Appraisal in Human-Robot Collaboration

Mahni Shayganfar, Charles Rich, Candace L. Sidner
Worcester Polytechnic Institute
Computer Science Department
100 Institute Road
Worcester, Massachusetts 01609
mshayganfar | rich | sidner @wpi.edu

ABSTRACT

We have investigated the mutual influence of affective and collaborative processes in a cognitive theory to support interaction between humans and robots or virtual agents. We have developed new algorithms for appraisal processes, as part of a new overall computational model for implementing collaborative robots and agents. We build primarily on the *cognitive appraisal* theory of emotions and the *SharedPlans* theory of collaboration to investigate the structure, fundamental processes and functions of emotions in a collaboration. We have evaluated our proposed algorithms by conducting an online user study.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Delphi theory

Keywords

Appriasal, Human-Robot/Agent Collaboration, Cognitive Modeling

1. INTRODUCTION

Sousa in The Rationality of Emotion [18] makes a good case for the claim that humans are capable of rationality largely because they are creatures with emotions. The idea of having robots or other intelligent agents living in a human environment has been a persistent dream from science fiction books to artificial intelligence and robotics laboratories. Collaborative robots are becoming an integral part of humans' environment to accomplish their industrial and household tasks. In these environments humans are involved in robots' operations and decision making processes. This involvement of humans influences the efficiency of robots' interaction and performance, and makes them dependent on the humans' cognitive abilities and mental states.

This work is part of a larger effort to build robots capable of generating and recognizing emotions in order to be better collaborators. In this paper, we report on the specific problem of appraising events within a collaborative interaction. Our contribution is to ground general appraisal concepts in the specific context and structure of collaboration. This work is part of the development of *Affective Motivational Collaboration Theory* which is built on the foundations of the *SharedPlans* theory of collaboration [7] and the *cognitive appraisal* theory of emotions [6].

We start by briefly introducing the Affective Motivational Collaboration Theory focusing on the collaboration and appraisal mechanisms as well as mental states. We then provide more details about the graph representation of the robot's mental state. Next, we describe the algorithms we developed to compute the value of four crucial appraisal variables. To compare the results from our algorithms with humans' decisions we have conducted a user study using crowd sourcing. In Section 7 we provide the results from our user study.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), John Thangarajah, Karl Tuyls, Stacy Marsella, Catholijn Jonker (eds.), May 9–13, 2016, Singapore.*

2. RELATED WORK

Our work builds on the general notions of appraisal theory, but focused on its application in human-robot collaboration. Computational appraisal models have been applied to a variety of uses including psychology, robotics, AI, and cognitive science. For instance, in [11] EMA is used to generate specific predictions about how human subjects will appraise and cope with emotional situations. Furthermore, appraisal theory has also been used in robots' decision making [5], or in their cognitive systems [8, 10]. Additionally, in the virtual agents community, empathy and affective decision-making is a research topic that has received much attention in the last two decades [12, 13, 14, 20]. However, EMA and several other examples in artificial intelligence and robotics which apply appraisal theory do not focus on the dynamics of collaborative contexts [1, 9, 11, 15].

The computational collaboration model in our work is strongly influenced by the SharedPlans theory [7]. However, our algorithms are also compatible with other collaboration theories, e.g., Joint Intentions theory [3], or STEAM as an hybrid collaboration theory [19]. These theories have been extensively used to examine and describe teamwork and collaboration. Yet, they have never been combined, as they are in our work. We believe a systematic integration of collaboration theories and appraisal theory can help us describe the underlying collaboration processes leading to the existing collaboration structures.

3. AFFECTIVE MOTIVATIONAL COLLABORATION THEORY

Affective Motivational Collaboration Theory deals with the interpretation and prediction of observable behaviors in a dyadic collaboration. The theory focuses on the processes regulated by emotional states. The observable behaviors represent the outcome of reactive and deliberative processes related to the interpretation of the self's relationship to the environment. Affective Motivational Collaboration Theory aims to explain both rapid emotional reactions to events as well as slower, more deliberative responses. The reactive and deliberative processes are triggered by two types of events: *external* events, such as the other's *utterances* and *primitive actions*, and *internal* events, comprising changes in the self's mental states, such as belief formation and emotional changes. The theory explains how emotions regulate the underlying processes when these events occur. It also elucidates the role of *motives* as goal-driven emotion-regulated constructs with which an robot can form new intentions to cope with events.

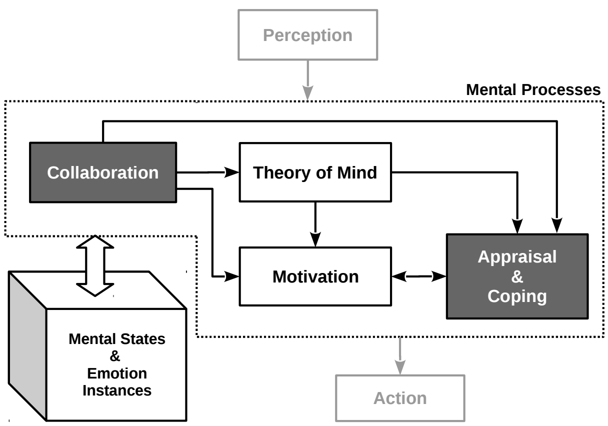

Figure 1: Computational framework based on Affective Motivational Collaboration Theory (arrows indicate primary influences between mechanisms).

