-refined into other goals until we reach leaf goals defined by a bLTFOL formula. The semantics of bLTFOL are standard, while the semantics of and/alt are those used in [15], as follows:

$$\begin{aligned} & AND(g_1, \ldots, g_n, g) \wedge S(g_1) \wedge \cdots \wedge S(g_n) \Rightarrow S(g) \\ & AND(g_1, \ldots, g_n, g) \wedge \exists i/D(g_i) \Rightarrow D(g) \\ & ALT(g_1, \ldots, g_n, g) \wedge \exists i/S(g_i) \Rightarrow S(G) \\ & ALT(g_1, \ldots, g_n, g) \wedge D(g_1) \wedge \cdots \wedge D(g_n) \Rightarrow D(G) \end{aligned}$$

where S(g), D(g) mean "g is satisfied", "g is denied" respectively.

It should be noted that bLTFOL is more expressive than, but logically equivalent to Propositional Logic since quantifiers can be replaced by propositional formulas, as in:

$$\forall x/D[\phi(x)] \equiv \phi(d_1) \wedge \phi(d_2) \wedge \cdots \wedge \phi(d_n),$$

where D = $d_1, d_2, \ldots, d_n$ and temporal operators can be replaced by quantifiers e.g.,

$$\mathbf{F}(\phi, \iota) \equiv \exists\ t\ /\ i\ [\phi(t)]$$

This equivalence is very useful in calculating E2S because the calculation is carried out with respect to the equivalent propositional expression for a goal g.

Let us consider the following example of *SecureNeighbourhood* goal defined as follows:

$$\begin{aligned} &SecureNeighbourhood ::= \\ &\quad \mathbf{G}_{[1,120]}[\mathit{deployed\_roadblocks}(20) \wedge \\ &\qquad \wedge\, \mathit{activated\_sirens}(12)] \end{aligned} \tag{3.2}$$

According to this specification, goal *SecureNeighbourhood* is satisfied when: *(i)* 20 roadblocks have been deployed, and *(ii)* 12 sirens are sounding to warn the population. All of that should continuously hold (Globally operator, **G**) since time t=1 to t=120.

Now let us consider a portion of the goal model of Fig. 2 that will be adopted for the extensive example in Sect. 6.

The goal we introduced in formula (3.2), is decomposed in Fig. 2 into two sub-goals: SoundAlarm, and EstablishRoadblocks (corresponding to the predicates used in formula (3.2)). According to this representation of the goal, we could formalize that as follows:

$$\begin{aligned} SecureNeighbourhood = &SoundAlarm \text{ AND} \\ &EstablishRoadblocks \end{aligned}$$

4 Effort

Phenomena of resistance-to-change abound in Science and Engineering. Consider kinematics in Physics where resistance-to-change , aka 'mass' (M), for physical objects is related to 'change in velocity', aka 'acceleration' (A) and 'force' (F) through Newton's Second Law, A * M = F. Again in Physics, 'friction' is a force that decelerates a moving physical object in accordance with Newton's Law. In Electrical Circuit Theory, 'resistance' (R) is related to 'change in current' and 'voltage' (the force) through a similar equation, R * I = V. And in Software Engineering, 'resistance to change' of a legacy software system is related to 'degree and rate of change' and 'effort' that play the role of 'change' and 'force' in physical phenomena. In Management Science, resistance to 'change of enterprise architectures' is related to 'change in the management of projects' (the force). As well, in the Social Sciences 'resistance to change' (aka 'conservatism') is related to 'social/political change' and 'social forces'. Finally, in Cognitive Science, 'cognitive inertia' (the resistance) is related to a 'cognitive achievement', such as new research results (the change) and thinking (the force). All cases use the same concepts, which we call the CRE metaphor, where 'change' C is related to 'resistance' R and 'effort' E through an equation of the form C * R = E, where the units for measuring the three quantities are domain-dependent, and for some domains, they don't even exist. For such cases, we can't claim any equation relating to the three qualities. We can, however, claim that if the resistance remains constant, the force required grows with a growing target change, while if change remains constant, the effort required grows with growing resistance. In this work, we adopt the CRE metaphor to measure the gap between partial and full satisfaction.

In Physics, effort amounts to a physical force that can be applied to an object instantaneously or over a period of time. In Engineering and Management, effort consists of the human and other resources deployed towards building or changing an artefact, be it a physical building, a software system, or an enterprise. And resistance is the innate inertia of all things physical, mental or social to resist change. Since goal satisfaction is by definition a problem of changing a state of affairs into a desired one, the goal, the C, R, E metaphor seems very appropriate for understanding and measuring partial satisfaction.

In our review of the literature, we noted four metrics proposed for measuring partial goal satisfaction: probabilistic, statistical, fuzzy and social. Probabilistic metrics rely on a priori probabilities. For example, if the desired state-of-affairs is 'coin heads up', then the action of flipping the coin has a probability of 0.5 of satisfying the goal. Statistical metrics rely on a posteriori probabilities, statistics, as with "Ambulance shall arrive at the scene of an accident within 15 minutes 95% of the time". Fuzzy metrics measure how much of the desired state of affairs has been achieved. For instance, if an ambulance arrives within 100 meters from the scene within 15 minutes, the ambulance dispatch requirement has been almost achieved, where 'almost' means (d - 100m)/d and d is the distance from the ambulance's initial location to the scene. Finally, social metrics measure partial satisfaction as the percentage of stakeholders who think the goal has been fully satisfied. So, if the ambulance arrives within 100m from the scene and can't get any closer because of in-between obstacles, then some stakeholders may decide that the goal has been fully satisfied, leading to a satisfaction value in the range between 0 and 1. Now, note that all these types of metrics cater to stakeholders who are users of the ambulance dispatch service and talk in different terms about the quality of the service.

The project manager that builds the Ambulance Dispatch system is also a stakeholder. However, her concerns are different from those of users of the service provided by the system-to-be. If she determines that, on the basis of existing traffic data and the current fleet of ambulances, the scene will be reached within 15 minutes 70% of the time, then she needs to consider growing the fleet size to improve the chances the requirement will be satisfied. The number of new ambulances required to satisfy the requirement is now measured in terms of effort. In summary, E2S is a metric that caters to stakeholders that are part of the development team for the system-to-be rather than users. Indeed, there is much anecdotal evidence about developer-think as they struggle to satisfy the requirements of their project fully, and that evidence suggests that they think in terms of resources needed to achieve full, or at least improve partial, satisfaction rather than the quality of the services offered by the system-to-be.

But why use effort instead of change or resistance to measure the gap between partial and full satisfaction? Resistance-to-change is notoriously difficult to grasp, observe and measure. This is why when we want to measure the mass of an object, we apply a standard force, gravity, and measure the change of location on a weight scale, i.e., we fix the force and measure change to determine resistance. Or, when we measure E2S for the Forest Fire scenario, we don't try to measure all the obstacles to achieving the goal at hand, but rather the estimated effort required to achieve the goal, despite the obstacles. And change is generally measured by the budget required, i.e. by the resources needed to bring it about.

5 Effort-to-Satisfaction (E2S)

The E2S metric is a function of two arguments $E2S(G, P_G)$ where G is a goal defined as presented in section 3, while P_G is a sub-plan of G, i.e., a set of sub-goals of G, possibly negative ones (i.e., of the form $\neg G'$), that have been satisfied so far.

In the sequel, we define a complement of P_G as another sub-plan of G, say P'_G such that $P_G \cup P'_G$ satisfy G.

E2S measures the minimum missing effort required to fully satisfy G. Formally,

$$E2S(G, P_G) = min\{effort(Complements(G, P_G))\} \tag{5.1}$$

where $Complements(G, P_G)$ is the set of all complements of P_G relative to G.

We assume that negation is only applied to single goals, as in $\neg G$, rather than goal expressions, as in $\neg(G_1 \wedge G_2)$.

