

Affective Goal and Task Selection for Social Robots

Matthias Scheutz and Paul Schermerhorn
Human-Robot Interaction Laboratory
Cognitive Science Program and School of Informatics
Indiana University
Bloomington, IN 47406, USA
{mscheutz,pscherme}@indiana.edu

Effective decision-making under real-world conditions can be very difficult. From a purely decision-theoretic standpoint, the optimal way of making decisions – rational choice – requires an agent to know the utilities of all choice options as well as their associated likelihoods of succeeding for the agent to be able to calculate the expected utility of each alternative and being able to select the one with the maximum utility. Unfortunately, such rational methods are in practice often not applicable (e.g., because the agent does not have reliable or sufficient knowledge) or feasible (e.g., because it is too time-consuming to perform all necessary calculations).

Psychologists have long hypothesized that humans are able to cope with time, knowledge and other resource limitations by employing *affective evaluations* (Clore, Gasper, & Conway, 2001) rather than rational ones. For affect provides fast, low-cost (although often less accurate) mechanisms for estimating the value of an object, event, or situation for an agent, as opposed to longer, more complex and more computationally intensive *cognitive evaluations* (e.g., to compute the expected utilities) (Kahneman, Wakker, & Sarin, 1997). Humans also rely on *affective memory*, which seems to encode implicit knowledge about the likelihood of occurrence of a positive or negative future event (Blaney, 1986). Finally, affect also influences human problem-solving and reasoning strategies, leading to global, top-down approaches when affect is positive, and local, bottom-up approaches when affect is negative (Bless, Schwarz, & Wieland, 1996).

For (autonomous) social robots that are supposed to interact with humans in natural ways in typically human environments, affect mechanisms are doubly important. For one, such robots will also have to find fast solutions to many of the same kinds of difficult problems that humans ordinarily face, often with the same degree of uncertainty—if not more. Hence, affect mechanisms in robotic architectures might help robots cope better with the intrinsic resource limitations of the real world. The second reason why affect mechanisms are essential for social robots is grounded in their intended role as *social agents* interacting with humans. For those interactions to be *natural* (and effective), robots need to be sensitive to *human affect*, both in its various forms of expression and in its role in human social interactions.

We have started to address affect mechanisms that can serve both functions in our DIARC architecture (Scheutz, Schermerhorn, Kramer, & Middendorff, 2006; Scheutz, Schermerhorn, Kramer, & Anderson, 2007). DIARC is a “distributed integrated affect cognition and reflection” architecture particularly intended for social robots that need to interact with humans in natural

ways. It integrates cognitive capabilities (such as natural language understanding and complex action planning and sequencing (Scheutz, Eberhard, & Andronache, 2004; Brick & Scheutz, 2007; Brick, Schermerhorn, & Scheutz, 2007)) with lower level activities (such as multi-modal perceptual processing, feature detection and tracking, and navigation and behavior coordination (Scheutz, McRaven, & Cserey, 2004; Scheutz & Andronache, 2004)) and has been used in several human subject experiments and at various AAI robot competitions (Scheutz et al., 2005, 2006; Schermerhorn, Scheutz, & Crowell, 2008; Schermerhorn et al., 2006). Most importantly, DIARC incorporates affect mechanisms throughout the architecture, which are based on “evaluation signals” generated in each architectural component, which effectively encode how “good” something (e.g., the current state of the world) is from the perspective of the component.

In this chapter, we will describe DIARC’s mechanisms for affective goal and task selection, and demonstrate the operation of these mechanisms with examples from human-robot interaction experiments.

1 Mood-Based Decision-Making

A perfectly rational agent with perfect information can make optimal decisions by selecting the action A with the highest expected utility $EU = \operatorname{argmax}_A (p_A \cdot b_A - c_A)$, where p_A is the probability of action A succeeding, b_A the benefit of A succeeding, and c_A the cost of attempting A . If the agent knows the costs and benefits of each alternative and also the probabilities of each action succeeding, it cannot be wrong about which is the most profitable choice. In reality, however, costs and benefits are only approximately known. More importantly, real-world constraints can make it difficult to estimate accurately the probabilities of success and failure and, moreover, the dependence of the probabilities on other factors (e.g., past successes and failure).

