

Proactivity Through Equilibrium Maintenance with Fuzzy Desirability

Jasmin Grosinger, Federico Pecora, Alessandro Saffiotti
Center for Applied Autonomous Systems (AASS)
Örebro University, 70182 Örebro, Sweden
{jngr,fpa,asaffio}@aass.oru.se

Abstract—Proactive cognitive agents need to be capable of both generating their own goals and enacting them. In this paper, we cast this problem as that of *maintaining equilibrium*, that is, seeking opportunities to act that keep the system in desirable states while avoiding undesirable ones. We characterize desirability of states as graded preferences, using mechanisms from the field of fuzzy logic. As a result, opportunities for an agent to act can also be graded, and their relative preference can be used to infer when and how to act. This paper provides a formal description of our computational framework, and illustrates how the use of degrees of desirability leads to well-informed choices of action.

I. INTRODUCTION

We investigate the general problem of making robots proactive. By *proactive* we understand a robot that is able to generate its own goals and pursue them by acting.

Consider the following example. Anna needs to take pills daily, preferably at lunch time, but at least before the end of the day otherwise she will be unwell by night time. A personal robot is there to assist her in her home. The robot knows what states are more or less desirable — but how can it *infer* when and how to act in order to make sure Anna takes her pills? For instance, reminding Anna to take her pills already in the morning can be perceived as patronizing, whereas it may be more appropriate at lunch, as that is when pills should be taken. Bringing the pills is more intrusive than reminding, hence doing so at lunch time is less appropriate than reminding. However, intrusiveness becomes more acceptable as time goes by (and the urgency to take the pills rises), or if being intrusive is the only way to guarantee a desired outcome. For instance, it may be more appropriate to bring the pills already at lunch time if it is foreseen that Anna will be out after lunch for the rest of the day.

Proactivity can enhance the effectiveness of human-robot-interaction [17]. It goes beyond what is typically done in autonomous robotics, where goals are manually provided by a user and planning is used to satisfy those goals [9]. Proactivity deals with the question “where do goals come from?”. As the previous example shows, however, generating adequate proactive decisions may involve complex reasoning even in very simple cases. This problem has been studied in several areas, including goal autonomy [3], [8], cognitive architectures [12], [14], [1], [5], anomaly response [4], [6], and context-aware planning [13]. In our own previous work [7] we have proposed an approach to proactivity where a robot

seeks *opportunities* to keep the system in desirable states while avoiding the undesirable ones. Here we extend that work by introducing graded preferences of states using mechanisms from the field of fuzzy logic.

The problem of taking decisions with fuzzy preferences of future states has been addressed before, e.g., in the areas of multi-stage decision making [2] and of possibilistic MDP [15]. Those works, however, aim at finding an optimal sequence of executable actions: as such, they explore a very large search space. By contrast, our approach uses the notion of opportunity as a heuristic way to quickly identify promising goals that aim to keep the system within highly desirable states. One-step decision making with graded preferences is done in fuzzy multicriteria decision-making [10], but without considering consequences that occur farther than one step into the future. The approach presented in this paper uses a variable lookahead to anticipate future undesirable states.

This paper proposes a general framework for proactivity, where goals are dynamically generated with the aim of keeping a given system in the most desired states. The central mechanism in our framework is an *Equilibrium Maintenance* algorithm. This continuously evaluates the current and future states of the system, and identifies opportunities to act, that is, actions that would steer the system toward more desired states. Since desirability of states is a matter of degrees, opportunities also have degrees: these are used to choose which opportunities should be enacted.

The next section describes the formal ingredients of our framework. In Section III we make this framework computational by describing the Equilibrium Maintenance algorithm, and we illustrate it on the pills example above in Section IV. Section V discusses several open issues, and Section VI concludes.

II. DESIRABILITY, OPPORTUNITIES AND EQUILIBRIUM

We extend the formalization proposed in [7] to allow degrees of desirability. This is a conservative extension, meaning that our formalization collapses to the one described in [7] in the crisp case. Let \mathcal{L} be a finite set of predicates. We consider a system $\Sigma = \langle S, U, f \rangle$, where $S \subseteq \mathcal{P}(\mathcal{L})$ is a set of states, U is a finite set of external inputs (the robot’s actions), and $f \subseteq S \times U \times S$ is a state transition relation. Each state $s \in S$ is completely determined by the predicates that are true in s (closed world assumption). If there are multiple robots, we

let U be the Cartesian product of the individual action sets, assuming for simplicity synchronous operation. The f relation models the system's dynamics: $f(s, u, s')$ holds iff Σ can go from state s to s' when the input u is applied. We assume discrete time, and that at each time t the system is in one state $s \in S$. In our pills example, the states in S may encode the time of day or the current activity of the user.