5.1 E2S of Goal Models

In this subsection, we will discuss how to calculate the E2S of a goal model (that is equivalent to calculate the E2S of its root goal) given that some of the sub-goals have been already satisfied. The approach is general, and it may also be used to calculate the E2S of any portion of a goal-tree.

The proposed solution is represented by the Algorithm 1 that identifies all complements through iterative refinements. It uses two main data structures: *Goals* is a list of goals to be still analysed (line 1); *Compl* is a list of sets (line

Algorithm 1 Calculation of $Complements(G, P_G)$

1: $Compl$:= [{G}] ▷ *stores the set of complements identified so far*
2: $Goals$:= [G] ▷ *stores unsatisfied goals identified so far*
3: **for all** g in $Goals$ **do**
4: remove g from $Goals$
5: **if** $g ::= g_1 \wedge \cdots \wedge g_n$ **then** ▷ *g is defined by a conjunction*
6: add each element of the set $\{g_1, \ldots, g_n\} - P_G$ to $Goals$
7: replace g with $\{g_1, \ldots, g_n\} - P_G$ in $Compl$
8: **else if** $g ::= g1 \vee \cdots \vee g_n$ **then** ▷ *g is defined by a disjunction*
9: ▷ *Note: none of $g_1, \ldots, g_n$ is satisfied otherwise g would be satisfied too*
10: add each goal $g_1, \ldots, g_n$ to $Compl$ and to $Goals$
11: **else if** $g ::= \forall x/D[g(x)]$ **then** ▷ *g is defined by an universal quantifier*
12: add each $g(d)$, $d \in D$ which is not in P_G to $Goals$
13: replace g with $\{g(d_i)\} - P_G$ in Compl
14: **else if** $g ::= \exists x/D[g(x)]$ **then** ▷ *g is defined by an existential quantifier*
15: add each $g(d), d \in D$ which is not in P_G to $Goals$
16: replace each complement c that includes g with k complements that are copies of c
17: but replace g with $g(d_j)$ where $g(d_j)$ is in $\{g(d_i)\} - P_G$ in $Compl$
18: **else if** $g ::= \mathbf{G}[g(t), i]$ **then** ▷ *g is defined as a Globally operator*
19: add each $g(t), t \in i$ which is not in P_G to $Goals$
20: replace all occurrences of g with the elements in the set $\{g(t)\} - P_G$ in $Compl$
21: **else if** $g ::= \mathbf{F}[g(t), i)]$ **then** ▷ *g is defined as a Finally operator*
22: add each $g(t), t \in i$ which is not in P_G to $Goals$
23: replace each complement c that includes g with k complements that are copies of c but replace g with $g(t)$ for each $g(t)$ in $\{g(t) \mid t \in i\} - P_G$ in $Compl$
24: **else if** $g ::= \mathbf{U}[g_1(t), g_2(t), i]$ **then** ▷ *g is defined as an Until operator*
25: add $g_1(t), g_2(t)$ for $t \in i$ to $Goals$, except for cases where they are elements of P_G,
26: replace every complement that includes an instance of g,
27: carbon-copy it many times,
28: replace in each copy g with $g_1(1), g_2(2), g_1(1), g_1(2), g_2(3), g_1(1), g_1(2), g_1(3), g_2(4)$ etc.
29: leaving out any $g_1(i)$ or $g_2(i)$ that appear in P_G
30: **else if** g is an atomic goal **then**
31: do nothing ▷ *This algorithm terminates when Goals is empty.*
32: **end if**
33: **end for**

2), where each set describes a complement, i.e. sub-plan for addressing a goal or a part of that.

Initially, each complement is described by a super-plan that subsumes the complement but includes goals that have already been satisfied. For example, initially, there is one complement $\{G\}$, meaning that if the complement is satisfied, then G will be satisfied.

The main loop of the algorithm (lines 3-36) refines a goal in $Goals$ according to its definition. This way, all sub-goals that are in P_G are removed, and only unsatisfied ones remain.

When a goal g is defined as a conjunction of sub-goals $g_1, \ldots, g_n$ (line 5-7), g is replaced by its subgoals g_i, where g_i has not been already satisfied, and a unique set containing all these subgoals is added to Compl.

When a goal g is defined as a disjunction of sub-goals $g_1, \ldots, g_n$ (line 8-10), then every complement where g appears is carbon-copied n times and in these copies g is replaced by g_i respectively.

The same principles apply for universal/existential quantification, Finally (**F**) and Until (**U**) because each of these operations introduces alternative plans and, therefore, alternative complements. A universal quantifier translates into a conjunction of terms over a set (lines11-13), whereas an existential quantifier translates into a disjunction of terms over a set (lines 14-17).

Similarly, a Globally operator translates into a conjunction of terms over time (lines 18-20), and a Finally is a disjunction of terms over time (lines 21-24).

The refinement process ends when atomic goals are reached (lines 31-34). The algorithm terminates when Goals is empty. Since we disallow circular definitions, termination is ensured.

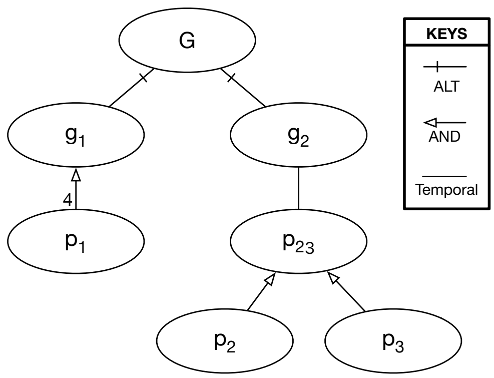

Fig. 1: The goal model used to provide an example of the procedure for calculating $E2S(G, P_G)$.

To exemplify Algorithm 1, we provide a running example with the goal model reported in Fig. 1, where:

$$G = g_1 \text{ ALT } g_2 \tag{5.2}$$
$$g_1 = p_{1_1} \wedge p_{1_2} \wedge p_{1_3} \wedge p_{1_4} \tag{5.3}$$
$$g_2 = \mathbf{F}_{1,3}(p_{23}) \tag{5.4}$$
$$p_{23} = p_2 \text{ AND } p_3 \tag{5.5}$$
$$p_{1_i}, p_2, p_3 \text{ are predicates}$$

Our aim is to calculate the E2S for goal G at t=0. As already stated, we use the AND/ALT operators to represent conjunction/ disjunction relationships between goals. We also use a *Temporal* relationship to address the situation when the super-goal formula includes a temporal operator. In this case, the argument of the temporal operator becomes a sub-goal of the goal-tree (i.e. goal g_2 and its sub-goal p_{23}, see also its definition formula (5.4)). Another interesting situation is represented by g_1 that is the conjunction of four different instances p_{1_i} of the predicate p_1. This is for instance, what happens if we define the goal $DeployRoadblocks(4)$. This is achieved when four different roadblocks are deployed, which means four different instances of the predicate *RoadblockDeployed* are True. This situation in the goal tree is reported as a small number just above the atomic goal.

Now, let us suppose $P_G = [p_{1_2}, p2_{(t=3)}]$, that means: p_{1_2} (the second instance of p_2) holds at t=0, and it will hold in the next time steps (some maintenance effort could be considered here but we are neglecting that for simplifying the example), also p_2 holds at t=3.

