Goal Formation through Interaction in the Situation Calculus: A Formal Account Grounded in Behavioral Science

Giuseppe De Giacomo
DIAG, Università degli Studi di Roma “La Sapienza”
Rome, Italy
degiacomo@diag.uniroma1.it

Yves Lespérance
EECS, York University
Toronto, ON, Canada
lesperan@eecs.yorku.ca

ABSTRACT

Goal reasoning has been attracting much attention in AI recently. Here, we consider how an agent changes its goals as a result of interaction with humans and peers. In particular, we draw upon a model developed in Behavioral Science, the *Elementary Pragmatic Model (EPM)*. We show how the EPM principles can be incorporated into a sophisticated theory of goal change based on the Situation Calculus. The resulting logical theory supports agents with a wide variety of relational styles, including some that we may consider irrational or creative. This lays the foundations for building autonomous agents that interact with humans in a rich and realistic way, as required by advanced Human-AI collaboration applications.

ACM Reference Format:
Giuseppe De Giacomo and Yves Lespérance. 2020. Goal Formation through Interaction in the Situation Calculus: A Formal Account Grounded in Behavioral Science. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Consider the following scenario: Barney is at home with Fred. Barney asks Fred what they should do. Fred answers by listing various options: play cards, watch a movie on a streaming service, have tea in the garden, or go out. Then Fred suggests to either play cards or watch a movie. Barney, on the other hand, proposes to either watch a movie or have tea in the garden. In the end they do neither of these things and actually go out.

If Barney and Fred were both human, then this would perhaps be a little unexpected, but not uncommon. But what if Fred was an artificial agent? If Fred was simply a virtual assistant, then this would indeed seem strange as such agents are normally subservient to their owner. But if Fred was instead a virtual companion, or a character in a game environment, then having such an option as a possible behavior would in fact be desirable, as it forms part of a rich believable human-like interaction. In this paper we examine this question, and develop a formal model of goal formation in interaction that can account for such behavior.

In recent years there has been growing interest in studying *goal reasoning*. In the words of [1], “intelligent systems may benefit from deliberating about, and changing their active goals when warranted. This flexibility may allow them to behave competently when they are not preencoded with a model that dictates what goals they should pursue in all encounterable situations.” These ideas have lead to the so called *goal-driven architectures* [8, 37, 40], where goals have a central role in determining the behavior of an intelligent system. In such systems/agents, the need to revise goals typically arises because of external changes in the situation the agent is acting in. Such a change of circumstances forces the agent to revise her goals to make them rationally compatible, if possible, with the new situation [9].

Besides external changes, another common reason to change goals is interaction with a human or peer. The typical case studied is that of an agent getting an order from outside. In this case, the agent would typically *adopt the new goal as ordered*, possibly *dropping* her current goals (i.e., acting according to an *acceptor* relational style in EPM, see later), or alternatively adapting her previous goal to the new order (i.e., acting according to a *sharer* relational style in EPM). More recently the notion of *agent rebellion* has been considered, where the agent actually refuses to adopt the new goal [2] (i.e., acting according to a *maintainer* relational style in EPM).

However there are applications in which we want the intelligent system/agent to act more like an ordinary person, not an obedient soldier or even a soldier with her own ethics. Humans make use of a much wider set of relational styles to revise their goals as a result of an interaction, including ways that we may consider irrational, perhaps because they inflame conflicts, or exhibit creativity.

In this paper, we take this point seriously and examine a rich psychological model, called the *Elementary Pragmatic Model (EPM)* [14, 18]. EPM is inspired by the work of Bateson [5], and was developed in Psychiatry as a tool for family therapy. An important feature of EPM is that its principles are formulated mathematically. Leveraging EPM, we take a radical departure from previous work in the area and instead consider a rich set of possible *goal/desire formation mechanisms* as a result of *interaction with others.* Specifically, we show that we can take a recent advanced theory of goals dynamics [27, 29] and integrate into it the rich EPM set of relational styles as a set of interaction-based goal change mechanisms.

Apart from our technical proposal, this paper shows that current theories of goal dynamics are ready to accomodate rich goal formation mechanisms such as the one considered here. (We discuss other such theories in the final section.) This is quite important in view of having intelligent agents interacting with us in a more human-like manner, *agents as companions rather than servants.* Such capabilities are crucial in advanced Human-AI collaboration applications for personal welfare, including AI-based digital assistants, e.g., realistic chatbots [20], or interactive entertainment and believable agents [4, 24, 42], as well as social welfare, including counseling/coaching applications, and automated facilitators for group interaction [52]. Moreover, moving away from a naive view of goal formation/adoption is becoming more and more important

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

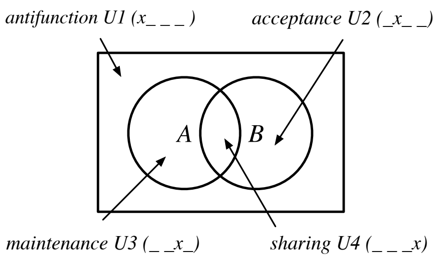

Figure 1: EPM coordinates

with the development of autonomy in AI, in particular to avoid the construction of Artificial Agents that act as fanatics even if mitigated by ethical principles, as recently advocated by Stuart Russell, who points out that agents should never be fanatically sure of the goals they are pursuing [45].

2 EPM AND RELATIONAL STYLES

In this paper we will follow a specific proposal: the Elementary Pragmatic Model (EPM), described mathematically in [48] and detailed with its clinical applications in [14, 18]. Like other models of the mind, EPM is a construction that promotes the study of psychological and psychopathological phenomena. EPM is rooted in the work of Gregory Bateson [5], which takes an interaction-based perspective on the human mind.

EPM is based on the idea that a subject's desires/goals change as a result of interaction: subject (i.e., her desire/goal) A changes to A' following interaction with the interlocutor (i.e,. her proposal) B. The results of such an interaction is described in terms of four "coordinates", depicted as regions in the Venn diagram in Figure 1:

- *sharing* coordinate U_4, ($\_\ \_\ \_\ x$), standing for $A \cap B$;
- *maintenance* coordinate U_3, ($\_\ \_\ x\_$), standing for $A \cap \bar{B}$;
- *acceptance* coordinate U_2, ($\_\ x\_\ \_$), standing for $\bar{A} \cap B$;
- *antifunction* coordinate U_1, ($x\_\ \_\ \_$), standing for $\bar{A} \cap \bar{B}$.

Each coordinate may be set to 0 or 1, according to whether the related set is included or not in the result of the interaction. For example 0101 stands for $(\bar{A} \cap B) \cup (A \cap B)$, i.e., for B. The combination of such four coordinates gives rise to sixteen functions, $F0, F1, \ldots, F15$, which are called "*relational styles*". To illustrate how the sixteen functions work, consider the diagram in Figure 1. It has four regions, corresponding to the coordinates. The combinations deriving from systematically filling in one, two, three, or four regions give rise to the sixteen possibilities, i.e., sixteen functions.

We illustrate the 16 relational styles with our "Barney and Fred" example from the introduction: Subject A (i.e., Fred) and interlocutor B (i.e., Barney) have the following options on what to do (for simplicity we assume these are the only options): "play cards", "watch a movie on a streaming service", "have tea in the garden", and "go out". Subject A expresses the desire to either "watch a movie" or "play cards". B instead suggests to A to either "watch a movie" or "have tea in the garden". We focus on A (i.e., Fred) and how his goals change through the interaction with B (i.e., Barney). The result of the interaction, based on each relational style is as described below:

F0 (0000) Void/Absent. *E.g.: Subject A decides they should not do anything.* No proposals, not even those which are shared with the other will be accepted. The subject is unable to establish any relationship.

F1 (0001) Sharer. *E.g.: Subject A decides they should "watch a movie".* Only the proposals shared by both subjects are accepted.

F2 (0010) Acceptor of one's own world only. *E.g.: Subject A decides they should "play cards".* Only the proposals of the subject himself which are not shared with the other remain after the interaction has taken place. He refuses any overlapping element.

F3 (0011) Maintainer of one's own world. *E.g.: Subject A keeps his own idea to either "watch a movie" or "play cards".* All the elements of the subject remain unaltered by the interaction. Every element is maintained independently of the other's proposal.

F4 (0100) Acceptor of the other's world without sharing. *E.g.: Subject A decides they should "have tea in the garden".* The subject accepts proposals of the other only if they are not his own.