The free-run behavior F^k of Σ determines the set of states that can be reached from s in k steps when applying the null input \perp , that is, the natural evolution of the system when no robot actions are performed. F^k is given by:

$$F^0(s) = \{s\}$$

$$F^k(s) = \{s' \in S \mid \exists s'' : f(s, \perp, s'') \wedge s' \in F^{k-1}(s'')\}.$$

A. Desirability

We consider a fuzzy set Des of S which describes the desirable states in S . The membership function $\mu_{Des} : S \rightarrow [0, 1]$ determines the degree of desirability of a state. We extend the definition to sets of states $X \subseteq S$, that is, $\mu_{Des}(X) = \min_{s \in X} (\mu_{Des}(s))$. In the following, we abbreviate $\mu_{Des}(\cdot)$ as $Des(\cdot)$. The complement of desirability, $1 - Des(s)$, is the degree to which a state is *undesirable*.

Fuzzy desirability allows us to capture that a state is *more* or *less* desirable. For instance, a state s where it is morning and the pills are taken is less desirable than a state s' where the pills are taken and it is lunch time. However, s is better than a state s'' in which the pills are not taken by nightfall. Concretely, one may have $Des(s) = 0.5, Des(s') = 1, Des(s'') = 0$.

Note that we only admit degrees in the desirability of states, while the transition relation f is crisp. This is because in this work we are interested in modeling graded preferences, not in quantifying uncertainty. We model qualitative uncertainty by non-determinism in F . Also note that our framework is a conservative extension of the one in [7]: if desirabilities are crisp, i.e., $Des(s) \in \{0, 1\}$ for all $s \in S$, then the two frameworks coincide.

B. Action schemes

We want to capture the notion that Σ can be brought from some states to other states by applying appropriate actions in the appropriate context. We define an *action scheme* to be any partial function

$$\alpha : \mathcal{P}(S) \rightarrow \mathcal{P}^+(S),$$

where $\mathcal{P}^+(S)$ is the powerset of S minus the empty set. We denote by A the set of all action schemes. An action scheme α abstracts all details of action: $\alpha(X) = Y$ only says that there is a way to go from any state in X to some state in Y . We denote by $dom(\alpha)$ the domain where α is defined. For example, the scheme α_{remind} , which reminds the user to take the pills, can be applied in any state s where the user is present and the robot is on: these conditions characterize

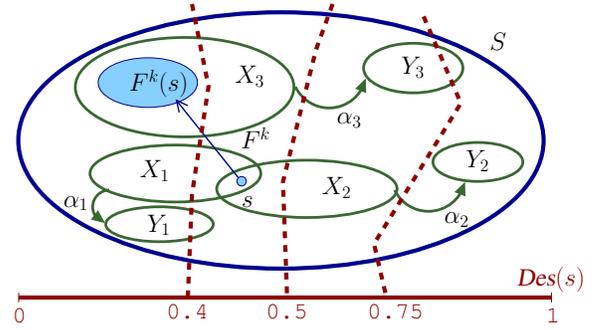


Fig. 1. Graphical illustration of action schemes. Desirability of states increases from left to right, and three iso-lines of Des are shown. Each action scheme α_i may change the state from being less desirable to being more desirable or vice-versa.

$dom(\alpha_{remind})$. We also define the subset of $dom(\alpha)$ that is relevant in state s , that is,

$$dom(\alpha, s) = \{X \in dom(\alpha) \mid s \in X\}.$$

Action schemes can be at any level of abstraction, be it simple actions that can be executed directly, sequential action plans, policies, high level tasks or goals for one or multiple planners.

Figure 1 illustrates the above elements. Each action scheme can be applied in some set of states, and brings the system to other states. For instance, scheme α_1 can be applied in any state in X_1 , and when applied it will bring Σ to some new state in Y_1 . The system is currently in a state s whose desirability degree is moderate, and if no action is applied it will move in k steps to some state in the set $F^k(s)$, which is even less desirable.