First steps, we initialise the *Compl*, and *Goals* variables: $Goals := [G]$,
$Compl := [\{G\}]$

Then we perform the first loop (**loop on goal G**) starting from line 8 (G is a disjunction of two sub-goals), we add g_1, g_2 to *Compl*. Result:

$$Goals := [g_1, g_2],$$
$$Compl := [\{g_1\}, \{g_2\}].$$

Now we proceed with a **loop on goal $\mathbf{g_1}$**. Goal g_1 is a conjunction of 4 atomic goals, we proceed as specified at line 5 of the algorithm. We add each element of the set of sub-goals of g_1 (that is not satisfied) to *Goals* and we replace g_1 with the same set in *Compl*. Result:

$$Goals := [g_2, p_{1_1}, p_{1_3}, p_{1_4}],$$
$$Compl := [\{p_{1_1}, p_{1_3}, p_{1_4}\}, \{g_2\}].$$

Next we perform a **loop on goal $\mathbf{g_2}$**. This goal is defined by a Finally operator and therefore we proceed as specified at line 21, we go through the temporal relationship in the goal three and we consider the 3 possible times the $p_2 3$ goal is evaluated in the Finally condition, they are: $p23_{(t=1)}, p23_{(t=2)}, p23_{(t=3)}$.We now add them to the variables, result:

$$Goals := [p_{1_1}, p_{1_3}, p_{1_4}, p23_{(t=1)}, p23_{(t=2)}, p23_{(t=3)}],$$
$$Compl := [\{p_{1_1}, p_{1_3}, p_{1_4}\}, \{p23_{(t=1)}\}, \{p23_{(t=2)}\}, \{p23_{(t=3)}\}].$$

We should now perform the **loops on $\mathbf{p_{1_1}, p_{1_3}, p_{1_4}}$**. We report the result of all of them at the same time since, according to line 30 of the algorithm, in each loop, we do nothing (we just remove the goal from *Goals*); the result is:

$$\begin{aligned}Goals :=& [p23_{(t=1)}, p23_{(t=2)}, p23_{(t=3)}],\\ Compl :=& [\{p1_1, p1_3, p1_4\}, \{p23_{(t=1)}\}, \{p23_{(t=2)}\},\\ & \{p23_{(t=3)}\}].\end{aligned}$$

Now we perform the **loop on goal $\mathbf{p23_{(t=1)}}$**. This is a conjunction of two (atomic) sub-goals p_2, and p_3 that are to be considered at t=1. The result is:

$$\begin{aligned}Goals :=& [p23_{(t=2)}, p23_{(t=3)}],\\ Compl :=& [\{p1_1, p1_3, p1_4\}, \{p2_{(t=1)}, p3_{(t=1)}\},\\ & \{p23_{(t=2)}\}, \{p23_{(t=3)}\}].\end{aligned}$$

The **loop on goal $\mathbf{p23_{(t=2)}}$** requires the same procedure we did before, but at t=2, resulting in:

$$\begin{aligned}Goals :=& [p23_{(t=3)}],\\ Compl :=& [\{p1_1, p1_3, p1_4\}, \{p2_{(t=1)}, p3_{(t=1)}\},\\ & \{p2_{(t=2)}, p3_{(t=2)}\}, \{p23_{(t=3)}\}].\end{aligned}$$

Finally, the **loop on goal $\mathbf{p23_{(t=3)}}$** has a peculiarity, goal $p2_{(t=3)}$ holds and therefore we do not add it to *Compl*. The final result we obtain is:

$$\begin{aligned}Goals :=& [\,],\\ Compl :=& [\{p1_1, p1_3, p1_4\}, \{p2_{(t=1)}, p3_{(t=1)}\},\\ & \{p2_{(t=2)}, p3_{(t=2)}\}, \{p3_{(t=3)}\}].\end{aligned}$$

We can now calculate the E2S for satisfying goal G as follows:

$$\begin{aligned}E2S(G, P_G) =& min\{E(Compl)\} =\\ =& min\{E([\{p1_1, p1_3, p1_4\},\\ & \{p2_{(t=1)}, p3_{(t=1)}\}, \{p2_{(t=2)}, p3_{(t=2)}\},\\ & \{p3_{(t=3)}\}])\} =\\ =& 2 * E_A\end{aligned}$$

where we suppose: $E(p1_1) = E(p1_3) = E(p1_4) = E(p3_{(t=3)}) = E_A$.

5.2 E2S of Atomic Goals

The first fundamental question, in adopting Algorithm 1, is related to how to calculate the E2S associated with a simple predicate.

Several other authors studied the problem of estimating the partial satisfaction of non-decomposable goals. A common approach is adopting some informed metrics for estimating the satisfaction [39]. This is a good solution when domain knowledge is available because it allows a fine metric for appreciating advancement in goal achievement. In our work, we mix domain functions and domain-independent values.

The first consideration is that we distinguish four different functions to be associated with atomic goals, according to distinct types of action:

- E_A is the effort associated with the Achievement of g, $F(g, i)$, meaning that g will be satisfied sometime in i.
- E_M is the effort associated with the Maintenance of g, $G(g, i)$, meaning that g will be maintained satisfied during i.
- E_D is the effort associated with the Denial of g, $F(\neg g, i)$, meaning that g will be denied sometime in i.
- E_P is the effort associated with the Prevention of g, $G(\neg g, i)$, meaning that g will be prevented throughout i.

The simplest case we propose to use is that E_A, E_M, E_D and E_P are constant values, this is a good choice when domain functions are unavailable or when they depend on purely qualitative assessments.

For example, if the system must address an atomic goal $p1$, we can assert the associated effort is $E_A(p1)$

However, E_A, E_M, E_D and E_P could also be defined as domain functions to represent a different kind of associated effort (for example, a non-linear behaviour). In this case, they are functions of some domain variables that directly measure a goal's current degree of satisfaction.

For example, if the system must maintain a goal $g = \forall x \in D, pred(x)$, the reader owns two ways of calculating the associated effort:

1. the goal could be translated into a conjunction of k atomic predicates (where k is the size of D), and therefore calculating the effort as the sum of k constant values ($k \cdot E_M(pred)$), or
2. a domain function can be used to evaluate the whole in a single step (for instance, if the goal is to ensure 20 roadblocks, the function could be

$E_M(unset) = Total\_Effort * unset/20))$ where $unset$ is a domain variable, and $Total\_Effort$ is a constant.

The use of domain functions introduces flexibility in modelling the resulting effort. For example, it is possible to use non-linear functions for empathizing an exponential/logarithmic growth or a non-sinusoidal waveform.

An immediate advantage of non-linear domain functions is to integrate thresholds for implementing value tolerance easily. Let us suppose a goal is fully satisfied when 10^6 sub-predicates hold. Indeed this is a sharp condition that may be too constraining in many real situations. In fact, it is to be expected that there would be no practical difference in fulfilling their total number minus five. Such tolerance is a flexibility element that we think adds a significant contribution to how a designer may model the problem at hand, and to how she can evaluate different solution strategies using the proposed E2S metric.

For relaxing the satisfaction constraints in goal formulas, we introduce the ***Value Threshold*** (VT) concept, i.e. a tolerance in achieving a goal when its satisfaction depends on several sub-predicates. The VT may be a value, a function of time, and of other parameters, although in the following examples, we will only refer to some fixed value for the sake of simplicity.

The simplest way to implement a VT is to modify the corresponding domain functions by adding a threshold. Supposing a goal ψ is the conjunction of n sub-predicates a_i, ψ_t is said to be (fully) satisfied at time t, if $n_A \geq n - n_{F_{Max}}$ at t, where:

- n_A is the number of achieved (satisfied) sub-predicates at time t.
- $n_{F_{Max}} = Round(n * VT)$ is the (integer) maximum number of not achieved sub-predicates that can be accepted for the predicate ψ still held.

This way, the success condition is built starting from the longest distance (maximum value of E2S) and progressively approaching the target value (minimum value of E2S). There are situations where the domain may require some specific value to be maintained for the predicate and oscillations may happen with positive/negative elongations from the target value. If this is the case, the previous satisfaction condition is to be modified as follows:

$$n - n_{F_{Max}} \leq n_A \leq n + n_{F_{Max}}$$

In order to calculate the value of E2S for a goal, weakened by a value tolerance VT so that $n_A \geq n - n_{F_{Max}}$, we introduce the following formula:

$$E2S_{\psi_t} = \begin{cases} E_A \cdot & \frac{n - n_{F_{Max}} - n_A}{n - n_{F_{Max}}}, \\ & (n - n_{F_{Max}} - n_A) \geq 0 \\ 0, & (n - n_{F_{Max}} - n_A) < 0 \end{cases} \quad (5.6)$$

The second row of this formula considers the case where the number of achieved sub-predicates n_A is greater than the success threshold, i.e.: $n_A > n - n_{F_{Max}}$. This means that despite the ψ predicate being already satisfied (condition $n_A = n - n_{F_{Max}}$), other sub-predicates are progressively achieved, thus reinforcing the fulfillment of ψ. This may seem an odd situation, but it is the logical consequence of accepting some tolerance in the satisfaction of goals depending on domain variable, the effort spent in satisfying these sub-predicates is not mandatory for achieving the satisfaction condition, and therefore it should not be considered in the $E2S_{\psi_t}$ because we are always concerned about calculating the minimum effort required to achieve the satisfaction, disregarding all the situations that require a greater effort.