F5 (0101) Acceptor of the other's world. *E.g.: Subject A changes his idea and is now willing to either "'watch a movie" or "have tea in the garden".* He substitutes for his own elements those of the other.

F6 (0110) Acceptor of one's own and of the other's world without sharing. *E.g.: Subject A is now willing to either "play cards" or "have tea in the garden".* The subject keeps his own elements, accepts the other's, but drops the shared ones.

F7 (0111) Acceptor of one's own and of the other's world. *E.g.: Subject A is now willing to either "watch a movie" , "play cards" or "have tea in the garden".* The subject is willing to accept the proposals of the other while maintaining his own.

F8 (1000) Acceptor of what only exists neither in one's own nor the other's world. *E.g.: Subject A now considers undesirable all proposed activities "watch a movie", "play cards", and "have tea in the garden", and elects to "go out" instead.*[1] The subject loses his own elements, refuses proposals of the other and takes only elements outside of the ring of interaction, i.e., of the worlds of the two subjects, making a "creative" choice.[2]

F9 (1001) Acceptor of what only exists or does not exist, in one's own and in the other's world. *E.g.: Subject A now considers undesirable "playing cards" and "having tea in the garden" and would like to either "go out " or "watch a movie".* This is as in F8, except that the shared proposals by the interacting subjects are kept.

F10 (1010) Anti-other or "Mary-Mary quite contrary". *E.g.: Subject A now considers undesirable "watching a movie" and "having tea in the garden" and would like to either "go out" or "play cards" .* The subject systematically refuses the other's proposals. In doing so, he even refuses elements from his own world.

F11 (1011) Complete maintainer of one's own world, with tendencies to expand. *E.g.: Subject A maintains his interest in "watching a movie" and "playing cards", but now also considers"going out".* In absolutely conserving its own point of view the subject also incorporates elements from outside of the ring of interaction.

F12 (1100) "Pseudoaltruist". *E.g.: Subject A changes his mind and becomes interested in "having tea in the garden", but also in "going out".* He rejects his own world, even the elements shared with the

[1] Notice that A still wants to do something with B (the context of the interaction) but none of the activities that A and B had originally in mind. In this case, the only remaining option is to "go out", but in general A could deliberate further to decide among the remaining choices.

[2] The relational styles F8–F15, which involve the antifunction coordinate, have been shown to be related to human creativity and have been utilized in tools for creativity development, see e.g.,[14, 15, 17, 46].

other, and accepts everything else. He could seem very complying wrt. the other, but he is "hiding an F3" through its complement.

F13 (1101) Exaggerated acceptor who refuses solely what exists in one's own world. *E.g.: Subject A gets interested in "watching a movie" and "having tea in the garden", plus the never proposed option to "go out".* He totally accepts the other's proposals as well as elements outside of the ring of interaction.

F14 (1110) Total acceptor who is nevertheless unable to share. *E.g.: Subject A changes his mind and gets interested in "playing cards" and "having tea in the garden", but now considers interesting also "going out"*. The subject avoids sharing at the cost of accepting elements outside of his own world.[3]

F15 (1111) Total acceptor. *E.g.: Subject A loses any particular preference and now considers doing any activity.* The subject will accept any proposals even those not formulated. He says "yes" to everything. His behavior (like F0) doesn't produce any information.[4]

Below, we use EPM to develop a rich descriptive (vs. normative) formal model of goal change as a result of interaction.

3 FORMAL PRELIMINARIES

Situation Calculus. Our base framework for modeling goal change is the situation calculus [36] as formalized in [44]. In this framework, a possible state of the domain is represented by a situation. There is a set of initial situations corresponding to the ways the agents believe the domain might be initially, i.e., situations in which no actions have yet occurred. Init(s) means that s is an initial situation. The actual initial state is represented by a special constant S_0. There is a distinguished binary function symbol do where $do(a, s)$ denotes the situation that results from performing action a in situation s. Thus the situations can be viewed as a set of trees, where the root of each tree is an initial situation and the edges represent actions ($s < s'$ means that s' is a successor of s). Relations (and functions) whose truth values vary from situation to situation, are called relational (functional, resp.) fluents, and are denoted by predicate (function, resp.) symbols taking a situation term as their last argument. There is a special predicate Poss(a, s) used to state that action a is executable in situation s.

We assume that we have a domain axiomatization that includes the following:[5] (1) action precondition axioms, one per action a characterizing Poss(a, s), (2) successor state axioms, one per fluent, that succinctly encode both effect and frame axioms and specify exactly when the fluent changes [44], (3) initial state axioms describing what is true initially including the mental states of the agents, (4) unique name axioms for actions, and (5) domain-independent foundational axioms Σ describing the structure of situations [33].

Knowledge. Following [39, 47], we model knowledge using a possible worlds account adapted to the situation calculus. $K(s', s)$ is used to denote that in situation s, the agent thinks that she could be in situation s'. Using K, the knowledge of an agent is defined as follows: $Know(\Phi, s) \stackrel{\text{def}}{=} \forall s'.K(s', s) \supset \Phi(s')$, i.e., the agent knows Φ in s if Φ holds in all of her K-accessible situations in s. Here and in the rest, Φ is a state formula, i.e., a formula with a single free situation variable, and $\Phi(s')$ is the formula that results from replacing free occurrences of this situation variable by s'. K is constrained to be reflexive, transitive, and Euclidean in the initial situation to capture the fact that agents' knowledge is true, and that agents have positive and negative introspection. As shown in [47], these constraints then continue to hold after any sequence of actions since they are preserved by the successor state axiom for K. We also assume that all actions are public, i.e., whenever an action (including exogenous events) occurs, the agent learns that it has happened. As in [27, 29], our framework supports knowledge expansion as a result of sensing actions [47] and some *informing* communicative actions.

Paths. Finally, to model general temporally extended properties (such as goals), we follow the approach of [27, 30], who extend the situation calculus with a new sort of *paths*, which are essentially infinite sequences of consecutive executable situations. We use variable p, possibly with annotations, to range over paths, and the special predicates $OnPath(p, s)$ (resp. $Starts(p, s)$) to state that situation s is on path p (resp. is the first situation on path p). These are axiomatized as in [27, 30].

Golog. To represent and reason about complex actions or processes obtained by executing atomic actions, *high-level programming languages* have been defined. Here we concentrate on Golog [34], which includes the following constructs:

$$\delta ::= \alpha \mid \varphi? \mid \delta_1;\delta_2 \mid \delta_1|\delta_2 \mid \pi x.\delta \mid \delta^*$$

In the above, α is an action term, possibly with parameters, and φ is situation-suppressed formula, that is, one with all situation arguments in fluents suppressed. As usual, we denote by $\varphi[s]$ the situation calculus formula obtained from φ by restoring the situation argument s into all fluents in φ. Program $\delta_1|\delta_2$ nondeterministically chooses between programs δ_1 and δ_2. Program $\pi z.\delta(z)$ nondeterministically "picks" an object d to bind to variable z and then executes program $\delta(z)$ with z assigned to d. Program δ^* performs δ zero or more times. We can define (**if** φ **then** δ_1 **else** δ_2) $\doteq$ $(\varphi?;\delta_1 \mid \neg\varphi?;\delta_2)$ and (**while** φ **do** δ) $\doteq$ $((\varphi?;\delta)^*;\neg\varphi?)$, e.g., **while** $\exists x.\neg OnTable(x)$ **do** $\pi z.\neg OnTable(z)?; table(z)$ repeatedly picks a block that is not on the table and tables it, until all blocks are on the table.

The semantics of Golog can be specified in terms of single-step transitions [13], using two predicates: *(i)* $Trans(\delta, s, \delta', s')$, which holds if one step of program δ in situation s may lead to situation s' with δ' remaining to be executed; and *(ii)* $Final(\delta, s)$, which holds if program δ may legally terminate in situation s. Using these predicates we can define whole computations as follows:

$$Do(\delta, s, s') \stackrel{\text{def}}{=} \exists \delta'.Trans^*(\delta, s, \delta', s') \wedge Final(\delta', s')$$

which says that by executing program δ in situation s we can get to situation s'. Also if we do not introduce concurrency, as here, we can define $Do(\delta, s, s')$ directly as in [34].

[3]Notice that this style of interaction has a *metaphoric flavor*: among the shared elements that are dropped one can consider all the concrete, everyday ones. What remains can describe concrete concepts only indirectly by drawing from either subjects' worlds or from outside them both (the presence of the antifunction coordinate adds an element of creativity) [14, 15, 17].