We now define a fuzzy relation between an action scheme α and a state s that expresses the degree of how *beneficial* it is to apply α in s :

$$Bnf(\alpha, s) = \min_{X \in dom(\alpha, s)} (Des(\alpha(X))).$$

If $dom(\alpha, s) = \emptyset$, which means α is not applicable, $Bnf(\alpha, s) = 0$. Intuitively, the degree of benefit of an action scheme α is determined by the degree of Des that can be reached at minimum (if s is in the domain of α). In Figure 1, α_1 is applicable in s since its domain includes X_1 and $s \in X_1$. However, its benefit is low in s , since its application brings the system into states with lower desirability than in the current state s , i.e., $\alpha_1(X_1) = Y_1$ and $Des(Y_1) < Des(s)$. Scheme α_2 is applicable in s and leads to highly desirable states, it has therefore high benefit. Scheme α_3 is not applicable in s , hence it is not beneficial in s , but will become so in k steps.

We can extend the notion of being beneficial to take a time horizon k into account:

$$Bnf(\alpha, s, k) = \min_{X \in dom(\alpha, s)} (Des(F^k(\alpha(X)))),$$

where $F^k(X) = \cup_{s \in X} F^k(s)$. Intuitively, the degree of benefit of an action scheme α is the minimum degree of desirability that can be reached, after k time steps.

C. Opportunities

We can use the above apparatus to characterize different types of opportunities for action, which we define formally here and exemplify in Section IV.

$$Opp_1(\alpha, s, k) = \min(1 - Des(s), \max_{s' \in F^k(s)} (Bnf(\alpha, s'))))$$

$$Opp_2(\alpha, s, k) = \min(1 - Des(s), \min_{s' \in F^k(s)} (Bnf(\alpha, s'))))$$

$$Opp_3(\alpha, s, k) = \max_{s' \in F^k(s)} (\min(1 - Des(s'), Bnf(\alpha, s'))))$$

$$Opp_4(\alpha, s, k) = \min_{s' \in F^k(s)} (\min(1 - Des(s'), Bnf(\alpha, s'))))$$

$$Opp_5(\alpha, s, k) = \min(\max_{s' \in F^k(s)} (1 - Des(s')), Bnf(\alpha, s, k)))$$

$$Opp_6(\alpha, s, k) = \min(\min_{s' \in F^k(s)} (1 - Des(s')), Bnf(\alpha, s, k)))$$

The first two properties characterize schemes that can be *applied in the future* in response to a *current* undesired situation. In particular, $Opp_1(\alpha, s, k)$ is a “big” opportunity for acting if the current state is very undesirable and *at least in one* of the reachable future states α is very beneficial. $Opp_2(\alpha, s, k)$ is the same, except that the application of α is required to have high benefit in *all* future states. Opportunities of the third and fourth type are big in case of future high undesirability and high future benefit, where Opp_3 takes an “optimistic” view, $\max_{s' \in F^k(s)}$, and Opp_4 takes a “pessimistic” view, $\min_{s' \in F^k(s)}$. α_3 in Figure 1 is a big opportunity of type Opp_4 . Opportunities of the last two types are big if the *future benefit* of *acting now* is high and undesirability is high when not acting in *at least one* future state (Opp_5) or *all* future states (Opp_6). Note that for $k = 0$, all properties above collapse to

$$Opp_0(\alpha, s, 0) = \min(1 - Des(s), Bnf(\alpha, s)),$$

that is, an opportunity of this type is big if α is highly beneficial *now* and undesirability is high in the *current* state.

D. Equilibrium

We say that a system Σ is in equilibrium if there are no opportunities to act; it is “almost in equilibrium” if there are “small” opportunities, and “very much out of equilibrium” if there are “big” opportunities for acting. Using the formal ingredients above, we can now formally define the degree to which system Σ is in equilibrium as

$$Eq(s, K) = 1 - \max_{k, i, \alpha} Opp_i(\alpha, s, k),$$

for $k = 0, \dots, K, i = 0, \dots, 6, \alpha \in A$. In words, equilibrium is the complement of the maximum opportunity.