So far, we have discussed how to weaken goal specifications by looking at the satisfaction of predicates but forgetting other relevant aspects of such specifications, namely: time constraints imposed by temporal operators. For instance, the goal: $g = G_{[0,300]} \forall x \in Sirens, SoundingSiren(x)$ prescribes that a given number of sirens (let us say 14) continuously play from time t=0' to t=300'. It is reasonable to think that if this condition is violated for a short time (for instance, 2 minutes) over the prescribed 5 hours, the damage to the expected system's behavior is very limited, if any. Such a situation suggests introducing some kind of time tolerance over the temporal condition imposed by the Globally operator.

Similar constraints may be relaxed in other operators as well. For instance, the Until condition imposes that in the goal $g = \phi \; U_{[t_0,t_{Max}]} \; \psi$, the ϕ variable holds until the time t where ψ

holds. What if ϕ holds until t-2 and ψ holds at t? This gap violates the Until operator prescriptions, but would it be dramatic in a practical application? Of course, that depends on the application domain, but these and a few other situations are worth studying and offer interesting opportunities for weakening the goal specifications. This study would offer the designer the chance to make the best profit from the opportunity offered by evaluating different goal-tree satisfaction strategies compared by means of the proposed E2S metric. Relaxing temporal constraints is a very complex issue, it would lead the current work in another direction, and therefore we postpone that to a future, different work.

6 An Evaluation Of The Proposed Approach

In this section, we propose an application of the proposed approach that is aimed to demonstrate how the E2S can be used in planning the alternative strategies for satisfying a given goal tree but also for appreciating, at runtime, the progress towards goal achievement.

The Fire Forest example. To illustrate the proposed approach, we refer to an extension of a well-known case study from the literature [39] that we extend to demonstrate the use of our newly defined algorithm for evaluating the advancement towards satisfaction in some scenarios. We suppose the forest affected by the fire hosts a natural park that tourists visit throughout the year. Moreover, the neighborhood contains several shops and a secondary school. In case of an emergency, an Emergency Manager (EM) is in charge to undertake the required actions to fight/control the fire, and ensure the safety of people and properties.

Fig. 2 reports a goal model for the emergency management activity, mainly focusing on *CarePublicSafety*, whereas the grey part, *ExtinguishFire*, is up to the Fire Department and, therefore it, is out of the control of the EM. It is worth noting that some goals are annotated with temporal constraints and a Value Threshold (VT) to facilitate reader's understanding at a glance. Commonly, the definition and analysis of such a goal model is done in collaboration by analysts, who know about goal modelling and reasoning, and domain experts (likely, in this case, from some civil protection agency [8]). The *SecureNeighbourhood* goal takes care of advising the population to leave the area and forbidding access to cars and people coming from outside. It is the disjunction of two alternative sub-goals, *SoundAlarm* and *EstablishRoadBlocks*. The former devoted to advise people to leave by means of sirens purposefully deployed in the area. Alternately, the latter goal specifies neighbourhood may be secured by establishing roadblocks that could prevent people from entering the zone while the emergency is running and counting on the patrolling action of police cars to advise the population to leave. Roadblocks may be unmanned or manned, i.e, unmanned roadblocks are steel mobile fences deployed by employees of the traffic department, whereas manned roadblocks require a police officer to stay on the spot.

The *PatrolNeighbourhood* goal considers that despite deployed men and roadblocks, the area cannot be considered perfectly closed because a portion of forest partially bounds it; therefore, vigilance is to be enacted by police patrolling units that intercept tourists entering from the forest and, at the same time, warn the population to leave by using the car's loudspeaker.

The experimental evaluation. The evaluation focuses on the Emergency Manager's role, who may exploit a real-time simulation for training purposes. We suppose the simulation works on given input i.e. a precise series of events. The EM may affect the output by making some decisions that affect the final result. The role of the E2S is to estimate the goodness of resulting outcomes. If these are unsatisfactory, she will repeat the simulation by adopting different strategies.

For the sake of conciseness, we will omit the specification of the goal G and sub-plan P_G of the general formula E2S(G, P_G, t) when they are evident from the context, time t is, instead, often reported because significant to the discussion.

We set up two different simulation scenarios. Scenario 1 is obtained by enacting the first strategy developed by the EM and deploying an initial set of assets (police cars, buses,...). The EM uses this scenario to tune her strategy, and to better face the emergency. Results are affected by unexpected events like, for instance, the fact that fire will damage some sirens, thus reducing their effectiveness in warning the population. This has

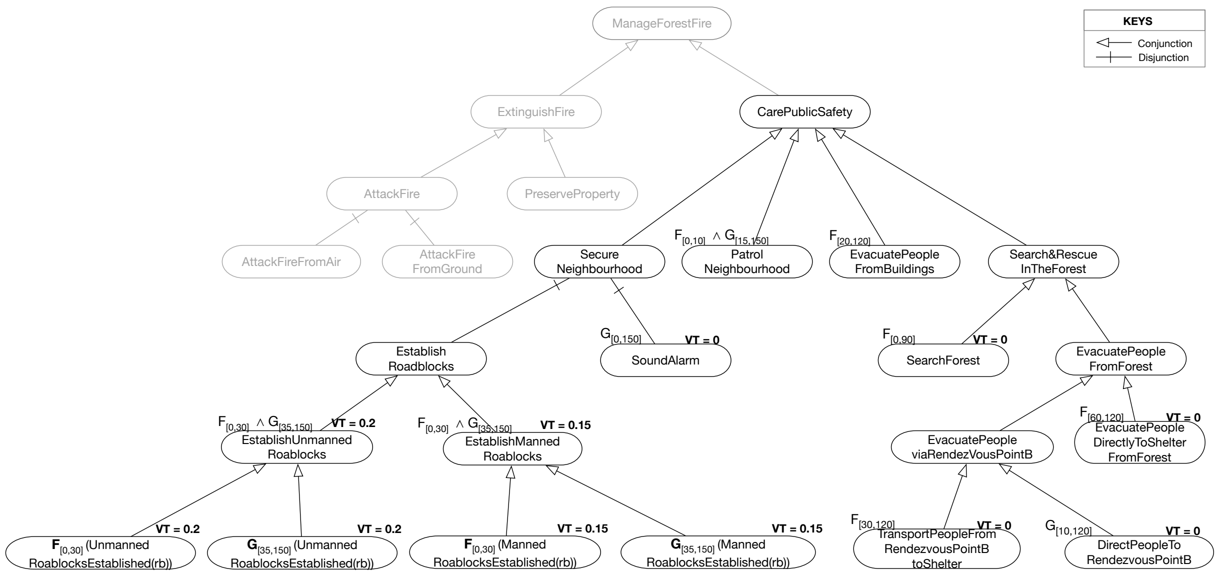

Fig. 2: The goal model for the forest fire case study.

a serious impact on the achievement of the goal *SoundAlarm*. As reported in the evaluation of the results, the first scenario fails in timely satisfying the goals.

Conversely, in Scenario 2 the EM takes corrective measures, and succeeds in satisfying the goal model in time.