[4]F15 hints at a chaotic behavior, since everything is considered interesting; as Nietzsche said, "One must still have chaos in oneself to be able to give birth to a dancing star."

[5]We will be quantifying over formulae, and thus we also include axioms for encoding formulae and programs as first order terms, as in [13]; furthermore, we will also be using integers, and include an axiomatization of these as well.

4 A MODEL FOR GOAL DYNAMICS

Our model of goal change through interaction is based on Khan and Lespérance's (KL) situation calculus-based account of goals and their dynamics [27–29]. Note that KL handles temporally extended goals, since it incorporates a semantics based on infinite paths. Such semantics is analogous to that for Linear-time Temporal Logic (LTL) [43], however the representation of the state is not propositional as in LTL, but fully first-order as in the Situation Calculus.

Prioritized Goals. KL formalize desires/goals with different priorities, which they call *prioritized goals* (p-goals, henceforth). These p-goals are not required to be mutually consistent and need not be actively pursued by the agent. Each p-goal has its own accessibility relation/fluent G. A path p is G-accessible (i.e., desirable) at priority level n in situation s, denoted by $G(p, n, s)$, if all the goals of the agent at level n are satisfied over this path and if it starts with a situation that has the same history (in terms of the actions performed so far) as s. The latter requirement ensures that the agent's p-goal-accessible paths reflect the actions that have been performed so far. n ranges over natural numbers. A smaller n represents a higher priority, and the highest priority level is 0. Thus, it is assumed that the set of p-goals is totally ordered according to priority. One says that an agent has p-goal ϕ at level n in situation s if and only if ϕ holds over all paths that are G-accessible at n in s:

$$PGoal(\phi, n, s) \stackrel{\text{def}}{=} \forall p.\, G(p, n, s) \supset \phi(p).$$

Here and below, ϕ is a path formula, i.e., a formula with a single free path variable, and $\phi(p)$ is the formula that results from replacing free occurrences of this path variable by p.

Example. We can specify the initial p-goals of the agent in our running example as follows:

$$\begin{array}{l} Init(s) \supset (G(p, 0, s) \equiv \exists s'.Starts(p, s') \land Init(s')) \land \\ \quad (G(p, 1, s) \equiv \exists s'.Starts(p, s') \land Init(s') \land \exists s''.(s' \leq s'' \land \\ \quad\quad OnPath(p, s'') \land \exists c.Activity(c) \land Doing(c, s''))) \land \\ \quad (G(p, 2, s) \equiv \exists s'.Starts(p, s') \land Init(s') \land \exists s''.(s' \leq s'' \land \\ \quad\quad OnPath(p, s'') \land \exists c.Activity(c) \land Doing(c, s'') \land \\ \quad\quad\quad (c{=}WatchMovie \lor c{=}PlayCards))) \land \\ \quad ((n > 2) \supset (G(p, n, s) \equiv \exists s'.Starts(p, s') \land Init(s'))) \end{array}$$

i.e., at the highest priority (level 0), the agent wants to be in an initial situation, then at the next highest priority (level 1), he wants to eventually be doing some activity (*Doing* is an ordinary fluent),then at the next highest priority (level 2), he wants to eventually be watching a movie or playing cards, and then at all lower priority levels (> 2), he wants to be in an initial situation. It follows that:

$$\begin{array}{l} PGoal(\exists s'.Starts(p, s') \land Init(s')(p), 0, S_0) \land \\ PGoal(\exists s', s''.(Starts(p, s') \land s' \leq s'' \land OnPath(p, s'') \land \\ \quad\quad \exists c.Activity(c) \land Doing(c, s''))(p), 1, S_0) \land \\ PGoal(\exists s', s''.(Starts(p, s') \land s' \leq s'' \land OnPath(p, s'') \land \\ \quad \exists c.Activity(c) \land Doing(c, s'') \land \\ \quad\quad (c{=}WatchMovie \lor c{=}PlayCards)))(p), 2, S_0) \land \\ (\forall n.n > 2 \supset PGoal(\exists s'.Starts(p, s') \land Init(s')(p), n, S_0)) \quad \blacksquare \end{array}$$

Chosen Goals. In terms of the agent's p-goals, KL then define the agent's *chosen goals* or intentions (c-goals) that the agent is committed to and actually pursues. These are required to be consistent with each other and with the agent's knowledge, i.e., not ruled out by what is known. The agent's c-goals are essentially the largest set of highest priority "realistic" p-goals that are consistent, where a given p-goal is preferred over all lower priority p-goals.

First, KL define *realistic* p-goal accessible paths:

$$G_R(p, n, s) \stackrel{\text{def}}{=} G(p, n, s) \land \exists s''.(Starts(p, s') \land K(s', s)),$$

i.e., paths that are G-accessible at n in s and start with a situation that is K-accessible in s. Thus G_R prunes out from G the paths that are known to be impossible.

Then, KL define the c-goal accessibility relation over paths $G_\cap(p, s)$, such that the agent has the c-goal that ϕ in situation s, i.e., $CGoal(\phi, s)$, if ϕ holds over all of her $G_\cap$-accessible paths in s:

$$CGoal(\phi, s) \stackrel{\text{def}}{=} \forall p.G_\cap(p, s) \supset \phi(p),$$

$G_\cap(p, s)$ is in fact defined by induction on the priority level n, by first defining the paths that are in the maximal consistent set of highest priority "realistic" p-goals up to level n, $G_\cap(p, n, s)$, and then taking the c-goal accessible paths to be those for which $G_\cap(p, n, s)$ holds for all levels n, i.e., $G_\cap(p, s) \stackrel{\text{def}}{=} \forall n.G_\cap(p, n, s)$. $G_\cap(p, n, s)$ is axiomatized as follows:[6]

$$\begin{array}{l} G_\cap(p, n, s) \equiv \\ \quad \textbf{if } n{=}0 \textbf{ then} \\ \quad\quad \textbf{if } \exists p'.\ G_R(p', n, s) \textbf{ then } G_R(p, n, s) \\ \quad\quad \textbf{else } Starts(p, s') \land K(s', s) \\ \quad \textbf{else if } \exists p'.(G_R(p', n, s) \land G_\cap(p', n-1, s)) \\ \quad\quad \textbf{then } (G_R(p, n, s) \land G_\cap(p, n-1, s)) \\ \quad\quad \textbf{else } G_\cap(p, n-1, s). \end{array}$$

That is, at level $n = 0$, $G_\cap(p, n, s)$ contains the G_R accessible paths at level 0 if there exist such a path (i.e., the agent's c-goals at level 0 are his *RPGoals* at level 0 if her *PGoals* at level 0 are consistent with what she knows), otherwise it contains all paths that start with a K-accessible situation (i.e., her c-goals at level 0 are the trivial goal to be on a path where what she knows holds). At any level $n > 0$, $G_\cap(p, n, s)$ contains all the paths that are in $G_\cap$ at the previous level $n - 1$ and are G_R accessible at level n if there exists such paths (i.e., her c-goals at level n are her c-goals at level $n - 1$ plus her *RPGoals* at level n if the agent's *PGoal* at level n is consistent with what she know and her c-goals up to level $n - 1$), otherwise, it is simply the paths that are in $G_\cap$ at the previous level $n - 1$ (i.e., the *PGoal* at level n is left out of the agent's c-goals because it is inconsistent with the agent's knowledge and higher priority goals).[7]

Example (cont.) All the agent's initial p-goals are consistent, so he initially has the c-goal to (be in an initial situation and) eventually be watching a movie or playing cards:

$$\begin{array}{l} CGoal(\exists s', s''.(Starts(p, s') \land Init(s') \land s' \leq s'' \land \\ \quad OnPath(p, s'') \land \exists c.Activity(c) \land Doing(c, s'') \land \\ \quad\quad (c = WatchMovie \lor c = PlayCards)), S_0) \quad \blacksquare \end{array}$$

Subgoals. KL also account for the relationship between supergoals and subgoals. They take a p-goal ψ to be a *subgoal* of a p-goal ϕ in s if and only if ψ has lower priority than ϕ and ψ is also a

[6] **if** ϕ **then** ψ_1 **else** ψ_2 is an abbreviation for $(\phi \supset \psi_1) \land (\neg\phi \supset \psi_2)$.