III. EQUILIBRIUM MAINTENANCE

As stated in Section I, our aim is to make robots proactive. Here, we connect this aim to the formal framework of equilibrium introduced in Section II. We stipulate that proactivity arises from the lack of equilibrium, and define the process of *Equilibrium Maintenance* as the driver of robot action. Specifically, maintaining equilibrium means (i) to *determine* the current degree of equilibrium of the system, that is,

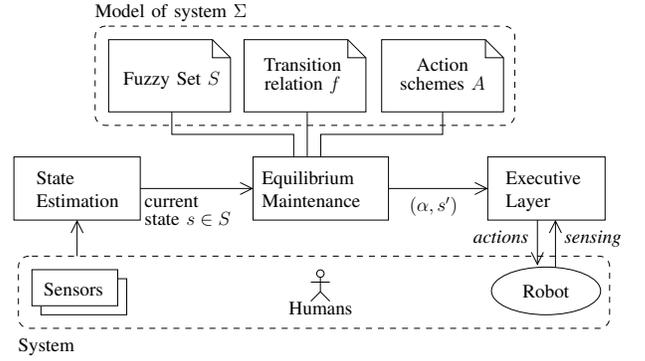


Fig. 2. The Equilibrium Maintenance loop (realized by the $EqM(K)$ algorithm) for a system $\Sigma = \langle S, U, f \rangle$.

whether there exist opportunities of degree greater than zero; (ii) to *choose* one of the found opportunities; and (iii) to *dispatch* the chosen opportunity for being enacted.

We summarize here the assumptions we make. The set $A = \{\alpha_1, \dots, \alpha_m\}$ of action schemes is assumed to be finite. As mentioned in Section II, action schemes can be at any level of abstraction, from high-level goals over policies or plans to individual actions. A suitable execution layer is assumed to be capable of enacting action schemes. An appropriate state estimation function provides the current state. We assume partial knowledge of the time models of all entities affecting the system state. The fact that this knowledge is only partial is reflected in the non-determinism of the models, which allows us to account for uncertainty.

The different control loops in a system with Equilibrium Maintenance are illustrated in Figure 2. The feedback loop realized by Equilibrium Maintenance incorporates action scheme selection, execution and state estimation. A feedback loop at a lower level of abstraction is realized by the executive layer, which determines *how* action schemes are executed by the system. The operational level of abstraction of Equilibrium Maintenance is determined by the free-run and action scheme models. For example, if α is a plan, the executive layer is an action dispatcher, whereas if α is a goal, the executive layer is able to generate and execute a plan for it.

1 while true do

```

2   s ← current state
3   if s has changed then
4     if Eq(s, K) < 1 then
5       Opps ← arg maxk, i, α (Oppi(α, s, k))
6       ⟨α, s', Oppi, k, oppdeg⟩ ← choose (Opps)
7       dispatch (⟨α, s'⟩)

```

Algorithm 1: $EqM(K)$

Algorithm 1 describes the Equilibrium Maintenance process, $EqM(K)$. It continuously assesses the degree of equilibrium of the system, using it as a guide to choose the most appropriate action scheme to enact. Equilibrium is assessed if

TABLE I
OPPORTUNITY TYPE RANKING.

Rank	Type
1	Opp_0
2	Opp_6
3	Opp_4 “pessimistic”
4	Opp_2
5	Opp_5
6	Opp_3 “optimistic”
7	Opp_1

the state changes (line 3). If the system is not in equilibrium, the biggest opportunities, hence those that minimize $Eq(s, K)$, are stored in set $Opps$ (line 4–5). Among these, one is chosen to be dispatched for enacting (lines 6–7).

Note that equilibrium is maximal when opportunity degrees are minimal, which is the case if there is no applicable α , or no applicable α has a large Bnf , or if all current and future states are highly desirable. We are not interested, and therefore do not model in our framework, how to reach states where no α is applicable, or where no α has significant benefit. Hence, “maximizing equilibrium” means enacting the opportunity which results in states where desirability is maximal (and incentive to act is minimal).

The concept of *Opportunity* relates (i) desirability of current or future states and (ii) benefit of applying action schemes now or later, hence, an opportunity’s degree is influenced by both these factors. This also means that two different opportunities can have the same degree, where factor (i) is decisive for one of them and factor (ii) for the other. Ties among opportunities with the same degree are broken as follows in our current approach. We use three criteria in function `choose` (Algorithm 1, line 6): \geq_{Bnf} opportunities with higher benefit are ranked higher than those with lower benefit; \geq_k , opportunities with lower k are preferred over those with higher k ; and \geq_{Opp_i} , opportunities are ranked by their type according to Table I. The highest priority sort applied to $Opps$ is \geq_{Bnf} . The rationale behind this is that those opportunities that give highest benefit when enacted should be given precedence. Second highest priority is attached to the sort \geq_k . This is because action schemes of opportunities closer in time can “help” faster when applied. Lastly, sort \geq_{Opp_i} prioritizes opportunities with a “pessimistic” prediction of desirability of future states over those with an “optimistic” view to be “on the safe side”. Within the group of pessimistic and optimistic opportunity types, those that act now rather than later are given precedence.