During the simulation, the effectiveness of the EM's choices is calculated, at each time step (whose duration is 5 mins), by using the Algorithm 1. To simplify calculations, we suppose the effort for atomic goals is the same to the different types of goals: for achievement, denial goals it is $E_A = E_D = 100$, while for maintenance and prevention goals it is $E_M = E_P = 0.1$ at each time step.

It is worth noting that each scenario proposes two different strategies, one for each of the alternative solutions to the goal model descending from the disjunction in the decomposition of the goal *SecureNeighbourhood*.

We provide both algorithm and data for the experiment replication. Algorithm 1, implemented in Scala, is freely available for download[1]. The spreadsheet we used for carrying on the experiments, quantitative data and other details about the experiment are provided as Supplementary Data.

[1] https://github.com/icar-aose/GoalComplements

6.1 Scenario 1

The objective of this scenario is to highlight how the E2S metric may be useful in comparing different EM's decisions and resource allocation. In particular, this scenario focuses on showing that unexpected events are properly reported by timely variations of E2S, and this may also be observed in the final outcome. Fig. 3.a reports the $E2S(t)$ for the *CarePublicSafety* goal, that is the root goal for this scenario.

After the fire breaks out, at time=0, the EM estimates an initial asset allocation (2 traffic department trucks, 6 police cars, 14 alarm sirens, 2 buses with 50 seats, 1 bus with 40 seats, 3 ranger cars, 1 electric company team.)

Consequently, in [0,15], $E2S(t)$ goes down because actions are done for *SecureNeighbourhood*, *PatrolNeighbourhood*, and *SearchForest*: roadblocks are progressively deployed, police cars reach their patrol area, and part of the forest is searched; all of that causes the diminishing value of E2S for the corresponding goals.

Indeed, $E2S(t)$ significantly differ for the two strategies (sirens vs roadblocks). In fact, while the activation of sirens is instantaneous and the corresponding goal is already satisfied at t=0, deploying roadblocks takes more time, and therefore, the goal *SecureNeighbourhood* is slower to be addressed (higher values are at $t < 30$).

At t=20 goal *EvacuatePeopleFromBuildings* activates, causing a big increase in the overall effort to satisfaction: transport and evacuate people raises as a consequence of the greater number of persons that reach the rendezvous points.

At t=75 and t=85, two unexpected events: 1) the fire damages some sirens and 2) the wind pulls down a few fences.

In [60',85'], the effort required for *CarePublicSafety* reaches its maximum values (for both the alternative strategies), mainly due the activation of *EvacuatePeopleDirectlyToShelterFromForest* and for the damaged sirens. Notably, the effect of the wind that pulls down a few fences does not alter the satisfaction of the goal because of the VT=0.2, which allows for neglecting the absence of 2 fences.

At t=80', the EM reacts by dispatching a team from the electric company to fix the issue and sending another police car to patrol the area and warn the population. This creates a distinction between the two strategies and the need for the EM to evaluate a trade-off.

After t=75', the strategy using sirens, which was supposed to be better than the other for its performance at the beginning of the simulation, suddenly proposes a worse result since an effort is needed to restore the functionality of the sirens while the roadblock are only minimally affected by the wind and passing people that move some of them. Conversely, the goal *EstablishMannedRoadblocks* is negatively affected by the order issued by the EM of diverting policemen from the roadblocks to the activity of warning the population (while patrolling the neighbourhood).

The corrective actions adopted by the EM have some effect on the situation over time, and the values of E2S descend in the following time steps, but the simulations end with a failure in achieving the goals. In fact, at t=120 both goals *EvacuatePeopleFromBuildings* and *EvacuatePeopleDirectlyToShelter* fail. These goals are defined under a Final operator within t=120. They fail because there are still people inside the forest at t=120.

Results clearly report that the EM has not saved the entire population from fire in the simulated scenario, and a better solution strategy is to be found.

Just to appreciate the capability of the E2S metrics to evaluate partial goal satisfaction, we can consider the descending values of this parameter for the goal *EvacuatePeopleFromBuildings*. It starts its observation interval at t=20, and the $E2S(t)$ is at the maximum value (nobody saved from rendezvous point A). The E2S decreases once bus B1 loads 50 persons and brings them away at t=35. From then on, the E2S decreases each time another bus arrives at the rendezvous point and loads other persons. The $E2S(t)$ effectively shows the effort still to be done to complete the evacuation of people from the area. Similar behaviour is featured by the E2S for goals *EvacuatePeopleDirectlyToShelter* and *Search&RescueInTheForest*. They are significant because they deal with different issues (evacuating people vs searching the forest), but nonetheless, the E2S metric allows us to compare their progress.

Considering this simulation's (bad) results, the EM guesses she should adopt two approaches to improve the results: relaxing goal specifications and employing more assets. The next subsection describes a second attempt to handle the emergency.

6.2 Scenario 2

The EM focuses on the goals that showed the most significant issues in the previous attempt. First, the siren system is highly efficient and redundant so that it may hold a significant tolerance; for this reason, the EM adds a 0.5 VT to the *SoundAlarm* goal. Another issue was the *EvacuatePeople* goal which the E2S trend was correctly descending but too slow, leading to a failure within time limits. For this reason, the EM increases the employed assets by adding four more buses and two ranger cars to the rendezvous point (with a general grown-up in the cost of emergency management). She also reorganizes ranger shifts so that more cars are promptly available in case of an emergency. This generates a quicker response, thus ensuring the satisfaction of the Globally condition in goal *DirectPeopleToRendezVousPointB*.

Fig. 3.b reports the $E2S(t)$ for the *CarePublicSafety*, i.e., the root goal, for this scenario.

In [0',80'], the curves behave similarly to scenario 1. At t=85', only one police car is ordered to go and reinforce the patrol cars that are already warning the population to leave. At t=90', the