Our focus is on the mechanisms depicted as mental processes in Figure 1 along with the mental states. Each mechanism includes one our more processes in our architecture. For instance, the *Collaboration* mechanism includes processes such as *Focus Shifting* and *Constraint Management*, or as we discuss in Section 6 the *Appraisal* mechanism includes processes to compute the values for different appraisal variables. The *mental states* includes self's (robot's) beliefs, intentions, motives, goals and emotion instances as well as the anticipated mental states of the other (human). The *Collaboration* mechanism maintains constraints on actions, including task states and the ordering of tasks (see Figure 2). The *Collaboration* mechanism also provides processes to update and monitor the shared plan. The *Appraisal* mechanism is responsible for evaluating changes in the self's mental states, the anticipated mental states of the other, and the state of the collaboration environment. The *Coping* mechanism provides the self with different coping strategies associated with changes in the self's mental states with respect to the state of the collaboration. The *Motivation* mechanism operates whenever the self a) requires a new motive to overcome an internal impasse in an ongoing task, or b) wants to provide an external motive to the other when the other faces a problem in a task. The *Theory of Mind* mechanism infers a model of the other's anticipated mental state. The self progressively updates this model during the collaboration.

3.1 Mental States

A brief description of mental states is provided as prerequisite knowledge for understanding the appraisal processes. The mental states shown in Figure 1 comprise the knowledge base required for all the mechanisms in the overall model. Mental states are conscious states of the mind providing the content for cognitive processes. Affective Motivational Collaboration Theory operates with the following mental states: beliefs, intentions, motives, goals and emotion instances. These mental states possess attributes, each of which provides a discriminating and unique interpretation of the related cognitive entities. The self uses these attributes whenever there is an arbitration in the internal cognitive processes. We only describe the attributes of beliefs and motives in this paper, since they are used in our appraisal algorithms. We do not provide the way we compute the attributes' values, due to the limited space.

3.1.1 Belief:

Beliefs are a crucial part of the mental states. We have two different perspectives on categorization of beliefs. In one perspective, we categorize beliefs based on whether they are shared between the collaborators. In the second perspective, beliefs are categorized based on who or what they are about. In this categorization, beliefs can be about the self, the other, or they can be about the environment. Beliefs can be created and updated by different processes. They also affect how these processes function as time passes.

The attributes of a belief are involved in arbitration procedures within different processes in Affective Motivational Collaboration Theory. They impact a range of these processes from the formation of new beliefs, the evaluation of an external event by the Appraisal mechanism, generation of new motives and updates on collaboration plan, to the activation of coping strategies and ultimately the self's behavior. We use six belief attributes in our framework. Belief *strength* is about how strongly the self holds salient beliefs about an object, an entity, or an anticipated behavior. *Accuracy* of a belief is the relation between that belief and the truth which that belief is about. The *frequency* of a belief is related to how regularly it appears as the result of an internal or an external event. The *recency* of a belief refers to how temporally close a particular belief is to the current state of collaboration. The *saliency* of a belief is a cognitive attribute that pertains to how easily the self becomes aware of a belief. The *persistence* of a belief refers to how resistant the belief is to changes.

3.1.2 Motive:

Motives are mental constructs which can initiate, direct and maintain goal-directed behaviors. They are created by the emotion-regulated Motivation mechanism. Motives can cause the formation of a new intention for the robot according to: a) its own emotional states (how the robot appraises the environment), b) its own private goal (how an action helps the robot to make progress), c) the collaboration goal (how an action helps to achieve the shared goal), and d) other's anticipated beliefs (how an action helps the other).

Motives can be compared on various dimensions [17], and they possess a set of attributes. The Motivation mechanism compares motives based on the quality of these attributes and chooses the one which is the most related to the current state of the collaboration. We have the following five motive attributes in our framework. The *insistence* of a motive defines the "interrupt priority level" of the motive, and how much that motive can attract the self's focus of attention. The *importance* of a motive is determined by the corresponding beliefs about the effects of achieving or not achieving the associated goal. The *urgency* of a motive defines how much time the self has to acknowledge and address that motive before it is too late. The *intensity* of a motive determines how actively and vigorously that motive can help the self to pursue the goal if adopted; rather than abandoning the goal and ultimately the collaboration. The *failure disruptiveness* attribute of a motive determines how disruptive failure is to achieving the corresponding goal.

3.1.3 Intention:

Intentions are mental constructs directed at goals and future actions. They play an essential role in taking actions according to the collaboration plan as well as behavior selection in the Coping mechanism. Intentions are also involved in selecting intention-related strategies, e.g., planning, seeking instrumental support and procrastination. Intentions possess a set of attributes, i.e., *Temporal Status, Direct Experience, Certainty, Ambivalence, Affective-Deliberative Consistency* which moderate the consistency between intention and behavior [4]. The details about these attributes are out of this paper's context.

3.1.4 Goal:

Goals help the robot to create and update its collaboration plan according to the current private and shared goal content and structure, i.e., the *Specificity, Proximity* and *Difficulty* of the goal. Goals direct the formation of intentions to take appropriate corresponding actions during collaboration. Goals also drive the Motivation mechanism to generate required motive(s). The details about goal's attributes are also out of this paper's context.