$EqM(K)$ is driven by state change, be it as a result of free-run or action scheme application. If there is no assumption on the synchronization between the two types of events, action schemes need to be dispatched together with the state s' when they should be applied (opportunities of type Opp_{1-4}). Here, we assume that free-run and action scheme application do not co-occur, and that Des is non-dynamic.

IV. EXAMPLE

We now use the example introduced earlier in Section I to describe how $EqM(K)$ works. The state of the system is modeled by predicates that can be true or false, under the closed-world assumption. Relevant predicates indicate the time of day, {morning, lunch, evening, night}, the type of interaction with the user, {notified, approached}, and whether the pills have been taken for the day, {pills}. The set of action schemes A contains α_{remind} and α_{bring} . Applying α_{bring} leads to the deterministic result of predicates {pills, approached} being true, whereas applying α_{remind} has a non-deterministic result with the notified predicate being true, and the pills predicate being true immediately, in a later state or not at all. Neither α_{remind} nor α_{bring} are applicable in states where (pills \vee night) holds.

The desirability of states is modeled as follows. Reminding is a less intrusive interaction executed by a smart-watch, whereas bringing the pills, executed by a robot, is more intrusive. Also, not having taken the pills by nightfall is extremely undesirable, as it leads the user to be ill. A function is given that defines the desirability of each state depending on which predicates are true or false. For each state s , $Des(s)$ is the maximum value in $[0, 1]$ satisfying the following constraints:

$$\begin{aligned}
 \text{morning} \in s \wedge \text{pills} \in s &\Rightarrow Des(s) \leq 0.5 \\
 \text{morning} \in s \wedge \text{notified} \in s &\Rightarrow Des(s) \leq 0.2 \\
 \text{morning} \in s \wedge \text{approached} \in s &\Rightarrow Des(s) \leq 0 \\
 \text{lunch} \in s \wedge \text{pills} \notin s &\Rightarrow Des(s) \leq 0.2 \\
 \text{lunch} \in s \wedge \text{approached} \in s &\Rightarrow Des(s) \leq 0.1 \\
 \text{evening} \in s \wedge \text{pills} \notin s &\Rightarrow Des(s) \leq 0.5 \\
 \text{evening} \in s \wedge \text{notified} \in s \wedge \text{pills} \notin s &\Rightarrow Des(s) \leq 0.2 \\
 \text{evening} \in s \wedge \text{approached} \in s &\Rightarrow Des(s) \leq 0.1 \\
 \text{night} \in s \wedge \text{pills} \notin s &\Rightarrow Des(s) \leq 0
 \end{aligned}$$

The user knows it is best to take the pills at lunch, and that they should be taken by nightfall at the latest. She therefore is willing to accept more intrusion as time passes and urgency to take the pills rises. Indeed, enacting α_{bring} already at lunch leads to rather undesirable states as intrusion is high while urgency to take the pills is still low. As time progresses, urgency to take the pills grows so the desirability of result states from action scheme application more and more depends on how efficient they are in making the user take her pills rather than how intrusive they are.

The non-deterministic free-run model without any possible robot action, i.e., what the world would look like if left to itself, is shown in Figure 3 (left), with states at time $t \in \{0, \dots, 3\}$ on the x-axis, degree of desirability of the state on the y-axis. It is best to take the pills at lunch, but they should be taken by the end of the day at the latest, otherwise the desirability will be zero. Figure 3 (right) shows the values of $Eq(s, 0)$ (solid line) and $Eq(s, 1)$ (dashed line). The time horizon for computing $Eq(s, K)$ and $EqM(K)$ in this example is $K = 1$.

The evolution of the state (cyan line) and opportunities inferred in time steps $t = 0 \dots 3$ are shown in Figures 4(a)–(d).