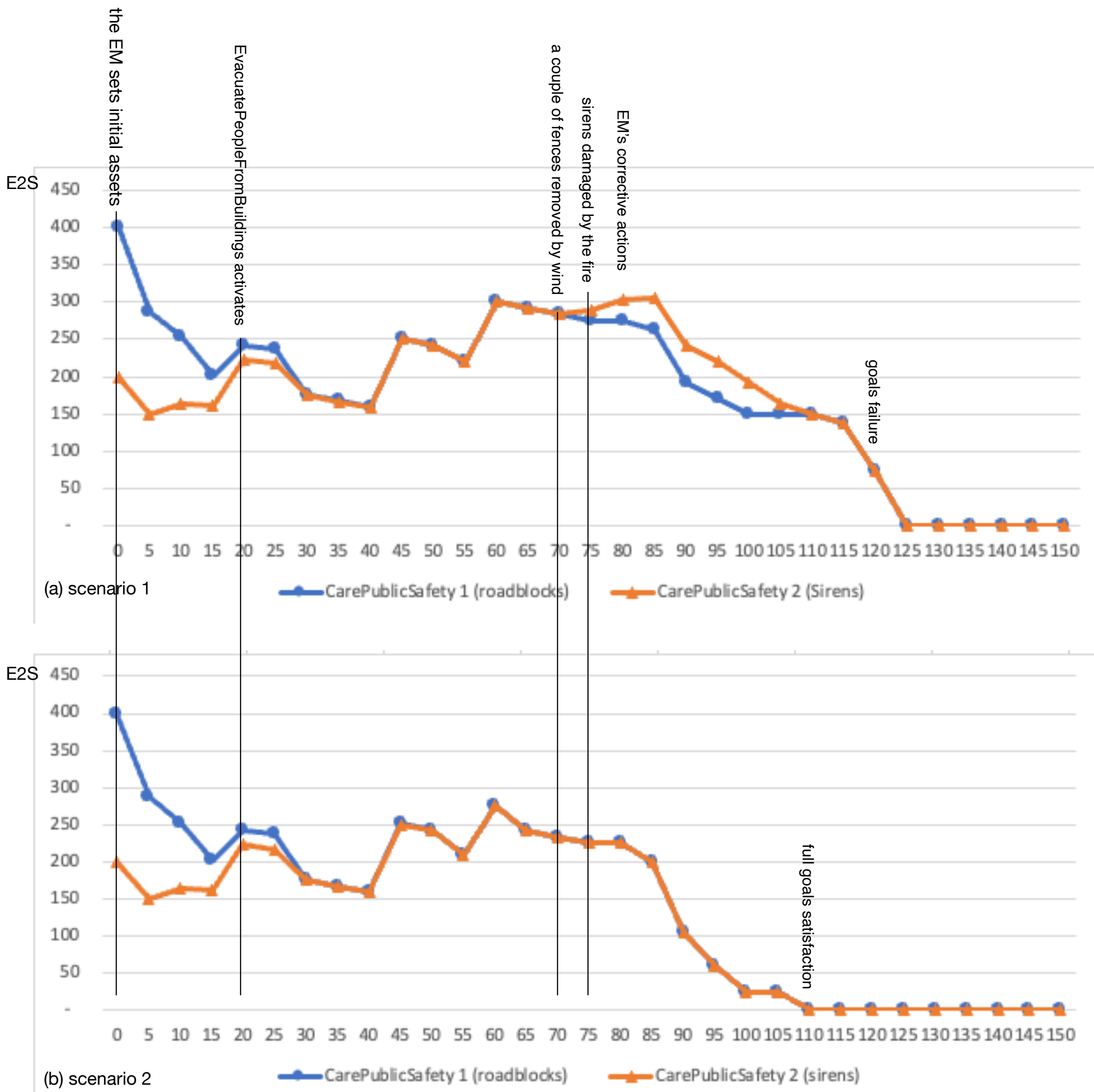

Fig. 3: Values of $E2S(t)$ for the goal CarePublicSafety for the two scenarios. Each graph reports the two alternative strategies.

availability of more ranger cars allows for a more effective rescue operation for those who cannot leave the forest independently.

As a consequence of this different set of orders, and the effectiveness of the greater number of employed assets, we can note that the value of $E2S(t)$ for the *CarePublicSafety* goal goes to zero before the end of the scenario. This is a dramatic difference from the previous case, and it depends on the fact that people are evacuated quicker by the greater number of buses, and the tourists in the forest are quickly rescued by the more numerous ranger cars.

The simulation proves that plan is now successful in saving all lives with a significant time margin (the root goal is satisfied at t=110'). Indeed other improvements may be studied regarding the velocity in bringing to safety the major number of persons. It is worth noting that although we consider the root goal satisfied already at t=110' (well before the end of crucial goals at t=120'), some residual effort is still to be spent because of the maintenance effort prescribed by the Globally conditions of other goals ending at t=150' (like the siren system). For instance, the analysis of the results of the previous two scenarios shows that both of them reach a significant distance from satisfaction at t=60'. This means that at half the duration of the observation interval, there is still a significant effort to be spent towards satisfaction. Further optimising the strategy to solution falls outside the scope of this example; we simply note that the process would be the same already sketched at the beginning of scenario 2.

In the next subsection, we will compare the results of the two scenarios to demonstrate the advantages of using E2S for evaluating them.

7 Results Evaluation and Discussion

In this section, we will evaluate the results provided by the previously introduced example, and we will use these results to deduce some information about the usefulness of the effort to satisfaction metric. Finally, we will discuss four potential application domains for the proposed approach.

7.1 Findings from the Experiment

Comparing and analysing the two scenarios of the Forest Fire example, we try to generalize how the E2S metric was useful in the real-time simulation.

The E2S allows to compare different strategies and appreciate the results of corrective actions in the flow of events. It proves to be a useful tool for deciding between alternative plans (according to their performance), and supporting a trade-off reasoning, if the case.

For instance, the differences between the two scenarios appear for $t > 70$, revealing that the first scenario is an upper bound for the second one. Moreover, for t∈[85,105], we can note that both the strategies (in scenario 2) show a rapidly descending curve; this is the consequence of the greater number of assets deployed in the second scenario. The EM correctly identified the problems delaying goal achievement in the first scenario, and the changes she made in the second attempt produced good results. The second scenario, in fact, ends with the successful achievement of the *CarePublicSafety* goal well before the deadline. We also note that the two strategies of the second scenario are not significantly different for $t > 70$, as the consequence of the EM's changes to the VT in the goal model. The comparison of the two curves represented in Fig. 3.b shows that the second strategy (using sirens) provides some significant advantage over the other in the initial part of the scenario. Moreover, achieving the final objective well before the prescribed deadline justifies the cost of employing more assets in the second scenario. All of that suggests the choice of implementing the siren strategy of the second scenario in case a real emergency arises in the studied environment. Such a strategy is more costly than the other studied in the first scenario, but it successfully achieves the goals well in time and shows a significant advantage over the other strategy (roadblocks) in the initial part of the emergency.

The E2S provides insights about the progress towards goal appraisal. The two Forest Fire scenarios highlight the usefulness of the proposed metric. The E2S provides the analyst significant run-time information about the progress towards the full goal satisfaction. So far, we considered E2S values computed within the context of some simulated scenarios, but they could also come from a real emergency during its actual development. Of course, in this case, the EM is not performing a run-time and real-time monitoring of the situation evolution and of the goodness of its choices. A growing value of E2S will warn the EM of a degrading situation that could need the employment of further resources to cope with the objectives in time successfully. Conversely, a constantly lowering value of E2S is an indicator of a situation that is developing positively. Of course, the E2S metric does not have any predictive feature; therefore, the trend could change at any time, but that would be reported by the new values, and again, they would become a warning sign for the EM.

The E2S alone is not enough to report risky situations. It is the case of sirens in scenario 1. Until fire broke the sirens, the system was performing very well, leaving the EM to think that sirens are the best solution to address the SecureNeighbourhood goal. They switch on quickly and ensure a low-effort solution, whereas using roadblocks and men is more expensive and time demanding. However, an external event highlights the fragility of using only sirens; when the goal fails, it is too late to recover with an alternative plan. Let us consider the values of $E2S_t(CarePublicSafety)$ over time as reported in the curves of Fig. 3. In both the scenarios, we can see that for t∈[0,30], the strategy using roadblocks has a curve that descends more slowly because the deployment of roadblocks requires time that is unnecessary to switch on sirens. This suggests that adopting the second strategy (sirens) is a better choice. The two strategies are equivalent in the time interval [30,70], but there are some significant differences after that. In the first scenario (Fig. 3.a) for t∈[75,115], the E2S of the second strategy (with sirens) raises because of the electricity blackout caused by fire and the consequent effort needed to restore the service. Indeed also roadblocks need some additional effort because wind and trespassing populations have removed some of them, but this has a minor effect on the overall result. In this portion of the first scenario, the first strategy performs better than the other. Anyway, both strategies fail at t=120 when they show some remaining E2S, although the observation interval of several goals is closing, and therefore they have not been achieved.

However, the E2S instrument can be used in combination with other techniques and tools to study the associated risk of a given plan. One example is executing many simulations and studying how the E2S changes according to external events. In the alternative, the analysts could couple the E2S with other specific analysis metrics that report on critical points and boundary behaviours. This second possibility is currently on our research agenda.

7.2 Application Domains and Guidelines

In this subsection, we will discuss some application domains where the proposed metric could be profitably applied, and we envisage some guidelines about how the E2S could be used in them.

7.2.1 Self-Adaptive Systems

We envisage a usage of E2S for the self-adaptive systems (SASs) domain. The trigger for adaptation may come from partial goal satisfaction, as an anticipation of new contextual conditions or even changes in user needs. In this context, the E2S metric can enhance the overall system awareness with knowledge of the 'cost' of adaptation. Typically, self-adaptive systems are provided with some engine for taking run-time decisions on choosing among different adaptation strategies. E2S can contribute precious information concerning the amount of remaining effort to satisfy a dangling goal. Moreover, it would help to compare alternative solutions to the problem-at-hand in terms of their effort deficit. In an earlier work [31], we offered a classification of self-adaptive systems with respect to their adaptation techniques. In that analysis, we focused on adaptations where the new behavior is defined on the fly, according to the context, available services and the goal-at-hand [30]. Here, an AI planning reasoner would compose behavior by assembling workflow services. The main problem with using a planner in this context is that heuristics must be handcrafted for each domain. In [29], we used a preliminary version of the E2S metric (actually, it was called R2S, Resistance-to-Satisfaction), representing a domain-independent metric for measuring the utility of a solution in the solution space. Given these initial results, we anticipate that the proposed metric could be extensively used in the context of self-adaptive systems.