[7] Note that paths in a p-goal (i.e. desired), unlike those in a c-goal, need not to be consistent with that the agent knows, i.e., start with a K-accessible situation. They must however have the correct action history. Thus (as in KL), they only need to be realistic wrt. the past action history, not the world state.

p-goal at all levels where ϕ is a p-goal:

$$\begin{array}{l} SubGoal(\psi, \phi, s) \stackrel{\text{def}}{=} \exists m.PGoal(\psi, m, s) \land \\ \quad \exists n.PGoal(\phi, n, s) \land \neg PGoal(\psi, n, s) \land \\ \quad\quad \forall m.PGoal(\psi, m, s) \supset PGoal(\phi, m, s) \land m > n \end{array}$$

In our example, the agent's p-goal to eventually be watching a movie or playing cards is a subgoal of that of eventually be doing some activity. As we will see below, this account guarantees several desirable properties of subgoal dynamics.

Basic Goal Dynamics. An agent's goals change when her knowledge changes as a result of the occurrence of an action (including exogenous events), or when she adopts or drops a goal. Here, we will mostly follow KL's formalization of basic goal dynamics, with one alteration: we will assume that every consistent (i.e., satisfied by some path) p-goal is a subgoal of another p-goal, where the "trivial" p-goal that the history of actions in the current situation has occurred is the root of the subgoal hierarchy (at priority 0). To ensure that this is the case, we require the initial state description to entail that $Init(s) \supset (G(p, 0, s) \equiv \exists s'.Starts(p, s') \land Init(s'))$, i.e., initially the root goal is the trivial goal to be in an initial situation. Note that the progression of this trivial goal will persist by the successor state axiom for G, see below. Given this assumption, it is sufficient to formalize two goal revision actions: $adopt(\psi, \phi, m)$, where the subgoal ψ is adopted relative to the parent goal ϕ at level m (which should be below the parent goal's level) and $drop(\phi)$, drop the goal ϕ.[8] Note that p-goals that are primary are simply adopted relative to the trivial p-goal at the root of the hierarchy. The *action precondition axioms for adopt and drop* are as follows:

$$\begin{array}{l} Poss(adopt(\psi, \phi, m), s) \equiv \exists n.PGoal(\phi, n, s) \land n < m, \\ Poss(drop(\phi), s) \equiv True \end{array}$$

i.e., the agent can adopt the subgoal that ψ w.r.t. the parent goal ϕ at level m in s if she already has the p-goal that ϕ at priority greater than m in s, and can always drop the p-goal that ϕ. The dynamics of p-goals is specified through the *successor state axiom for G*:

$$\begin{array}{l} G(p, n, do(a, s)) \equiv \forall \phi, \psi, m.(a \neq adopt(\psi, \phi, m) \land a \neq drop(\phi) \\ \quad \land Progressed(p, n, a, s)) \lor \\ \exists \phi, \psi.(a = adopt(\psi, \phi, m) \land SubGoalAdopted(p, n, m, a, s, \psi, \phi)) \\ \lor \exists \phi.(a = drop(\phi) \land Dropped(p, n, a, s, \phi)). \end{array}$$

Firstly, to handle the occurrence of a non-adopt/drop (i.e. regular) action a, one progresses all G-accessible paths to reflect the fact that this action has just happened; this is done using the $Progressed(p, n, a, s)$ construct, which replaces each G-accessible path p' with starting situation s', by its suffix p provided that it starts with $do(a, s')$:

$$\begin{array}{l} Progressed(p, n, a, s) \stackrel{\text{def}}{=} \\ \exists p', s'.G(p', n, s) \land Starts(p', s') \land Suffix(p, p', do(a, s')) \end{array}$$

$$\begin{array}{l} Suffix(p, p', s) \stackrel{\text{def}}{=} OnPath(p', s) \land Starts(p, s) \land \\ \quad \forall s'.s \leq s' \supset (OnPath(p, s') \equiv OnPath(p', s')) \end{array}$$

Any path over which the first action performed is not a is eliminated from the respective G accessibility level.

[8] In KL there are two adopt actions, $adopt(\phi, m)$ for adopting ϕ as a primary goal, i.e., not as a subgoal (where m is the level at which ϕ is adopted), and $adoptRelTo(\psi, \phi)$ for adopting a subgoal ψ w.r.t. a supergoal ϕ, (where ψ is always adopted at the level just below the parent goal ϕ). Our approach unifies these two forms of adoption.

When adopting a subgoal, one must capture the dependencies between a goal and the subgoals and plans adopted to achieve it. In particular, subgoals and plans adopted to bring about a goal should be dropped when the parent goal becomes impossible, is achieved, or is dropped. KL handle this as follows: adopting a subgoal ψ w.r.t. a parent goal ϕ adds a new p-goal that contains both this subgoal and this parent goal, i.e., $\psi \land \phi$, at a priority lower than that of the parent, shifting down all the ones below.[9] This ensures that when the parent goal is dropped, the subgoal is automatically dropped as well, since as we will see, when we drop the parent goal ϕ, we drop all the p-goals at all levels that imply ϕ including $\psi \land \phi$. Also, this means that dropping a subgoal does not necessarily drop the supergoal. This is formalized below:

$$\begin{array}{l} SubGoalAdopted(p, n, m, a, s, \psi, \phi) \stackrel{\text{def}}{=} \\ \quad \textbf{if } n < m \textbf{ then } Progressed(p, n, a, s) \\ \quad \textbf{else if } n = m \textbf{ then} \\ \quad\quad \exists k.highestLevel(\phi, s) = k \land \\ \quad\quad\quad Progressed(p, k, a, s) \land \psi(p) \\ \quad \textbf{else } (n > m) \; Progressed(p, n-1, a, s) \end{array}$$

$$\begin{array}{l} highestLevel(\phi, s) = k \stackrel{\text{def}}{=} \\ \quad PGoal(\phi, k, s) \land \forall \ell.\ell < k \supset \neg PGoal(\phi, \ell, s) \end{array}$$

$highestLevel(\phi, s)$ finds the highest level k where ϕ is a $PGoal$.

To handle dropping a p-goal ϕ, one replaces the propositions that imply the dropped goal in the agent's goal hierarchy by the "trivial" proposition that the history of actions in the current situation has occurred. Thus in addition to progressing all G-accessible paths, one adds back all paths that have the same history as $do(a, s)$ to the G-accessibility levels where the agent has the p-goal that ϕ:

$$\begin{array}{l} Dropped(p, n, a, s, \phi) \stackrel{\text{def}}{=} \\ \quad \textbf{if } PGoal(\phi, n, s) \textbf{ then} \\ \quad\quad \exists s'.Starts(p, s') \land SameHist(s', do(a, s)) \\ \quad \textbf{else } Progressed(p, n, a, s). \end{array}$$

Note also that procedural goal/subgoals can be handled by using Golog [34]: the goal to execute program δ now can be represented by the path formula $\exists s, s'.Starts(p, s) \land OnPath(p, s') \land Do(\delta, s, s')$.

Properties of Goal Dynamics. KL have shown several results about p-goal/c-goal dynamics. Let $\mathcal{D}_{KL}$ be the set of axioms and definitions in the KL theory of "optimizing agents" [27], i.e., the foundational axioms of the situation calculus with knowledge, the axiomatization of paths, the axioms encoding formulas and programs as terms, and the axioms specifying goals and their dynamics as outlined above. Proposition 4.4.15 in [27] states that $\mathcal{D}_{KL}$ entails that the agent no longer has the progression of a p-goal ϕ after dropping it, unless it is strongly inevitable:[10]

$$\begin{array}{l} \mathcal{D}_{KL} \models PGoal(\phi, n, s) \land \\ \quad \neg StronglyInevitable(ProgOf(\phi, drop(\phi), p), do(drop(\phi), s)) \\ \quad \supset \neg PGoal(ProgOf(\phi, drop(\phi), p), n, do(drop(\phi), s)) \\ \text{where } ProgOf(\phi, a, p) \stackrel{\text{def}}{=} \\ \quad\quad \exists p', s'.Starts(p', s') \land Suffix(p, p', do(a, s')) \land \phi(p') \end{array}$$

[9] KL assume that the subgoal is always adopted at the level immediately below that of the parent; we generalize this below.

[10] A condition ϕ is strongly inevitable in situation s iff ϕ holds for all paths that start in a situation with the same action history as s; see [27] for the formal definition. Also $ProgOf(\phi, a, p)$ means that the progression of ϕ holds after action a over path p.