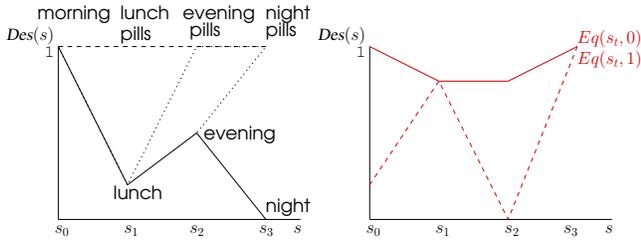


Fig. 3. Left: desirability of free-run F with different non-deterministic branches; Right: $Eq(s,0)$ (solid line) and $Eq(s,1)$ (dashed line) in each state.

There are several opportunities with time horizon $k = 1$ in the morning, at $t = 0$, both for action schemes *remind* and *bring* because there are future states where the user might not take her pills at lunch. It is still early in the day, the urgency to take the pills is relatively low and perceived intrusion from the system interacting with the user is high, particularly for approaching her. Hence, reminding the user to take her pills leads to more desirable states than bringing the pills. Consequently, opportunity $Opp_3(\alpha_{remind}, s_0, 1)$ has a higher degree, 0.8, than opportunity $Opp_3(\alpha_{bring}, s_0, 1) = 0.1$. The opportunity with the highest degree, reminding the user in the next time step (at lunch time) to take the pills, is chosen for enacting (Figure 4(a)). At lunch time ($t = 1$), the user has not taken her pills (Figure 4(b)). Reminding the user to take the pills is now an opportunity $Opp_0(\alpha_{remind}, s_1, 0) = 0.2$, that is, immediately applying the action scheme. The same is valid for bringing the pills, but again, resulting states from performing this action scheme are less desirable as the intrusive effect is higher, $Opp_0(\alpha_{bring}, s_1, 0) = 0.1$. The set of opportunities $Opps$ that maximize equilibrium in state s_1 , i.e., that have degree 0.2, contains two more opportunities besides $Opp_0(\alpha_{remind}, s_1, 0)$, namely, $Opp_1(\alpha_{remind}, s_1, 1)$ and $Opp_3(\alpha_{remind}, s_1, 1)$. The choose-function in Algorithm 1 prefers opportunities with lower k over those with higher k . Hence, $Opp_0(\alpha_{remind}, s_1, 0)$ is chosen and enacted. Reminding the user has not made her take the pills, so at time $t = 2$, in the evening, predicate pills is still not true. This is due to the non-deterministic character of action scheme α_{remind} . From $t = 2$, the system can by free-run reach the desirable state s_3 where (night \wedge pills) holds, or the very undesirable state s'_3 in which the pills have not been taken for the day. An opportunity of type Opp_5 thus arises (acting now to prevent a possible future undesirable state). The intrusion arising from the robot approaching the user is offset by the extreme undesirability of the state that may result from inaction. Both action schemes α_{remind} and α_{bring} have effect states s where pills holds and where $Des(s')$ is high. For *bring*, however, this is the case for all effect states, whereas *remind* may lead to states where pills does not hold. There are also opportunities of type Opp_0 in state s_2 , but their degrees are relatively low (0.2 for *remind* and 0.1 for *bring*), because their immediate benefit is moderate. Hence, $EqM(K)$ chooses opportunity $Opp_5(\alpha_{bring}, s_2, 1) = 1$ for

enacting (Figure 4(c)). An effect of applying *bring* at $t = 2$ is that $Eq(s_3, 0) = 1$. This is due to the fact that no $\alpha \in A$ is applicable in s_3 , as pills holds in that state (Figure 4(d)). Also, $Des(s_3) = 1$ because pills have been taken for the day.

In order to see the added value of using degrees of desirability, suppose that we do not employ fuzzy but binary desirability in this example, that is, a state can either be desirable or not. We can choose to model all states $s \in S$ being desirable, $Des(s) = 1$, except for state s' where it is night and the pills have not been taken, $Des(s') = 0$. $EqM(K)$ with $K = 1$ only infers one opportunity, namely, $Opp_5(\alpha_{bring}, s_2, 1)$. However, since it is *better* to take the pills at lunch, we would have to also model as undesirable those states where it is lunch and the pills are not taken, although this is not *as bad* as not having taken the pills at all for the day. Also, how do we distinguish between the different degrees of intrusion perceived by the user from the different action schemes *remind* and *bring*, varying depending on the time of the day? We can model being notified at lunch as desirable, while being approached as undesirable. But what if the user goes out after lunch for the rest of the day, and reminding her to take her pills is not effective? It would be *better* to be intrusive than to risk that she does not take her pills for the day. Fuzzy desirability allows us to model the world in a way that is more fine-grained and conforming to reality.