7.2.2 AI Planning

AI planning is one of the best application domains for the proposed metrics. Different planners adopt approaches that may differ significantly, but all of them perform an exploration of the state space. Starting from an initial space, the problem consists in applying the available actions to visit new states, thus forming a tree of possible states whose root is the initial state itself. Each newly visited state is characterized by facts representing a partial (or full) satisfaction condition for the problem goals. Classically, the problem solution is the shortest path from the root to a state representing full goal satisfaction.

The proposed E2S metric perfectly fits the "Schemes for Guiding Search" concept introduced by Simon in his book about the Sciences of Artificial [35]. Let us suppose the problem consists of achieving a specific world state (the goal) while starting from an initial state and applying sequences of actions taken from a given set. Each sequence of action represents a potential path to the objective; let us also suppose that, in a specific situation, no path has still achieved the goal. The problem is choosing which path should be further extended to achieve the goal. Simon argues about the importance of having some 'value' that reports how promising a specific path is compared to the others. When possible, metrics coming from the application domain are used to evaluate each step, but such (informed) metrics are not always available. The idea is that the path exhibiting the lowest value of E2S should be the first to be further explored. This information would minimize possible solutions' search space and offer a reasonable design approach.

The E2S acts as a semi-informed metric, providing the possibility to estimate a promising path in the state space, even when rigorous domain-dependent measures could not be employed.

A prototypical concept of the E2S, the R2S presented in [9], already proved beneficial in such applications as demonstrated in [29].

Adopting such an approach needs a formalization of goals and their formulation in terms of predicates. Using domain-dependent functions for such predicates would ensure an improvement in the estimation quality, but even when that is not possible, common Boolean predicates may be used to evaluate what effort is still needed to satisfy the goal starting from the current state of the world.

7.2.3 Intelligent Agents

We envisage an application of the E2S in the reasoning of intelligent agents. Such agents are supposed to be reactive (they perceive their environment and react to changes in it), proactive (they exhibit a goal-directed behaviour and actively behave to address their objectives), social (they interact with other agents and use these interactions to pursue their goals) [41]. Often such agents are associated with an 'intentional stance' consisting of mental states such as beliefs, desires, and intentions (goals); they pursue their goals by enacting their capabilities according to some plan.

Unfortunately, the execution of such a plan may fail to satisfy all intended goals for many reasons, like a lack of resources, a lack of cooperation by other agents, or simply because environmental conditions operate against that. In a classical approach, in such a case, the agent should evaluate the opportunity to adopt corrective actions or to desist from pursuing that goal.

The process an agent enacts to deliberate if it wants to promote a desire (some state of the world it would potentially like to achieve) to an intention (a state of the world it will actively try to achieve) is usually addressed as practical reasoning. It is significant to consider how Bratman [5] defined that: "Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes."

This sounds very similar to how a human being deals with what she desires/values/cares about. Indeed, there is a significant nuance that we find very relevant in the reasoning human beings perform: humans also consider partial satisfaction for the above-cited 'options'. When a human being weighs conflicting options, (s)he also activates a background line of thoughts: what if I try to conciliate desire A with B by shading the achievement of both?

We believe agents, like humans, should also consider partial goal satisfaction for their 'options' while performing their practical reasoning process. In other words, intelligent agents should explicitly consider the possibility of a trade-off among different goals. Such trade-off would consist of accepting the partial satisfaction of some goal that could not be fully achieved because of the lack of resources, temporal constraints, norm/law violations, conflicts with other goals, etc.

Therefore, the deliberation phase in a trade-off reasoning becomes committing to a desire (goal) and its intended degree of satisfaction. This generates the need for an estimation metric, like the E2S one.

The E2S metric may have two different uses in the agent life: deliberation and plan execution. During the deliberation phase, as already said, E2S allows the agent to compare different solutions coming from the combination of different plans and resources. Solutions need to be compared to the 'distance' (the effort to satisfaction) they finally achieve from satisfying the agent's goals. During the execution of the selected plan, the agent can compare the expected state of the world with the attained one. This can be seen as an implementation of the perception loop introduced in [6] since, in this way, the agent continuously compares the expected behaviour with the real one, and when they differ, it can quickly enact corrective actions.

7.2.4 Requirements engineering

To make effort relevant to requirements engineering (RE), we need to argue that RE does not finish with requirements engineers deriving a specification from requirements and passing it on to developers. Rather, it continues as developers create and then execute a development plan. If things go wrong with the development plan, e.g., some functional/non-functional requirements can't be implemented on time or completely, then replanning needs to be done to consider ALL the alternatives in the initial goal model rather than the chosen specification. This constitutes an extension of RE into the development process territory. And, of course, the evaluation of alternatives has to be based on an E2S metric, as argued in this paper.

8 Conclusions and Future Works

Ihis paper proposes a novel metric, E2S, for estimating the degree of satisfaction (E2S) of a goal defined in terms of a goal model using bLTFOL. The metric measures the minimum effort required to bring about full satisfaction. The calculation of E2S for a goal model is performed by adopting an algorithm that evaluates alternative solutions to full satisfaction for a goal, given a goal model and a set of already satisfied sub-goals in the model. Calculation of E2S presupposes domain knowledge on the effort required to carry out domain actions, such as blocking off a road to traffic, or patrolling a road around the clock.

The availability of the E2S metric allows for a design/run-time assessment of how close a system is to fulfilling its requirements and what is the best strategy to full satisfaction, measured in terms of effort.

To offer more flexibility in applying the E2S to real world scenarios, we proposed an initial study on how to weaken goals of the goal model by relaxing their specification. We suggested introducing a value threshold that specifies some tolerance degree in the domain-dependent function that quantifies goal satisfaction.

The paper illustrates the application of our proposal with an example based on a forest fire case study adopted from the literature. In particular, we considered the problem of an emergency manager who has to plan the best management approach for a fire scenario. The availability of a metric dealing with partial goal satisfaction allows the comparison of different solution strategies, the run-time monitoring of the situation as the scenario unfolds, and the immediate appreciation of the consequences of selected choices.

In the future, we plan to study the possibility of merging the proposed approach with others where informed metrics are available; this would ensure a higher level of precision in estimating some properties related to goal satisfaction. We also plan to extend the work with the explicit consideration for different weights in the goals of the goal model, and, finally, we aim to identify other ways for relaxing goal constraints; for instance, including the relaxation of temporal limits or some kind of specific allowance for the achievement of the desired condition within a specific delay.

9 Acknowlegments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] Shaull Almagor, Udi Boker, and Orna Kupferman. 2013. Formalizing and reasoning about quality. In *International Colloquium on Automata, Languages, and Programming*. Springer, 15–27.

[2] Konstantinos Angelopoulos, Vítor E Silva Souza, and John Mylopoulos. 2014. Dealing with multiple failures in zanshin: a control-theoretic approach. In *Proceedings of the 9th*

International Symposium on Software Engineering for Adaptive and Self-Managing Systems. 165–174.

[3] Nelly Bencomo and Amel Belaggoun. 2013. Supporting decision-making for self-adaptive systems: from goal models to dynamic decision networks. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 221–236.

[4] Nelly Bencomo, Amel Belaggoun, and Valerie Issarny. 2013. Dynamic decision networks for decision-making in self-adaptive systems: a case study. In *2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 113–122.