3.1.5 Emotion Instance:

Emotions in mental states are emotion instances that are elicited by the Appraisal mechanism, e.g., *Joy, Anger, Hope, Worry*. These emotion instances include the robot's own emotions as well as the anticipated emotions of the other which are created with the help of the processes in the Theory of Mind mechanism. Each emotion has its own functionality in either the intrapersonal or interpersonal level. These emotions not only regulate the self's internal processes, but also assist the self to anticipate the other's mental states.

4. EXAMPLE SCENARIO

The example scenario is part of a much larger interaction we are implementing to test our theory. This example shows a very short part of an interaction between a robot and an astronaut during their collaboration. Their mission is to finish installing a few solar panels together. However, the astronaut encounters a measurement tool problem:

Astronaut [turn t-1]: Oh no! Finishing the quality check of our installation with this measurement problem is so frustrating. I think we should stop now!

Robot [turn t]: I see. This is frustrating. But, I can help you with the measurement tool and we can finish the task as originally planned.

As shown, the robot in turn t, acknowledges the astronaut's frustration and appropriately responds to her problem.

5. COLLABORATION

The Collaboration mechanism constructs a hierarchy of goals associated with tasks in the form of a hierarchical task network (see Figure 2), and also manages and maintains the constraints and other required details of the collaboration including the inputs and outputs of individual tasks, the *preconditions* (specifying whether it is appropriate to perform a task), and the *postconditions* (specifying whether a just-completed task was successful). Collaboration also keeps track of the focus of attention, which determines the salient objects, properties and relations at each point, and shifts the focus of attention during the interaction.

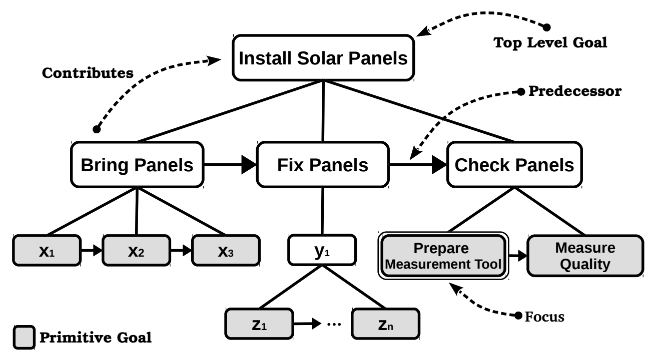

Figure 2: Collaboration structure (shared plan).

Here, we briefly describe the methods which retrieve information about the collaboration structure, and are used in our algorithms to compute the values of appraisal variables. In these methods, ε_t is the event corresponding to time t, and g_t is a given goal at time t.

- *recognizeGoal(ε_t)* returns the unique goal to which the given event (action, utterance, or emotional expression) directly contributes, or *ambiguous* if this method does not recognize a goal in the plan.

- *topLevelGoalStatus(g_t)* returns the status of the top level goal whether it is ACHIEVED, FAILED, BLOCKED, INAPPLICABLE, PENDING, or IN PROGRESS. In our example, "Install Solar Panels" is the top level goal.

- *currGoalStatus(g_t)* returns the current goal status whether it is ACHIEVED, FAILED, BLOCKED, INAPPLICABLE, PENDING, or IN PROGRESS. In our example, "Prepare Measurement Tool" is the current (focused) goal.

- *precondStatus(g_t)* returns the status of the precondition for the given goal whether it is SATISFIED, UNSATISFIED or UNKNOWN. For instance, the precondition for fixing a panel is whether the panel is appropriately located on its frame.

- *doesContribute(g_t)* returns whether the given goal contributes to another goal in the higher level of the plan hierarchy. For instance, an abstract (nonprimitive) goal of "Bring Panels" contributes to the higher level goal of "Install Solar Panels".

- *extractContributingGoals(g_t)* returns all the contributing goals of the given goal. For instance, the "Prepare Measurement Tool" and "Measure Quality" are two goals contributing to the "Check Panels" nonprimitive goal.

- *extractPredecessors(g_t)* returns the predecessors of the given goal. For instance, the "Prepare Measurement Tool" goal is the predecessor of another goal called "Measure Quality".

- *extractInputs(g_t)* returns all the required inputs for the given goal. For example, the goal "Fix Panels" requires inputs such as the *welding tool* and the *panel*.

- *isAvailable(g_t)* returns whether the given input is available. For instance, if the *welding tool* is required for the goal "Fix Panels", is it available now?

- *isAchieved(g_t)* returns whether the given goal is achieved, i.e., whether all the postconditions of the given goal are SATISFIED.

- *isFocused(g_t)* returns whether the focus is on given goal now. The focus is on the goal "Prepare Measurement Tool". The focused goal is the goal that the robot currently is pursuing.

- *getResponsible(g_t)* returns responsible agents of the given goal. In a dyadic collaboration, both of the agents can be partly responsible for a nonprimitive goal, while each robot is responsible for one or more primitive goal. For instance, both the robot and the astronaut are responsible for the nonprimitive goal of "Install Solar Panels", whereas it is only the astronaut who is responsible for the primitive goal of "Prepare Measurement Tool".

6. APPRAISAL PROCESSES

We consider four appraisal variables to be the most important variables in a collaboration context, i.e., *Relevance* (Algorithm 1), *Desirability* (Algorithm 2), *Expectedness* (Algorithm 3), and *Controllability* (Algorithm 4). There are other appraisal variables introduced in psychological [16] and computational literature [6]. We believe most of these variables can be straightforwardly added to our appraisal mechanism later. All of the algorithms in this section use mental states of the robot (discussed in Section 3.1) which are formed based on the collaboration structure. These algorithms use the corresponding recognized goal of the most recent event at each turn.