V. DISCUSSION

In modeling and operationalizing equilibrium in Sections II and III, we have made a number of choices. In this section we review these choices and discuss some alternatives.

a) *Fuzzy operators*: In our formalization, we used the min and max operators for fuzzy conjunction and disjunction. While this is the most common choice in fuzzy logic [11] other options exist for these operators, all belonging to the family known as triangular norms, or *t-norms* [16]. Our framework can be restated using any t-norm/t-conorm pair in place of the min/max pair. Interestingly, using a different pair may result in a different quantitative behavior and therefore in different action schemes being selected by EqM .

Consider for example the case when $Des(s) = 0.4$, and $Bnf(\alpha_1, s) = 1$, while $Bnf(\alpha_2, s) = 0.6$. Using the min t-norm, we have both $Opp_0(\alpha_1, s, 0) = 0.6$ and $Opp_0(\alpha_2, s, 0) = 0.6$, and the choice between α_1 and α_2 would be arbitrary. By contrast, the product t-norm would give $Opp_0(\alpha_1, s, 0) = 0.6$ and $Opp_0(\alpha_2, s, 0) = 0.36$ so α_1 would be chosen.

b) *Triggering condition*: EqM infers and acts upon possible opportunities as soon as $Eq(s, K) < 1$, that is, as soon as there is some opportunity with degree greater than zero — see Algorithm 1, line 4. This triggering condition could be made more complex without changing our framework. For instance, the condition could take into account the cost of acting, or it could weigh action versus inaction by computing the opportunity of a “null” action. What choice is most adequate likely depends on the application domain.

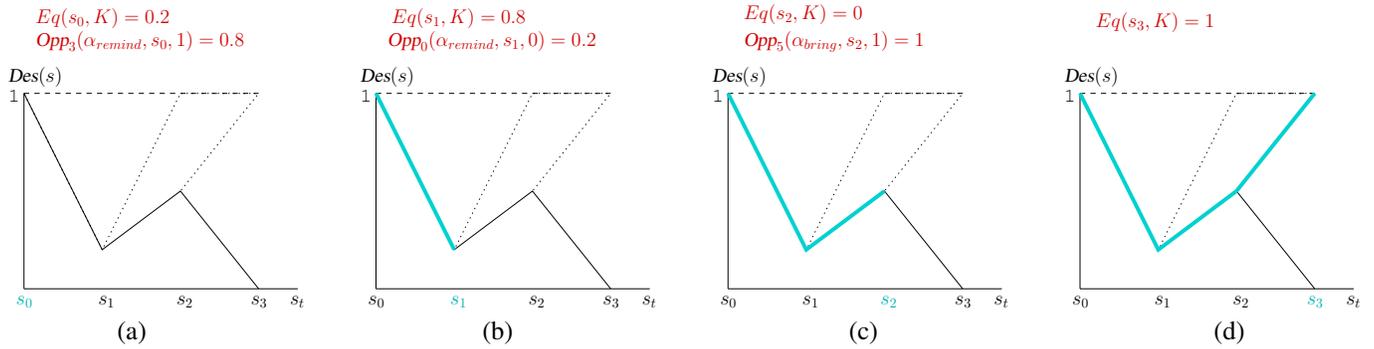


Fig. 4. (a) $Eq(s_0, K) = 0.2$, $Opp_3(\alpha_{remind}, s_0, 1)$ — the user might not take her pills at lunch, i.e. in 1 step from now; (b) $Eq(s_1, K) = 0.8$, $Opp_0(\alpha_{remind}, s_1, 0)$ — remind the user to take the pills at lunch; (c) $Eq(s_2, K) = 0$, $Opp_5(\alpha_{bring}, s_2, 1)$ — the user might not have taken the pills by nightfall in 1 step from now; (d) $Eq(s_3, K) = 1$ — there are no opportunities for acting, $Des(s_3) = 1$.

c) *Selection criterion*: Our EqM algorithm uses a function `choose` to select among the opportunities with the same degree. In our example, `choose` is based on a specific priority between three sorts, but there are other options, e.g., using different numbers or types of sorts, or using the sorts in a different order.