This result still holds here as we have not changed how the G relation is affected by *drop*. KL also show that dropping a supergoal results in all its subgoals being dropped as well, but this only applies in the downward direction, i.e., dropping a subgoal does not cause its supergoals to be dropped. This holds here as well.

Regarding the effects of *adopt*, we can show some new results for our modified goal dynamics axiomatization. Let $\mathcal{D}^+_{KL}$ be $\mathcal{D}_{KL}$ with the successor state axiom for G and the precondition axiom for *adopt* replaced by the ones above. We can show that $\mathcal{D}^+_{KL}$ entails that the agent does have the p-goal that ψ at level m after adopting it as a subgoal of ϕ at that level (if executable):

Theorem 4.1.

$$\mathcal{D}^+_{KL} \models Poss(adopt(\psi, \phi, m), s) \supset PGoal(\psi, m, do(adopt(\psi, \phi, m), s))$$

Proof (sketch). We prove this similarly to Prop. 5.3.1 in [27]; there the subgoal ψ is always adopted at the level immediately below that of the supergoal ϕ, while here it is adopted at level m. The antecedent $Poss(adopt(\psi, \phi, m), s)$ ensures that ϕ is a p-goal in s at a level higher than m (and $highestLevel(\phi, s)$ is well defined). The result follows by the successor state axiom for G. □

We can also show that the adopted subgoal is a c-goal provided it is consistent with higher priority c-goals (and its parent is a c-goal):

Theorem 4.2.

$$\begin{aligned}\mathcal{D}^+_{KL} \models\ & \exists n.PGoal(\phi, n, s) \wedge n < m \wedge \exists p.G(p, n, s) \wedge G_\cap(p, n, s) \wedge \\ & highestLevel(\phi, s) = n \wedge \neg CGoal(\neg\exists p', s'.Starts(p, s') \wedge \\ & Suffix(p', p, do(adopt(\psi, \phi, m), s')) \wedge \phi(p) \wedge \psi(p'), m-1, s) \\ & \supset CGoal(\psi, m, do(adopt(\psi, \phi, m), s))\end{aligned}$$

Proof (sketch). We prove this similarly to Prop. 5.3.2 in [27]. The idea is as follows. All p-goals above level m will be progressed when $adopt(\psi, \phi, m)$ occurs. Since ψ (after the *adopt*) is consistent with c-goals above level m in s, all the p-goals that are c-goals are also consistent with ψ. We can also show that p-goals that are not c-goals up to level m remain so after after the *adopt*. Thus the p-goal levels that are c-goals up to level m remain the same after the *adopt*. ψ is added as a p-goal at level m by the adopt. It follows that some G-accessible paths from level m will be included in $G_\cap$ after the *adopt* and thus ψ is a c-goal at level m in $do(adopt(\psi, \phi, m), s)$. □

KL also prove some results about the persistence of achievement p-goals and c-goals under certain conditions.

Note that actions that affect the agent's knowledge such as sensing/informing actions also lead to changes in c-goals, as these are intentions that must remain consistent with what is known. P-goals on the other hand are just desires, and need not be consistent with what is known; they are only progressed when an action occurs,

5 GOAL CHANGE THROUGH INTERACTION

Let's formalize the changes in goals that occur due to an agent interacting with another depending on her relational style. We represent such an interaction as a complex action/program $interact(agt_2, \psi_1, \psi_2, F)$, where the subject with relational style F proposes her p-goal ψ_1 to the interlocutor agt_2 who instead proposes his p-goal ψ_2. We require that ψ_1 and ψ_2 be proper proposals:

$$\begin{aligned}& ProperProposal(\psi) \overset{\text{def}}{=} \\ & \quad \forall.a_1 = drop(\phi_1) \wedge a_2 = adopt(\phi_2, \phi_3, k) \supset \\ & \quad\quad (ProgOf(\psi, do(a_2, do(a_1, s)) \equiv ProgOf(\psi, do(a_1, s) \equiv \psi(s))\end{aligned}$$

This essentially means that the proposal ψ does not talk about the immediate dynamics (next 2 steps) of the agent's goals.[11] This complex action is defined as follows:

$$\begin{aligned}& interact(agt_2, \psi_1, \psi_2, F) \overset{\text{def}}{=} \\ & \quad \pi k.highestLevel(\psi_1) = k?; applyAttitude(F, \psi_1, \psi_2, k)\end{aligned}$$

$$\begin{aligned}& applyAttitude(F, \psi_1, \psi_2, k) \overset{\text{def}}{=} \\ & \quad \textbf{if } F \in \{F_1, F_2, F_3\} \textbf{ then } drop(False) \textbf{ else } drop(\psi_1); \\ & \quad \pi\phi.mostSpecCompatSuperGoal(\psi_1, \psi_2) = \phi?; \\ & \quad\quad adopt(map(F, \psi_1, \psi_2), \phi, k)\end{aligned}$$

$$\begin{aligned}& mostSpecCompatSuperGoal(\psi_1, \psi_2, s) = \phi \overset{\text{def}}{=} \\ & \quad \exists k.\forall p.\ G(p, k, s) \equiv \phi(p) \wedge \\ & \quad Subgoal(\psi_1, \phi, s) \wedge \neg PGoal(\neg\psi_2, k, s) \wedge \\ & \quad \forall k', \phi'.k < k' \wedge [\forall p.G(p, k', s) \equiv \phi'(p)] \supset \\ & \quad\quad \neg(Subgoal(\psi_1, \phi', s) \wedge \neg PGoal(\neg\psi_2, k', s))\end{aligned}$$

$$\begin{aligned}& map(F_j, \psi_1, \psi_2) \overset{\text{def}}{=} \\ & \quad (U_4(F_j) \wedge \psi_1 \wedge \psi_2) \vee (U_3(F_j) \wedge \psi_1 \wedge \neg\psi_2) \vee \\ & \quad (U_2(F_j) \wedge \neg\psi_1 \wedge \psi_2) \vee (U_1(F_j) \wedge \neg\psi_1 \wedge \neg\psi_2)\end{aligned}$$

Essentially, $interact(agt_2, \psi_1, \psi_2, F)$ amounts to executing $applyAttitude(F, \psi_1, \psi_2, k)$, where the level of the revised subgoal k is the highest level where ψ_1 is a p-goal of the agent. $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to the agent dropping the current subgoal ψ_1 (unless $F \in \{F_1, F_2, F_3\}$ and the agent maintains subgoal ψ_1, in which case we do $drop(False)$, which has no effects), and then adopting at level k a boolean function of ψ_1 and ψ_2 that depends on F, $map(F, \psi_1, \psi_2)$, relative to supergoal ϕ, the most specific supergoal of ψ_1 that is compatible with ψ_2 in the situation (the functions U_i simply project the i-th coordinate of the relational style F). The different cases of $applyAttitude(F, \psi_1, \psi_2, k)$ according to $map(F, \psi_1, \psi_2)$ capture how the relational style affects how goals change as a result of the interaction. It is easy to see that executing $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to executing the following program:

$$\begin{aligned}& case(F)\ \{ \\ & \quad 0000 \quad drop(\psi_1); adopt(False, \phi, k) \\ & \quad 0001 \quad drop(False); adopt(\psi_1 \wedge \psi_2, \phi, k) \\ & \quad 0010 \quad drop(False); adopt(\psi_1 \wedge \neg\psi_2, \phi, k) \\ & \quad 0011 \quad drop(False); adopt(\psi_1, \phi, k) \\ & \quad 0100 \quad drop(\psi_1); adopt(\psi_2 \wedge \neg\psi_1, \phi, k) \\ & \quad 0101 \quad drop(\psi_1); adopt(\psi_2, \phi, k) \\ & \quad 0110 \quad drop(\psi_1); adopt((\psi_1 \vee \psi_2) \wedge \neg(\psi_1 \wedge \psi_2), \phi, k) \\ & \quad 0111 \quad drop(\psi_1); adopt(\psi_1 \vee \psi_2, \phi, k) \\ & \quad 1000 \quad drop(\psi_1); adopt(\neg\psi_1 \wedge \neg\psi_2, \phi, k) \\ & \quad 1001 \quad drop(\psi_1); adopt((\psi_1 \wedge \psi_2) \vee (\neg\psi_1 \wedge \neg\psi_2), \phi, k) \\ & \quad 1010 \quad drop(\psi_1); adopt(\neg\psi_2, \phi, k) \\ & \quad 1011 \quad drop(\psi_1); adopt(\psi_1 \vee \neg\psi_2), \phi, k) \\ & \quad 1100 \quad drop(\psi_1); adopt(\neg\psi_1, \phi, k) \\ & \quad 1101 \quad drop(\psi_1); adopt(\neg\psi_1 \vee \psi_2, \phi, k) \\ & \quad 1110 \quad drop(\psi_1); adopt(\neg\psi_1 \vee \neg\psi_2, \phi, k) \\ & \quad 1111 \quad drop(\psi_1); adopt(True, \phi, k)\ \}\end{aligned}$$

Let's discuss these different cases of $applyAttitude(F, \psi_1, \psi_2, k)$.