6.1 Relevance

Relevance as an appraisal variable measures the significance of an event for the robot. An event can be evaluated to be relevant if it has a positive utility or it can causally impact a state with a positive utility [11]. Relevance is an important appraisal variable since the other appraisal variables are more meaningful only for the relevant events.

Algorithm 1 determines the relevance of the given event with respect to the current mental state. The relevance of the event depends on the significance of the event with respect to the current collaboration status. The significance of an event is determined based on the utility of the event as it is also presented in [6, 11]. We believe although the utility of the event represents the significance of the event, the other collaborator's expressed emotion also plays a role by influencing the significance of the utility through a threshold. As a result, evaluating the relevance of the events can cause a collaborative robot to effectively respond only to the events which can positively impact the status of the shared goal without dedicating all other resources to every single event. The relevance process also benefits from the information that the collaboration structure contains, e.g., shared goal.

Algorithm 1 (Relevance)

function ISEVENTRELEVANT(Event ε_t)

  $g_t \leftarrow recognizeGoal(\varepsilon_t)$

  $\mathcal{U} \leftarrow$ GETEVENTUTILITY(g_t)
  $\tau_t \leftarrow$ GETEMOTIONALTHRESHOLD(g_t)

  if ($\mathcal{U} \geq \tau_t$) **then**
    return RELEVANT
  else
    return IRRELEVANT

After perceiving an event, it is the belief about that event which represents the event in robot's mental state. Also, the *recognizeGoal* returns the goal (g_t) to which the current event contributes, unless it is *ambiguous*; g_t represents the shared goal at time (turn) t within the shared plan. We compute the utility ($0 \leq \mathcal{U} \leq 1$) of the event based on the values of the attributes associated with the existing beliefs in the mental state, as well as the attributes of the motive associated with the recognized goal. We use three of the belief attributes discussed in Section 3.1.1 to compute belief related part of the utility:

- *Strength*: The extent to which the pre and post conditions of a goal and its predecessors and/or contributing goals are SATISFIED or UNSATISFIED makes a belief about the goal more stronger. Respectively, an UNKNOWN pre and post condition status of a goal and its predecessors and/or contributing goals forms beliefs with lower strength.

- *Saliency*: Beliefs related to the goal at the top of the focus stack are more salient than beliefs related to any other goal in the plan, whether those goals are already ACHIEVED or FAILED, or they will be pursued to achieve in the future.

- *Persistence*: The recurrence of a belief over the passage of time (turns) increases the persistence of the belief. Beliefs occuring only in one turn have the lowest value of persistence.

We also use two of the motive attributes discussed in Section 3.1.2 to compute motive related part of the utility:

- *Urgency*: There are two factors imapcting the urgency of a motive: a) whether the goal directing the given motive is the predecessor of another goal for which the other collaborator is responsible, and b) whether achieving the goal directing the given motive can mitigate the other collaborator's negative valenced emotion.
- *Importance*: A motive is important if failure of the directing goal causes an impasse in the shared plan (i.e., no further goal is available to achieve), or achievement of the directing goal removes an existing impasse.

We compute the utility of an event based on these five attributes. The value of each attribute is between 0 and 1, and we consider the same weight for each attribute. These weights can be learned or modified when our framework is fully implemented. The value of the overall utility is computed using a simple weighted averaging function which makes the overall value to be between 0 and 1.

The significance of an event in a collaborative environment is based not only on the utility of the event, but it is also influenced by the perceived emotion of the human collaborator. The human's emotion influences the decision about the utility of the event in form of a threshold value τ_t (see Algorithm 1). For instance, a positive expressed emotion of the human reduces the threshold value which consequently makes the robot find an event relevant with even a slightly positive utility. This threshold value (τ_t) is currently determined based on whether the valence of the human's perceived emotion is positive (e.g., happiness) or negative (e.g., anger). This approach will be replaced by using a Fuzzy Logic system on a modified three-dimensional space of the somatic markers (to support some of the social emotions) associated with the human's expressed emotion as they are described in [2]. In this space every emotion can be mapped to a vector of three values, i.e., *Arousal, Valence* and *Stance*. The details about this process is out of this paper's context. Finally, we make our decision about the relevance of an event with respect to the human's emotional state. Consequently, an event can be considered *irrelevant* even though the utility has a relatively positive value.

6.2 Desirability

Desirability characterizes the value of an event to the robot in terms of whether the event facilitates or thwarts the collaboration goal. Desirability captures the valence of an event with respect to the robot's preferences [6]. In a collaborative robot, preferences are biased towards those events facilitating progress in the collaboration. An event is desirable if it facilitates the state of the shared goal, or if it inhibits the state of a goal that is inconsistent with respect to the shared goal.