[5] M.E. Bratman. 1990. What is intention? In *Intentions in Communication*, Morgan JL Cohen PR and Pollack ME (Eds.). MIT Press, Cambridge, MA.

[6] Antonio Chella and Irene Macaluso. 2009. The perception loop in CiceRobot, a museum guide robot. In *Neurocomputing*. Vol. 72(4-6). Elsevier.

[7] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos. 2000. *Non-Functional Requirements in Software Engineering*. Kluwer Publishing.

[8] Massimo Cossentino, Davide Guastella, Salvatore Lopes, Luca Sabatucci, and Mario Tripiciano. 2022. Linguistic and semantic layers for emergency plans. *Intelligenza Artificiale* 16, 1 (2022), 7–25.

[9] Massimo Cossentino, Luca Sabatucci, and Salvatore Lopes. 2018. Partial and full goal satisfaction in the musa middleware. In *Multi-Agent Systems. EUMAS 2018. Lecture Notes in Computer Science*, Vol. 11450. Springer, 15–29.

[10] Davide Dell'Anna, Fabiano Dalpiaz, and Mehdi Dastani. 2018. Validating goal models via Bayesian networks. In *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*. IEEE, 39–46.

[11] Davide Dell'Anna, Fabiano Dalpiaz, and Mehdi Dastani. 2019. Requirements-driven evolution of sociotechnical systems via probabilistic reasoning and hill climbing. *Automated Software Engineering* 26, 3 (2019), 513–557.

[12] Davide Dell'Anna, Mehdi Dastani, and Fabiano Dalpiaz. 2018. Runtime norm revision using Bayesian networks. In *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, 279–295.

[13] Paul J Feltovich, Jeffrey M Bradshaw, William J Clancey, Matthew Johnson, and Larry Bunch. 2007. Progress appraisal as a challenging element of coordination in human and machine joint activity. In *International Workshop on Engineering Societies in the Agents World*. Springer, 124–141.

[14] Li Feng-Lin, J. Horkoff, J. Mylopoulos, A. Borgida, G. Guizzardi, R. Guizzardi, and Liu Lin. 2014. Non-Functional Requirements as Qualities, with a Spice of Ontology. In *22nd IEEE International Requirements Engineering Conference (RE'14)*.

[15] Paolo Giorgini, John Mylopoulos, Eleonora Nicchiarelli, and Roberto Sebastiani. 2002. Reasoning with goal models. In *International Conference on Conceptual Modeling*. Springer, 167–181.

[16] Paolo Giorgini, John Mylopoulos, and Roberto Sebastiani. 2005. Goal-oriented requirements analysis and reasoning in the tropos methodology. *Engineering Applications of Artificial Intelligence* 18, 2 (2005), 159–171.

[17] Jennifer Horkoff and Eric Yu. 2013. Comparison and evaluation of goal-oriented satisfaction analysis techniques. *Requirements Engineering* 18, 3 (2013), 199–222.

[18] Ivan J Jureta, Alexander Borgida, Neil A Ernst, and John Mylopoulos. 2014. The requirements problem for adaptive systems.

ACM Transactions on Management Information Systems (TMIS) 5, 3 (2014), 1–33.

[19] Kangjin Kim and Georgios Fainekos. 2013. Minimal specification revision for weighted transition systems. In *2013 IEEE International Conference on Robotics and Automation*. IEEE, 4068–4074.

[20] Kangjin Kim and Georgios Fainekos. 2014. Revision of specification automata under quantitative preferences. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5339–5344.

[21] Morteza Lahijanian, Shaull Almagor, Dror Fried, Lydia Kavraki, and Moshe Vardi. 2015. This time the robot settles for a cost: A quantitative approach to temporal logic planning with partial satisfaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29 (1).

[22] Victor Lesser, Keith Decker, Thomas Wagner, Norman Carver, Alan Garvey, Bryan Horling, Daniel Neiman, Rodion Podorozhny, M Nagendra Prasad, Anita Raja, et al. 2004. Evolution of the GPGP/-TAEMS domain-independent coordination framework. *Autonomous agents and multi-agent systems* 9, 1-2 (2004), 87–143.

[23] Emmanuel Letier and Axel Van Lamsweerde. 2004. Reasoning about partial goal satisfaction for requirements and design engineering. In *ACM SIGSOFT Software Engineering Notes*, Vol. 29 (6). ACM, 53–62.

[24] Marvin L. Manheim. 1966. *Hierarchical Structure: A Model of Design and Planning Processes*. MIT Press, Cambridge, MA.

[25] J. Mylopoulos, L. Chung, and B. Nixon. 1992. Representing and Using Non-Functional Requirements: A Process-Oriented Approach. *IEEE Transactions on Software Engineering* 18(6) (1992), 483–497.

[26] Chi Mai Nguyen, Roberto Sebastiani, Paolo Giorgini, and John Mylopoulos. 2018. Multi-objective reasoning with constrained goal models. *Requirements Engineering* 23, 2 (2018), 189–225.

[27] Luis Hernan Garcia Paucar and Nelly Bencomo. 2019. Knowledge base k models to support trade-offs for self-adaptation using markov processes. In *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE, 11–16.

[28] Luis Hernan Garcia Paucar, Nelly Bencomo, and Kevin Kam Fung Yuen. 2019. Arrow: automatic runtime reappraisal of weights for self-adaptation. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 1584–1591.

[29] L. Sabatucci. 2020. MonteCarlo Tree Search with Goal-Based Heuristic. In *35th Workshop of the UK PLANNING AND SCHEDULING Special Interest Group*.

[30] Luca Sabatucci and Massimo Cossentino. 2015. From means-end analysis to proactive means-end reasoning. In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press, 2–12.

[31] Luca Sabatucci, Valeria Seidita, and Massimo Cossentino. 2018. The four types of self-adaptive systems: a metamodel. In *International Conference on Intelligent Interactive Multimedia Systems and Services*. Springer, 440–450.

[32] R. Sebastiani, P. Giorgini, and J. Mylopoulos. 2004. Simple and Minimum-Cost Satisfiability for Goal Models. In *International Conference on Advanced Information Systems Engineering (CAiSE'04)*.

[33] Steven Shapiro, Yves Lespérance, and Hector J Levesque. 2005. Goal change. In *IJCAI*. 582–588.

[34] Vítor E Silva Souza, Alexei Lapouchnian, William N Robinson, and John Mylopoulos. 2011. Awareness requirements for adaptive systems. In *Proceedings of the 6th international symposium on Software engineering for adaptive and self-managing systems*. 60–69.

[35] Herbert A. Simon. 2019. *The Sciences of the Artificial*. MIT press.

[36] John Thangarajah, James Harland, David N Morley, and Neil Yorke-Smith. 2014. Quantifying the completeness of goals in BDI agent systems. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*. IOS Press, 879–884.

[37] John Thangarajah, James Harland, and Neil Yorke-Smith. 2015. Estimating the progress of maintenance goals. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1645–1646.

[38] Jana Tumova, Gavin C Hall, Sertac Karaman, Emilio Frazzoli, and Daniela Rus. 2013. Least-violating control strategy synthesis with safety rules. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*. 1–10.

[39] M Birna Van Riemsdijk and Neil Yorke-Smith. 2010. Towards reasoning with partial goal satisfaction in intelligent agents. In *International Workshop on Programming Multi-Agent Systems*. Springer, 41–59.

[40] Maja Vukovic and Peter Robinson. 2005. Goalmorph: Partial goal satisfaction for flexible service composition. In *International Conference on Next Generation Web Services Practices (NWeSP'05)*. IEEE, 6–pp.

[41] M. Wooldridge. 2002. *An Introduction to MultiAgent Systems*. Wiley.

[42] Yi Zhou and XiaoPing Chen. 2004. Partial Implication Semantics for Desirable Propositions.. In *KR*. 606–612.

[43] Yi Zhou, Leendert Van Der Torre, and Yan Zhang. 2008. Partial goal satisfaction and goal change: weak and strong partial implication, logical properties, complexity. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 413–420.