First, let's consider the cases where the agent maintains and possibly refines her goal ψ_1, i.e., $F1$, $F2$, and $F3$. When the agent's attitude is $F1$, she elects to share the interlocutor's goal, and thus we take $applyAttitude(F, \psi_1, \psi_2, k)$ to amount to $adopt(\psi_1 \wedge \psi_2, \phi, k)$,

[11] This requirement is not essential, but it greatly simplifies the definition of $interact$ and the statement of our theorems, as the proposals are not affected by progression over the goal change actions that implement $interact$.

i.e., adopt the conjunction of the interlocutor's goal ψ_2 with the agent's original goal ψ_1, at the same priority level as ψ_1. The parent of the new goal is the most specific supergoal of ψ_1 that is compatible with ψ_2.[12] Note that we do *drop*($False$), which has no effect, just for uniformity, so that in every case *applyAttitude* involves a *drop* followed by an *adopt*.

Example. $interact(agt_2, \psi_1, \psi_2, F1)$ with the subject proposing to eventually be watching a movie or playing cards and the interlocutor proposing to be watching a movie or having tea

$$\psi_1 = \exists s''.(Starts(p, s') \land Init(s') \land s' \leq s'' \land \exists c.Activity(c) \land Doing(c, s') \land (c{=}WatchMovie \lor c{=}PlayCards))$$

$$\psi_2 = \exists s''.(Starts(p, s') \land Init(s') \land s' \leq s'' \land \exists c.Activity(c) \land Doing(c, s') \land (c{=}WatchMovie \lor c{=}TeaInGarden))$$

amounts to performing $applyAttitude(F, \psi_1, \psi_2, k)$ with the most specific compatible supergoal

$$\phi = \exists s''.(Starts(p, s') \land Init(s') \land s' \leq s'' \land \exists c.Activity(c) \land Doing(c, s'))$$

and the level of the revised subgoal $k = 2$, which amounts to $adopt(\psi_1 \land \psi_2, \phi, k)$, i.e., the "shared" p-goal to eventually be watching a movie at level 2 replacing her original p-goal ψ_1. ■

When the agent's attitude is $F2$, she elects to maintain her goal while rejecting the interlocutor's goal, and thus we take $applyAttitude(F, \psi_1, \psi_2, k)$ to amount to $adopt(\phi_1 \land \neg\psi_2, \phi, k)$, i.e., adopt the conjunction of the negation of the interlocutor's goal ψ_2 with the agent's original goal ψ_1, at the same priority level as ψ_1, relative to the most specific compatible supergoal.

When the agent's attitude is $F3$, she elects to maintain her goal unchanged, without adopting the interlocutor's goal, and thus we take $applyAttitude(F, \psi_1, \psi_2, k)$ to amount to readopting her goal ψ_1; her goals remain essentially unchanged.

Secondly, let's consider the cases where the agent accepts the interlocutor's goal, i.e., $F4$, $F5$, $F6$, and $F7$. In these, unless ψ_1 implies ψ_2 the agent is relaxing her strict preference for ψ_1, adopting instead the interlocutor's goal possibly in conjunction with her own goal ψ_1. Thus for $F5$, where the agent accepts the interlocutor's goal without rejecting her own, $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to dropping ψ_1 and then adopting ψ_2 relative to the parent, at the same level as her original goal. For $F7$, the agent accepts the interlocutor's goal while maintaining hers, and $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to dropping ψ_1 and then adopting $\psi_1 \lor \psi_2$ relative to the parent, at the same level as her original goal. For $F6$, the agent accepts the interlocutor's goal while maintaining hers, but rejects both goals holding, and thus $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to dropping ψ_1 and then adopting $(\psi_1 \lor \psi_2) \land \neg(\psi_1 \land \psi_2)$ relative to the parent, at the same level as her original goal. Finally, for $F4$, the agent accepts the interlocutor's goal while rejecting hers, and $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to dropping ψ_1 and then adopting $\neg\psi_1 \land \psi_2$ relative to the parent, at the same level.

[12] This essentially replaces the subject's original goal ψ_1 by $\psi_1 \land \psi_2$. A reasonable alternative would be to do $adopt(\psi_2, \psi_1, k+1)$, i.e., adopt the interlocutor's goal ψ_2 relative to the subject's own goal ψ_1, at a priority level just below that of ψ_1. The difference in this case would be that the subject's would retain a higher priority for ψ_1 compared to $\psi_1 \land \psi_2$ and might fall back to the former after renouncing the latter.

Thirdly we have the so called anti-functions $F8$ to $F15$ where the agent is willing to accept paths that satisfy neither her original goal ψ_1 nor the interluctor's goal ψ_2. This introduces a creative element, giving rise to goals that were unforeseen before the interaction. For instance for $F8$, the agent decides to pursue paths where neither her original goal ψ_1 nor the interluctor's goal ψ_2, and $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to dropping ψ_1 and then adopting $\neg\psi_1 \land \neg\psi_2$ relative to the parent, at the same level as her original goal. For our example, this amounts to adopting the goal of eventually going out (the domain theory includes unique names and domain closure axioms for the 4 activities). For $F9$, the agent decides to pursue paths where neither her original goal ψ_1 nor the interluctor's goal ψ_2 as well as paths where both goals hold, and $applyAttitude(F, \psi_1, \psi_2, k)$ amounts to dropping ψ_1 and then adopting $(\neg\psi_1 \land \neg\psi_2) \lor (\psi_1 \land \psi_2)$ relative to the parent, at the same level as her original goal. Notice that in our formalization, although the original goal is dropped and replaced by some combination that includes $\neg\psi_1$ and $\neg\psi_2$, such a combination is still a subgoal of the most specific compatible supergoal of ψ_1 and ψ_2. In other words the context of the interaction remains unchanged. This gives the context for the creative element, avoiding a disruptive reconsideration of unrelated goals. Note also that in many of theses cases, the subject has merely rulled out certain options (e.g., for $F14$, rulled out $\phi_1 \land \psi_2$) and must later decide how she wants to realize the parent goal ϕ under these constraints.

Note that case $F0$, i.e., 0000, and case $F15$, i.e., 1111, superficially look alike. In both, we are dopping the original goal remaining with the supergoal. However in $F0$ the level k simply disappears while in $F15$ the super goal is readopted at level k. Thus in $F0$ the agent does not have any goal at level k while in $F15$ it has the context as goal at level k but without committing to any means to achieve it.

Let us show formally that *interact* has the right effects on the subject's p-goals in all of these cases. Let $\mathcal{D}_{GFI}$ be $\mathcal{D}^+_{KL}$ augmented with the axiomatization of Golog in [13] and the axioms and definitions for goal formation through interaction presented in this section. First, we can show (using Theorem 4.1) that in all cases, after the interaction the agent has adopted the changed p-goal $map(F, \psi_1, \psi_2)$ obtained by applying the agent's relational style to the two proposals:

Theorem 5.1. *For any* $j = 0, \ldots, 15$

$$\mathcal{D}_{GFI} \models highestLevel(\psi_1, s){=}k \land Do(interact(agt_2, \psi_1, \psi_2, F_j), s, do(a_2, do(a_1, s))) \supset PGoal(map(F_j, \psi_1, \psi_2), k, do(a_2, do(a_1, s)))$$

We can also show that in all cases but $F1$, $F2$, and $F3$, the agent has dropped her proposed goal ψ_1 after the interaction unless it is strongly inevitable:

Theorem 5.2. *For any* $j = 0, 4, \ldots, 15$

$$\mathcal{D}_{GFI} \models highestLevel(\psi_1, s){=}k \land k' \neq k \land \neg StronglyInevitable(\psi_1, do(a_1, s)) \land Do(interact(agt_2, \psi_1, \psi_2, F_j), s, do(a_2, do(a_1, s))) \supset \neg PGoal(\psi_1, k', do(a_2, do(a_1, s)))$$