Desirability plays an important role in the overall architecture; it makes the processes involved in the other mechanisms (e.g., Motivation and Theory of Mind), and consequently the robot's mental state, congruent with the collaboration status which is a collaborative robot's desire. Therefore, it causes the robot to dismiss events causing inconsistencies in the robot's collaborative behavior. Moreover, desirability is also crucial from the collaboration's point of view. A collaborative robot needs to know whether its own and the other collaborator's actions, utterances, and emotional expressions are desirable in terms of their consistence

Algorithm 2 (Desirability)

1: **function** ISEVENTDESIRABLE(Event ε_t)

2: $g_t \leftarrow$ *recognizeGoal*(ε_t)

3: **if** (*topLevelGoalStatus*(g_t) = ACHIEVED) **then**
4: **return** MOST-DESIRABLE
5: **else if** (*topLevelGoalStatus*(g_t) = FAILED) **then**
6: **return** MOST-UNDESIRABLE
7: **else if** (*topLevelGoalStatus*(g_t) = BLOCKED) **or** (*topLevelGoalStatus*(g_t) = INAPPLICABLE) **then**
8: **return** UNDESIRABLE
9: **else if** (*topLevelGoalStatus*(g_t) = PENDING) **or** (*topLevelGoalStatus*(g_t) = INPROGRESS) **then**

10: **if** (*currGoalStatus*(g_t) = ACHIEVED) **then**
11: **return** DESIRABLE
12: **else if** (*currGoalStatus*(g_t) = FAILED) **then**
13: **return** MOST-UNDESIRABLE
14: **else if** (*currGoalStatus*(g_t) = BLOCKED) **or** (*topLevelGoalStatus*(g_t) = INAPPLICABLE) **then**
15: **return** UNDESIRABLE
16: **else if** (*topLevelGoalStatus*(g_t) = PENDING) **or** (*currGoalStatus*(g_t) = INPROGRESS) **then**
17: **return** NEUTRAL

with the status of the current shared goal. In other words, the collaboration mechanism uses the appraisal process of desirability to coordinate what the self or the other does, says, and expresses during collaboration. Reciprocally, the appraisal mechanism and in this case the desirability process use the collaboration structure to obtain their required information.

Algorithm 2 provides a process in which the desirability of an event is computed with regard to the status of the shared goal; i.e., it operates based on whether and how the event changes the status of the current shared goal. It receives the current event, ε_t, and decides whether and how the event is DESIRABLE OR UNDESIRABLE. First, the algorithm checks the status of the collaboration's top level goal (lines 3 to 9), and if the the top level goal is still INPROGRESS or it is PENDING, it continues by checking the status of the current shared goal (lines 10 to 17). If any of the top level and current shared goals are ACHIEVED in these two steps, the robot interprets the event as a desirable one. However, if any of these goals are BLOCKED, the event will be considered undesirable by the robot.

The algorithm continues in the case of an unknown status of the current shared goal, and checks whether the precondition(s) of the associated goal with the current event, g_{ε_t}, are satisfied (lines 17 to 21). The robot prefers the satisfied preconditions and interprets the event as desirable while unsatisfied preconditions are undesirable for the robot. For instance, a satisfied precondition of a future goal is still desirable for the robot to some extent. Note that the robot also checks the ambiguity of the associated goal with the current event (line 15). An ambiguous goal is a goal which is not recognized in the robot's plan, and it is undesirable for the robot. Finally, if the preconditions of the associated goal with the current event are unknown, the robot checks whether this goal, g_{ε_t}, contributes to the current shared goal, g_t (lines 22 to 25). As a result a contributing goal will obtain a neutral desirability in comparison with a noncontributing goal which will be undesirable for the robot.

6.3 Expectedness

Expectedness is the extent to which the truth value of a state could have been predicted from causal interpretation of an event [11]. In the collaboration context the expectedness of an event measures the congruency of the event with respect to the existing knowledge about the shared goal. Therefore, expected events are those of which beliefs about them are congruent to the status of the collaboration since their associated goals are expected with respect to the shared plan.

Expectedness underlies a collaborative robot's attention by evaluating the congruence of events with respect to the structure of an existing shared plan. Congruent beliefs in a robot's mental state will lead to more consistent and effective outcomes of the processes in the overall architecture. Therefore, a collaborative robot uses expectedness to maintain its own mental state towards the shared goal. The robot will also be able to respond to unexpected but relevant events. As a result, the collaboration mechanism uses expectedness to maintain the robot's attention and subsequently its mental state with respect to the shared goal. In parallel, the appraisal mechanism uses the underlying information of the collaboration structure to evaluate the expectedness of an event.

Algorithm 3 (Expectedness)

1: **function** IsEventExpected(Event ε_t)

2: $\quad g_t \leftarrow \mathit{recognizeGoal}(\varepsilon_t)$
3: $\quad g_{top} \leftarrow \mathit{getTopGoal}(g_t)$

4: $\quad$ **if** $(\mathit{isLive}(g_t))$ **then**
5: $\qquad$ **if** $(\mathit{isNeccessaryFocusShift}(g_t))$ **then**
6: $\qquad\quad$ **return** MOST-EXPECTED
7: $\qquad$ **else**
8: $\qquad\quad$ **return** EXPECTED
9: $\quad$ **else**
10: $\qquad$ **if** $(\mathit{isPath}(g_t, g_{top}))$ **then**
11: $\qquad\quad$ **return** UNEXPECTED
12: $\qquad$ **else**
13: $\qquad\quad$ **return** MOST-UNEXPECTED

In Algorithm 3 we provide the process of the expectedness based on the shared plan and status of the shared goal. The key point in this algorithm is the status of the current shared goal (g_t) and its relationship with the goal associated with the current event (g_{ε_t}). The algorithm receives the current mental graph, $\mathcal{G}_t$, and the current event, ε_t, from input, and decides whether the current event is expected.