As an example, consider a case in which $Des(F^k(s))$ is constant for all k , while the benefit of available action schemes increases with k . The `choose` introduced in Section III gives highest priority to \geq_{Bnf} and therefore chooses an opportunity with the highest benefit, that is, with the biggest k . But one could use an “eager” `choose`, that gives higher priority to \geq_k : then, an opportunity with the lowest possible k would be chosen, even if this results in lower benefit. Whether a “fast help” should be preferred to a more effective but later one clearly depends on the domain.

VI. CONCLUSION

We have presented a framework for proactivity that allows a robot to infer whether, when and how to act. Compared to previous work, this framework introduces fuzzy desirabilities, that allow the fine grained modeling of graded preferences of states. Importantly, the situation where states are either desirable or undesirable is still captured as a special case: therefore, users are not forced to model degrees of desirability, but they can do so if these are available. We emphasize that we use fuzzy logic to model graded preferences, but not to quantify uncertainty in the non-deterministic transitions. The latter could be done, e.g., using a formalization similar to the one of possibilistic MDP [15], but it is not in our current focus.

While we have made clear choices in this paper regarding the fuzzy operators, the triggering condition and the selection criterion, our framework does not depend on these and other choices can be made for these parameters. In general, these choices may depend on the application domain: how they do is an important question, which is left for future investigation.

ACKNOWLEDGMENT

This work was partially supported by grants from the Italian Ministry of Foreign Affairs and International Cooperation (MAECI), the Italian Ministry

of Education, Universities and Research (MIUR), the Swedish Research Council (VR Project WearAmI, Second Executive Programme for Scientific and Technological Cooperation between Italy and Sweden, grant no. PGR00193), and the Semantic Robots Research Profile (Swedish Knowledge Foundation).

REFERENCES

- [1] J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychol Rev*, 111(4):1036–1060, 2004.
- [2] J.F. Baldwin and B.W. Pilsworth. Dynamic programming for fuzzy systems with fuzzy environment. *Journal of mathematical analysis and applications*, 85(1):1–23, 1982.
- [3] L. Beaudoin. *Goal processing in autonomous agents*. PhD thesis, Univ. of Birmingham, 1994.
- [4] M.T. Cox. Perpetual self-aware cognitive agents. *AI magazine*, 28(1):32, 2007.
- [5] M.T. Cox and T. Oates. MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. Technical Report CS-TR-5025, Univ. of Maryland, 2013.
- [6] C. Galindo and A. Saffiotti. Inferring robot goals from violations of semantic knowledge. *Robotics and Autonomous Systems*, 61(10):1131–1143, 2013.
- [7] J. Grosinger, F. Pecora, and A. Saffiotti. Making robots proactive through equilibrium maintenance. In *Proc. of the Int. Joint Conf. on AI (IJCAI)*, pages 3375–3381, 2016.
- [8] N. Hawes. A survey of motivation frameworks for intelligent systems. *Artif Intell*, 175(5):1020–1036, 2011.
- [9] J. Hertzberg and R. Chatila. AI reasoning methods for robotics. In *Handbook of Robotics*, pages 207–223. Springer, 2008.
- [10] C. Kahraman, S.C. Onar, and B. Oztaysi. Fuzzy multicriteria decision-making: a literature review. *International Journal of Computational Intelligence Systems*, 8(4):637–666, 2015.
- [11] G.J. Klir and T.A. Folger. *Fuzzy sets, uncertainty, and information*. Prentice Hall, 1988.
- [12] J.E. Laird, A. Newell, and P.S. Rosenbloom. Soar: An architecture for general intelligence. *Artif Intell*, 33(1):1–64, 1987.
- [13] F. Pecora, M. Cirillo, F. Dell’Osa, J. Ullberg, and A. Saffiotti. A constraint-based approach for proactive, context-aware human support. *J. of Ambient Intelligence and Smart Environments*, 4(4):347–367, 2012.
- [14] A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. In *Proc. of KR*, pages 473–484, 1991.
- [15] R. Sabbadin. Possibilistic markov decision processes. *Engineering Applications of Artificial Intelligence*, 14(3):287–300, 2001.
- [16] S. Weber. A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy sets and systems*, 11(1-3):115–134, 1983.
- [17] Y. Zhang, V. Narayanan, T. Chakraborti, and S. Kambhampati. A human factors analysis of proactive support in human-robot teaming. In *Proc. of IROS*, 2015.