Proof (sketch). We have that $Poss(drop(\psi_1), s)$ since $interact(agt_2, \psi_1, \psi_2, F_j)$ is executable in s. Then by Prop. 4.4.15 in [27] (discussed earlier in Sec. 4) and ψ_1 being a proper proposal, it follows that $\neg PGoal(\psi_1, k', do(a_1, s))$ for all k'. In the case where $k' < k$, by the successor state axiom for G, the G-accessible paths at level k' in $do(a_2, do(a_1, s))$ are simply progressions over a_2 of G-accessible paths in $do(a_1, s)$ at the same level. Thus $\neg PGoal(\psi_1, k', do(a_2, do(a_1, s)))$. The case for $k' > k$ is similar, but the set of paths is shifted down one level. □

This holds for all levels except that of the new goal replacing ψ_1; to show it for level k, we need additional conditions on ψ_2. Also, we can show (by Theorem 4.2) that afterwards, $map(F, \psi_1, \psi_2)$ is a c-goal as well, if it is consistent with higher priority c-goals:

Theorem 5.3. *For any* $j = 0, \ldots, 15$

$$\begin{aligned}
\mathcal{D}_{GFI} \models\ & highestLevel(\psi_1, s)=k \wedge \\
& Do(interact(agt_2, \psi_1, \psi_2, F_j), s, do(a_2, do(a_1, s))) \wedge \\
& mostSpecCompatSuperGoal(\psi_1, \psi_2, do(a_1, s)) = \phi \wedge \\
& \exists n.PGoal(\phi, n, do(a_1, s)) \wedge n < k \wedge \exists p.G(p, n, do(a_1, s)) \wedge \\
& \quad G_{\cap}(p, n, do(a_1, s)) \wedge highestLevel(\phi, s) = n \wedge \\
& \neg CGoal(\neg \exists s_1, s_2, p'.Starts(p, do(a_1, s_1)) \wedge \\
& \qquad Do(interact(agt_2, \psi_1, \psi_2, F_j), s_1, s_2) \wedge \phi(p) \wedge \\
& \qquad Suffix(p', p, s_2) \wedge map(F, \psi_1, \psi_2)(p'),\ k-1, do(a_1, s)) \\
& \supset CGoal(map(F, \psi_1, \psi_2), k, do(a_2, do(a_1, s))).
\end{aligned}$$

6 DISCUSSION

In this paper, we have focussed on formalizing how an agent adopting a relational style for a certain interaction with an interlocutor changes her goals. But note that some experimental studies on humans have shown that they adopt all relational styles [14, 16].

Essentially these experiments show that humans adopt all relational styles according to a statistical "pattern" assigning a normalized weight to each of the functions $F0, F1, \ldots, F15$. Such a pattern tends to have a similar shape for all normal individuals with a predominance of $F3$ and $F1$ but with all functions with a non-zero weight. These results mean that perhaps our artificial agents should also interact using a distribution over all relational styles according to a similar pattern. However, no one has as yet done experiments on the frequency of switching from one relational style to the next. This is an important issue for building artificial agents that are believable and akin to humans in their interactions [4, 35, 42].

Another fundamental question is how do subjects' relational styles themselves evolve. In EPM, the relational styles $F0, F1, \ldots, F15$, adopted by the subject are themselves objects on which the sixteen functions can be applied. This gives a formal model within EPM of how relational styles change as a result of interactions. Thus we get a table (the *paradox table*) of 256 (16×16) possible changes that allows one to forecast of how interactions may alter the relational style of the interacting subjects. These have been used in clinical practice as a guide to the therapist on how to act towards the patient [14, 17–19, 32]. We can foresee the use of this dynamics of relational styles to improve computer mediated coaching and group facilitation applications, where a artificial agent guides the interaction to help resolve uncomfortable or conflict situations [19, 31].

Finally, we observe that some of the relational styles in EPM are linked to creativity. This has lead to research on the use of EPM to develop creativity-enhancing techniques [15, 17, 46]. So far this has been used for creativity enhancement in humans, but it could also be used to help develop artificial agents that display creativity.

There has been much work on various frameworks for representing agents' goals and their dynamics in recent years [12, 51]. Much of this work has been motivated by the need to support declarative goals in agent programming languages [11, 21–23], to ensure that plan execution is tied to the achievement of the associated goals; for instance, if a plan fails to achieve its goal, another plan can be selected, and if a goal is achieved serendipitously, the associated plans can be dropped. Most of these frameworks only handle restricted forms of temporally extended goals, and few provide a model-theoretic semantics. None provide notions like the EPM relational styles. The KL framework is very general, handles arbitrary temporally extended goals, and has a well developed semantics. Postulates for goal/intention revision in the presence of beliefs are proposed in [10, 26]. [27] discusses which of these hold in KL.

We used the KL framework as a foundation for our account of goal formation through interaction, but the essence of the account is not tied to this particular framework and should readily be adaptable to others. The main requirements are support for goal adoption and contraction, as well as a hierarchy of subgoals.

In conclusion, the technical contributions of this paper are as follows: first in Section 4, we have generalized of the subgoal adoption mechanism of KL to allow the priority of the new subgoal to be specified and proved some important properties of the resulting goal dynamics framework, and second in Section 5, we have formalized an account of goal change through interaction based on EPM in our framework and proved that it satisfies some key requirements. More generally, we have shown how one can incorporates the rich range of relational styles from EPM into formal accounts of goal changes, a contribution, which can be fruitful for the realization of new kinds of artificial agents that are not purely rational servants.

In future work, we would like to extend our account to model how an agent's relational style is selected depending on the situation and how it evolves. We would also like to refine our account to incorporate models of emotions [41, 49], trust [6, 50], and norms [3], and how they affect goal change in interactions. We also want to examine the relationship of our EPM-based approach with work on argumentation frameworks [25, 38] and communication protocols, although the EPM deals with more basic considerations, namely interaction styles and attitudes, which are not necessarily rational. Our notion of interaction, like the EPM one, is abstract. Clearly one major way of making interaction concrete is through dialog and using conceptual tools such as speech act theory. So one could integrate our framework with a dialog model and extract from the dialog the relational styles in the interaction. This is a very compelling avenue for further research and some work on EPM such as [7] provides a good starting point. Finally, we would like to evaluate the usefulness of the account in applications.

ACKNOWLEDGMENTS

Work supported by the European Research Council under the European Union's Horizon 2020 Programme through the ERC Advanced Grant WhiteMech (No. 834228), as well as the National Science and Engineering Research Council of Canada.

REFERENCES

[1] David W. Aha. 2018. Goal Reasoning: Foundations, Emerging Applications, and Prospects. *AI Magazine* 39, 2 (2018), 3–24.

[2] David W. Aha and Alexandra Coman. 2017. The AI Rebellion: Changing the Narrative. In *AAAI*. AAAI Press, 4826–4830.

[3] Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W. N. van der Torre (Eds.). 2013. *Normative Multi-Agent Systems*. Dagstuhl Follow-Ups, Vol. 4. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

[4] Joseph Bates. 1994. The Role of Emotion in Believable Agents. *Commun. ACM* 37, 7 (1994), 122–125.

[5] Gregory Bateson. 1979. *Mind and Nature*. E. P. Dutton, New York.

[6] Robin Cohen, Mike Schaekermann, Sihao Liu, and Michael Cormier. 2019. Trusted AI and the Contribution of Trust Modeling in Multiagent Systems. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, 1644–1648.

[7] Luigi Colazzo, Daniela Silvestri, and Luisa Mich. 1992. An application of the elementary pragmatic model to electronic communication. *Annali dell'Istituto superiore di sanità* 28 (02 1992), 245–251.

[8] Michael T. Cox. 2007. Perpetual Self-Aware Cognitive Agents. *AI Magazine* 28, 1 (2007), 32–46.

[9] Michael T. Cox, Dustin Dannenhauer, and Sravya Kondrakunta. 2017. Goal Operations for Cognitive Systems. In *AAAI*. AAAI Press, 4385–4391.

[10] Célia da Costa Pereira, Andrea Tettamanzi, and Leila Amgoud. 2006. Goal Revision for a Rational Agent. In *ECAI (Frontiers in Artificial Intelligence and Applications)*, Vol. 141. IOS Press, 747–748.