First, we need to extract the goal in the current mental graph and the recognized goal associated with the current event. Similar to the desirability algorithm (Algorithm 2), we check whether the g_{ε_t} is ambiguous. In the case of ambiguity in g_{ε_t}, we consider the current event unexpected since an effective collaboration requires perceivable and unambiguous goals associated with the events. We continue by the comparison of the current shared goal and the recognized goal associated with the current event with respect to the shared plan. If these two goals are not the same, it is possible that the current shared goal is already achieved. The event will be unexpected (line 10) if the current shared goal is not achieved and the current event does not refer to the same goal. However, if the current goal is achieved, it is important to see whether its parent is also achieved (line 12). This step is important because the event can be expected if the new goal contributes to the parent of the recently achieved goal. Therefore, if the parent goal in the hierarchical plan is not achieved, the contribution of the associated goal to the current event can help us to decide whether the event is expected (lines 12 to 16). However, if the parent goal is already achieved, the new goal can contribute (as a child) to the recently achieved shared goal, i.e., g_t, which is also expected (line 19). On the contrary, if the new goal does not contribute to g_t, it might be a goal in another branch in the shared plan which has received focus and should be achieved. In such a case, again, the event will be expected; otherwise we consider the event unexpected (lines 21 to 24).

6.4 Controllability

Controllability is the extent to which an event can be influenced, and it is associated with a robot's ability to cope with an appraised event [6]. Thus, a robot can determine whether the outcome of an event can be altered by some actions under either of the collaborators' control. In other words, controllability is a measure of a robot's ability to maintain or change a particular state as a consequence of an event.

Algorithm 4 (Controllability)

1: **function** IsEventControllable(Event ε_t)

2: $\quad \alpha \leftarrow$ GetAgencyRatio(ε_t)
3: $\quad \beta \leftarrow$ GetAutonomyRatio(ε_t)

4: $\quad \lambda \leftarrow$ GetSucPredecessorsRatio(ε_t)
5: $\quad \mu \leftarrow$ GetAvailableInput(ε_t)

6: $\quad \mathcal{U} \leftarrow \frac{\omega_0 \cdot \alpha + \omega_1 \cdot \beta + \omega_2 \cdot \lambda + \omega_3 \cdot \mu}{\omega_0 + \omega_1 + \omega_2 + \omega_3}$

7: $\quad \tau_t \leftarrow$ getEmotionalThreshold()

8: $\quad$ **if** $(\mathcal{U} \geq \tau_t)$ **then**
9: $\qquad$ **return** CONTROLLABLE
10: $\quad$ **else**
11: $\qquad$ **return** UNCONTROLLABLE

Controllability is also important for the overall architecture. For instance, the robot can choose to ask or negotiate about a collaborative task which is not controllable; it can cause the robot to interpret or predict the other's emotional state (e.g., anger if the task is blocked, i.e., uncontrollable for the other), or form a new motive to establish an alternative goal for the current uncontrollable event. In general, other mechanisms in the architecture use the appraisal process of controllability in their decision making processes; meanwhile controllability uses the information from the collaboration structure, e.g., successful predecessors of a goal.

An important determinant of one's emotional response is the sense of control over the events occurring. This sense of subjective control is based on one's reasoning about self's power. For instance, the robustness of one's plan for executing actions can increase sense of power and subsequently the sense of control. In the collaboration context, we have translated the sense of control into a combination of four different factors including a) *agency* and b) *autonomy* of the robot, as well as the ratios of c) *successful predecessors*, and d) the *available inputs* of a given goal (i.e., g_{ε_t}) in the shared plan.

In Algorithm 4, we compute the controllability of an event based on these four factors (lines 2 to 5). Algorithms 4a to 4d are used to illustrate the underlying processes of all these factors. We use weighted averaging over these four factors to compute the utility of an event in terms of controllability of the event. However, the value of all these weights are set to 1.0 for the purpose of simplicity at this stage of the project. We will adjust these weights after further investigating the influence of these factors, and implementing other mechanisms in the overall architecture. After computing the value of the utility, we compare this value to an emotional threshold similar to what we discussed in Algorithm 1. This comparison leads to our decision about the controllability of an event (lines 8 to 11).

Algorithm 4a (Get Agency Ratio)

function GETAGENCYRATIO(Event ε_t)
  $g_t \leftarrow \mathit{recognizeGoal}(\varepsilon_t)$
  $\mathcal{M}_t \leftarrow \mathit{getActiveMotive}(g_t)$
  if $(\mathcal{M}_t \neq \emptyset)$ **then**
    if $(\mathcal{M}_t \cdot \mathit{type} = \text{INTERNAL})$ **then**
      return 1.0
    else
      return 0.0
  else
    return 0.0

Agency is the capacity of an individual to act independently in any given environment. In a collaborative environment sometimes collaborators are required to act independently of each other. Hence, they need to have some internal motives that are formed based on their own mental state rather than being reinforced by the other. These internal motives will lead the collaborators to acquire new intentions towards new goals whenever it is required. In Algorithm 4a, if there is a path between the belief about the current event and the recognized goal associated with that event, we extract the motive within the acquired path. We consider maximum agency value denoted as α in Algorithm 4 (i.e., $\alpha = 1.0$) if the robot's mental state possesses an internal motive towards the recognized goal; otherwise we consider the minimum agency value (i.e., $\alpha = 0.0$) for no motives or external motives only. Note that the process of forming new internal motives is out of this paper's context.