[11] Mehdi Dastani. 2008. 2APL: a practical agent programming language. *Autonomous Agents and Multi-Agent Systems* 16, 3 (2008), 214–248.

[12] Mehdi Dastani, M. Birna van Riemsdijk, and Michael Winikoff. 2011. Rich goal types in agent programming. In *AAMAS*. IFAAMAS, 405–412.

[13] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. 2000. ConGolog, A Concurrent Programming Language Based on the Situation Calculus. *Artificial Intelligence* 121, 1–2 (2000), 109–169.

[14] Piero De Giacomo. 1993. *Finite systems and infinite interactions: The logic of human interaction and its application to psychotherapy*. Bramble Books.

[15] Piero De Giacomo. 1999. *Mente e Creatività (Mind and Creativity)*. Franco Angeli.

[16] Piero De Giacomo, Luciano L'Abate, Francesco Margari, Andrea De Giacomo, Wanda Santamato, and Rita Masellis. 2008. Sentences with Strong Psychological Impact in Psychotherapy: Research in Progress. *Journal of Contemporary Psychotherapy* 38, 2 (2008), 65–72.

[17] Piero De Giacomo, Luisa Mich, Carlos Santamaria, Laura G. Sweeney, and Andrea De Giacomo. 2012. Information Processing. In *Paradigms in Theory Construction*, L'Abate L. (Ed.). Springer.

[18] Piero De Giacomo and Alberto Silvestri. 1985. An Elementary Pragmatic Model in Family Therapy. *International Journal of Family Therapy* 6 (1985), 245–263.

[19] Rodolfo A. Fiorini, Piero De Giacomo, and Luciano L'Abate. 2016. Wellbeing Understanding in High Quality Healthcare Informatics and Telepractice. In *ICIMTH (Studies in Health Technology and Informatics)*, Vol. 226. IOS Press, 153–156.

[20] Asbjørn Følstad, Petter Bae Brandtzæg, Tom Feltwell, Effie L.-C. Law, Manfred Tscheligi, and Ewa A. Luger. 2018. SIG: Chatbots for Social Good. In *CHI Extended Abstracts*. ACM.

[21] James Harland, David N. Morley, John Thangarajah, and Neil Yorke-Smith. 2014. An operational semantics for the goal life-cycle in BDI agents. *Autonomous Agents and Multi-Agent Systems* 28, 4 (2014), 682–719.

[22] James Harland, David N. Morley, John Thangarajah, and Neil Yorke-Smith. 2017. Aborting, suspending, and resuming goals and plans in BDI agents. *Autonomous Agents and Multi-Agent Systems* 31, 2 (2017), 288–331.

[23] Koen V. Hindriks, Wiebe van der Hoek, and M. Birna van Riemsdijk. 2009. Agent programming with temporally extended goals. In *AAMAS (1)*. IFAAMAS, 137–144.

[24] Philip Hingston (Ed.). 2012. *Believable Bots: Can Computers Play Like People?* Springer.

[25] Joris Hulstijn and Leendert W. N. van der Torre. 2004. Combining goal generation and planning in an argumentation framework. In *NMR*. 212–218.

[26] Thomas F. Icard III, Eric Pacuit, and Yoav Shoham. 2010. Joint Revision of Beliefs and Intention. In *KR*. AAAI Press.

[27] Shakil M. Khan. 2018. *Rational Agents: Prioritized Goals, Goal Dynamics, and Agent Programming Languages with Declarative Goals*. Ph.D. Dissertation. Dept. of Electrical Engineering and Computer Science, York University, Toronto, ON Canada. [http://www.cse.yorku.ca/~lesperan/papers/SKhanPhD.pdf].

[28] Shakil M. Khan and Yves Lespérance. 2009. Prioritized Goals and Subgoals in a Logical Account of Goal Change - A Preliminary Report. In *DALT (Lecture Notes in Computer Science)*, Vol. 5948. Springer, 119–136.

[29] Shakil M. Khan and Yves Lespérance. 2010. A logical framework for prioritized goal change. In *AAMAS*. IFAAMAS, 283–290.

[30] Shakil M. Khan and Yves Lespérance. 2016. Infinite Paths in the Situation Calculus: Axiomatization and Properties. In *KR*. AAAI Press, 565–568.

[31] Naoki Koyama, Kazuaki Tanaka, Kohei Ogawa, and Hiroshi Ishiguro. 2017. Emotional or Social?: How to Enhance Human-Robot Social Bonding. In *HAI*. ACM, 203–211.

[32] Luciano L'Abate and Piero De Giacomo. 2003. *Intimate Relationships and How to Improve Them*. Praeger.

[33] Hector J. Levesque, Fiora Pirri, and Raymond Reiter. 1998. Foundations for the Situation Calculus. *Electron. Trans. Artif. Intell.* 2 (1998), 159–178.

[34] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. 1997. GOLOG: A logic programming language for dynamic domains. *The Journal of Logic Programing* 31, 1-3 (1997), 59–83.

[35] A. Bryan Loyall, W. Scott Neal Reilly, Joseph Bates, and Peter Weyhrauch. 2004. System for authoring highly interactive, personality-rich interactive characters. In *Symposium on Computer Animation*. The Eurographics Association, 59–68.

[36] John McCarthy and Patrick J. Hayes. 1969. Some Philosophical Problems From the Standpoint of Artificial Intelligence. *Machine Intelligence* 4 (1969), 463–502.

[37] Matthew Molineaux, Matthew Klenk, and David W. Aha. 2010. Goal-Driven Autonomy in a Navy Strategy Simulation. In *AAAI*. AAAI Press.

[38] Ariel Monteserin and Analía Amandi. 2011. Argumentation-based negotiation planning for autonomous agents. *Decision Support Systems* 51, 3 (2011), 532–548.

[39] Robert C. Moore. 1985. A Formal Theory of Knowledge and Action. In *Formal Theories of the Common Sense World*, J. R. Hobbs and Robert C. Moore (Eds.). Ablex Publishing, Norwood, NJ, 319–358.

[40] Dana S. Nau. 2007. Current Trends in Automated Planning. *AI Magazine* 28, 4 (2007), 43–58.

[41] Magalie Ochs, David Sadek, and Catherine Pelachaud. 2012. A formal model of emotions for an empathic rational dialog agent. *Autonomous Agents and Multi-Agent Systems* 24, 3 (2012), 410–440.

[42] Florian Pecune, Magalie Ochs, Stacy Marsella, and Catherine Pelachaud. 2016. SOCRATES: from SOCial Relation to ATtitude ExpressionS. In *AAMAS*. ACM, 921–930.

[43] Amir Pnueli. 1977. The Temporal Logic of Programs. In *FOCS*. 46–57.

[44] Ray Reiter. 2001. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.

[45] Stuart Russell. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Random House.

[46] Victoria Sakhnini, Luisa Mich, and Daniel M. Berry. 2017. Group versus individual use of power-only EPMcreate as a creativity enhancement technique for requirements elicitation. *Empirical Software Engineering* 22, 4 (2017), 2001–2049.

[47] Richard B. Scherl and Hector J. Levesque. 2003. Knowledge, action, and the frame problem. *Artif. Intell.* 144, 1-2 (2003), 1–39.

[48] Alberto Silvestri, Piero De Giacomo, Gianpaolo Pierri, Ezio Lefons, Maria Teresa Pazienza, and Filippo Tangorra. 1980. A Basic Model of Interacting Subjects. *Cybernetics and Systems* 11, 1-2 (1980), 115–129.

[49] Bas R. Steunebrink, Mehdi Dastani, and John-Jules Ch. Meyer. 2012. A formal model of emotion triggers: an approach for BDI agents. *Synthese* 185, Supplement-1 (2012), 83–129.

[50] W. T. Luke Teacy, Michael Luck, Alex Rogers, and Nicholas R. Jennings. 2012. An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling. *Artif. Intell.* 193 (2012), 149–185.

[51] M. Birna van Riemsdijk, Mehdi Dastani, and John-Jules Ch. Meyer. 2009. Goals in conflict: semantic foundations of goals in agent programming. *Autonomous Agents and Multi-Agent Systems* 18, 3 (2009), 471–500.

[52] Yuyu Xu, Pedro Sequeira, and Stacy Marsella. 2017. Towards modeling agent negotiators by analyzing human negotiation behavior. In *ACII*. IEEE Computer Society, 58–64.