Autonomy is the ability to make decisions without the influence of others. Autonomy implies acting on one's own and being responsible for that. In a collaborative environment, tasks are delegated to the collaborators based on their capabilities. Therefore, each collaborator is responsible for the delegated task and the corresponding goal. In Algorithm 4, β denotes the value of autonomy with regard to the event (ε_t). This value is the ratio of the number of the contributing goals to g_{ε_t} for which the robot is responsible over the total number of contributing goals to g_{ε_t} (see Algorithm 4b). If the goal associated with the current event

Algorithm 4b (Get Autonomy Ratio)

function GETAUTONOMYRATIO(Event ε_t)
  $g_t \leftarrow \mathit{recognizeGoal}(\varepsilon_t)$
  $\Phi_g \leftarrow \mathit{extractContributingGoals}(g_t)$
  for all $\phi_g^i \in \Phi_g$ **do**
    if $(\mathit{getResponsible}(\phi_g^i) = \text{SELF})$ **then**
      $\mathit{count}_{self} \leftarrow \mathit{count}_{self} + 1$
  return count_{self} / $\Phi_g.\mathit{total}()$

corresponds to a nonprimitive task, the algorithm checks the responsible agent for each primitive task contributing to the nonprimitive one and returns a value of which $(0 \leq \beta \leq 1)$. However, if the associated goal of the current event corresponds to a primitive task the value of β would be 0 or 1. In general, higher autonomy leads to a more positive value of controllability.

Algorithm 4c (Get Succeeded Predecessors Ratio)

function GETSUCPREDECESSORSRATIO(Event ε_t)
  $g_t \leftarrow \mathit{recognizeGoal}(\varepsilon_t)$
  $\Theta g \leftarrow \mathit{extractPredecessors}(g_t)$
  for all $\theta_g^i \in \Theta_g$ **do**
    if $(\mathit{isAchieved}(\theta_g^i))$ **then**
      $\mathit{count}_{achieved} \leftarrow \mathit{count}_{achieved} + 1$
  return $\mathit{count}_{achieved}$ / $\Theta g.\mathit{total}()$

The structure of a shared plan accommodates the order of the required predecessors of a goal. Predecessors of a goal, g, are other goals that the collaborators should achieve before trying to achieve goal g. In Algorithm 4c, we use the ratio of successfully achieved predecessors of the recognized goal (g_{ε_t}) associated with the current event over the total number of predecessors of the same goal. This ratio (denoted as λ in Algorithm 4) is another factor used to compute the controllability of an event. If all of the predecessors of the given goal are already achieved, then $\lambda = 1$ which is the maximum value for λ. On the contrary, failure of all of the predecessors will lead to $\lambda = 0$. Therefore, higher λ value positively impacts the value of controllability for the current event.

Inputs of a task are the required elements that the collaborators use to achieve the specified goal of the task. These inputs are also part of the structure of a shared plan. In Algorithm 4d, we extract the required inputs of the associated goal with the current event, and check whether all the required inputs are available for the goal g_{ε_t}. The outcome will be the ratio of the available required inputs over the total required inputs of the goal associated with the current event. This value (denoted as μ in Algorithm 4) will be bound to 0 and 1. Similar to the other factors in the controllability process, the closer the value of μ gets to 1, the more positive impact it has on the overall controllability value of the event.

Algorithm 4d (Get Available Input Ratio)

```
function GetAvailableInputRatio(Event ε_t)
    g_t ← recognizeGoal(ε_t)
    X_g ← extractInputs(g_t)
    for all χ_g^i ∈ X_g do
        if (IsAvailable(χ_g^i)) then
            count_available ← count_available + 1
    return count_available / X_g.total()
```

7. EVALUATION

We conducted a between subject user study using an online crowdsourcing website – CrowdFlower[1]. Each group originally had 40 subjects. To increase the quality of our subjects' answers, we limited the visibility of our questionnaires to a few English speaking countries, i.e., United States, Britain, and Australia. We also limited our subject pools to those that have acquired the highest confidence level on the crowdsourcing website. Our questionnaires included 2 or 3 test questions (depending on the length) to check the sanity of the answers. We eliminated subjects providing wrong answers to our test questions. We also eliminated subjects with an answering time less than 2 minutes. The final number of accepted subjects in each group is provided in Table 1.

To minimize the background knowledge necessary, we used a simple domestic example of preparing a peanut butter and jelly sandwich, and a hard boiled egg sandwich for a hiking trip. We provided clear textual and graphical instructions for all four questionnaires. The instructions focused on hypothetical collaboration between the test subject and an imaginary friend, Mary; their goal was to prepare two sandwiches. We also provided a brief description as well as a simple example for each appraisal variable, e.g., *relevance*, at the end of the corresponding instructions. We prepared four different online questionnaires for the appraisal variables: *relevance, desirability, expectedness* and *controllability*. All these questions were designed based on different factors that we use in our algorithms (see Section 6). Each question provided three options. One option provided a distinct alternative; another option was used to provide a dichotomy with the first alternative, and a third option was used to check whether the subjects perceived the other two options as equal. Figure 3 shows an example question from one of our questionnaires.

We conducted the user study to compare the results with the implemented algorithms discussed in Section 6. As we mentioned, each question had 3 answers. Therefore, a totally random distribution would result in 33% agreement with our algorithms results. However, the average ratio indicating similarity between human subjects decisions and the output of our algorithms is significantly higher than 33%. The total number of subjects' answers similar to a) the *relevance* algorithm (n=29) averaged 71.3% (s=10.7%), b) the *desirability* algorithm (n=35) averaged 77.8% (s=15.0%), c) the *expectedness* algorithm (n=33) averaged 78.5% (s=12.0%), and d) the *controllability* algorithm (n=33) averaged 74.3% (s=15.8%). It is worth noting that the human subjects agreed 100% on some questions, while one some other questions there was a much lower level of agreement.

[1] http://www.crowdflower.com

Imagine you have pressed the two slices of bread (one covered with strawberry jam and one covered with peanut butter) together and passed it to Mary. Which of the following two actions is **more expected**?

A. Mary puts the given sandwich into a zip lock bag after cutting it in half.

B. Mary puts some pickles on another slice of bread.

C. Equally expected.

Figure 3: An example question from the expectedness questionnaire.

Table 1: Results From Our User Study

appraisal variables	# of subjects	mean	stdev	p-value
Relevance	29	0.713	0.107	<0.001
Desirability	35	0.778	0.150	<0.001
Expectedness	33	0.785	0.120	<0.001
Controllability	33	0.743	0.158	<0.001

The results indicate that our algorithms provide a sufficiently accurate representation of human's appraisal. The p-values obtained based on a one-tailed z-test (see Table 1) show the probability of human subjects' data being generated from a random set. The very small p-values indicate that the data set is not random; in fact, the high percentage of similarity shows it is strongly inclined towards the four appraisal algorithms.

These results indicate that there is a similarity between human subjects' answers and the results from our appraisal algorithm. However, if human subjects thought exactly the same as our algorithms' decision processes, the average number of similarity to our result would be 100%. We believe the difference between these two results could be because of several reasons, including: a) the fact that we conducted our study online and had less control on our subjects, b) our algorithms might require further granularity, c) the difference between decision making processes of individuals, which can be affected by other factors such as personality, gender, and culture. While it maybe possible to achieve a higher level of agreement between humans and the algorithms results, these results indicate that the current algorithms are adequate to be used in collaboration context.

Acknowledgments

This work is supported by the National Science Foundation under award IIS-1012083. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] C. Adam and E. Lorini. A BDI emotional reasoning engine for an artificial companion. In *Workshop on*

Agents and multi-agent Systems for AAL and e-HEALTH (PAAMS), volume 430, pages 66–78. Springer, 2014.

[2] C. Breazeal. *Designing Sociable Robots*. MIT Press, 2002.

[3] P. Cohen and H. J. Levesque. *Teamwork*. SRI International, 1991.

[4] R. Cooke and P. Sheeran. Moderation of cognition-intention and cognition-behaviour relations: A meta-analysis of properties of variables from the theory of planned behaviour. *British Journal of Social Psychology*, 43:159–186, 2004.

[5] A. C. Gonzalez, M. Malfaz, and M. A. Salichs. An autonomous social robot in fear. *IEEE Transactions Autonomous Mental Development*, 5(2):135–151, 2013.

[6] J. Gratch and S. C. Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269–306, 2004.

[7] B. J. Grosz and C. L. Sidner. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 417–444. MIT Press, Cambridge, MA, 1990.

[8] E. Hudlicka. Reasons for emotinos: Modeling emotinos in integrated cognitive systems. In W. D. Gary, editor, *Integrated Models of Cognitive Systems*, volume 59, pages 1–37. New York: Oxford University Press, 2007.

[9] H.-R. Kim and D.-S. Kwon. Computational model of emotion generation for human-robot interaction based on the cognitive appraisal theory. *Journal of Intelligent and Robotic Systems*, 60(2):263–283, 2010.

[10] R. P. Marinier III and J. E. Laird. Emotion-driven reinforcement learning. In *CogSci 2008*, 2008.

[11] S. C. Marsella and J. Gratch. EMA: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90, March 2009.

[12] S. W. McQuiggan and J. C. Lester. Modeling and evaluating empathy in embodied companion agents. *International Journal of Human-Computer Studies*, 65(4):348–360, 2007.

[13] A. Paiva, J. Dias, D. Sobral, R. Aylett, P. Sobreperez, S. Woods, C. Zoll, and L. Hall. Caring for agents and agents that care: Building empathic relations with synthetic agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 194–201, 2004.

[14] M. Pontier and J. F. Hoorn. How women think robots perceive them - as if robots were men. In *International Conference on Agents and Artificial Intelligence (ICAART-2)*, pages 496–504, 2013.

[15] P. S. Rosenbloom, J. Gratch, and V. Ustun. Towards emotion in sigma: From appraisal to attention. In J. Bieger, B. Goertzel, and A. Potapov, editors, *Artificial General Intelligence*, volume 9205 of *Lecture Notes in Computer Science*, pages 142–151. Springer International Publishing, 2015.

[16] K. R. Scherer, A. Schorr, and T. Johnstone. *Appraisal Processes in Emotion: Theory, Methods, Research*. Oxford University Press, 2001.

[17] A. Sloman. Motives, mechanisms, and emotions. *Cognition & Emotion*, 1(3):217–233, 1987.

[18] R. D. Sousa. *The Rationality of Emotion*. MIT Press, 1990.

[19] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[20] J. D. Velàsquez. Modeling emotions and other motivations in synthetic agents. In *Proceedings of the 14th Nnational Conference on Artificial Intelligence AAAI-97*, pages 10–15, 1997.