Rational approaches probabilities that are often not available to robots. Without knowledge of the probabilities of failure and success associated with each potential alternative, it is not possible to calculate expected utility. Humans, on the other hand, are subject to the same kinds of real-world constraints, yet are able to make good evaluations, which are hypothesized to involve affective states (“gut feelings”) in important ways (e.g., to help them prioritize goals).

Let an agent’s overall affective state – its “mood” – be represented by two state variables, one which records positive affect (A_P), and the other of which records negative affect (A_N) (Sloman, Chrisley, & Scheutz, 2005). A_P and A_N are reals in the interval $[0, 1]$ that are influenced by the performance of the agent’s various subsystems (e.g., speech recognition). When a subsystem records a success, it increases the level of positive affect, and when it fails, it increases the level of negative affect. Specifically, success increases A_P by $\Delta A_P = (1 - A_P) \cdot inc$ (failure updates A_N analogously), where inc is a value (possibly learned) that determines the magnitude of the increase within the available range. This update function ensures that A_P remains in the interval $[0, 1]$. Both affective states are also subject to regular decay, bringing their activations in the absence of triggering events back to their rest values (i.e., 0): $\Delta A_P = A_P \cdot dec$ (Scheutz, 2001). Given that affective states can encode knowledge of recent events (e.g., the success or failure of recent attempts), they can be used to estimate probabilities (that take past evidence into account without the need for prior knowledge of the probabilities involved).

Consider, for example, a case in which the robot is deciding whether to ask for directions to

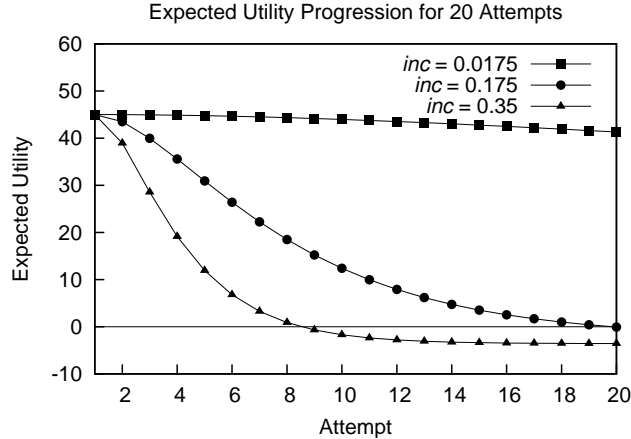


Figure 1: The expected utilities calculated at each attempt by the agent for various values of inc .

some location. The robot does not know that it is in a noisy room where speech recognition is problematic. All else being equal (i.e., with both affect states starting at rest and no affect triggers from other sources), the value of inc determines how many failed communication attempts the agent will make of before giving up. With greater inc , the value of A_N rises faster, leading the agent to reduce its subjective assessment of the expected benefit (i.e., to become “pessimistic” that the benefit will be realized).

The agent makes online choices based on the expected utility of a single attempt, using the affect states A_P and A_N to generate an “affective estimate” of the likelihood of success $a = f(A_P, A_N)$. Examples presented below define f as follows: $f(A_P, A_N) = \frac{1}{2} + \frac{(1+A^{+2}-A^{-2})}{2}$.¹ This value is then used in the calculation of the expected utility of an action: $u = a \cdot b - c$.

The effect of positive and negative affect is to modify the benefit the agent expects to receive from attempting the action. When both A_P and A_N are neutral (i.e., $A_P = A_N = 0$), the decision is based solely on a comparison of the benefit and the cost. However, given a history of actions, the agent may view the benefit more optimistically (if $A_P > A_N$) or pessimistically (if $A_P < A_N$), potentially making decisions that differ from the purely rational choice (overestimating true benefits or costs).

We can now demonstrate with a simple example of how overall mood states could be used in a beneficial way in the agent’s decision making. Figure 1 depicts for the communication example the effect of various values of inc on estimates of utility: one that is too optimistic, willing to continue into the foreseeable future; one that is too pessimistic, stopping fairly early; and one that is more reasonable, stopping at about the point where the costs will outweigh the benefits. This suggests that the value of inc could be defined as a function of b and c to improve the likelihood that A_N will rise quickly enough to end the series of attempts before costs exceed potential benefits, for example. The agent could employ reinforcement learning to determine the value of inc for individual actions.

While the activation of each affective state is subject to decay, the rate of decay is slow enough

¹ A_P and A_N are squared to amplify the difference between the two, which amplifies the effect of the dominant state on the agent’s decision process.

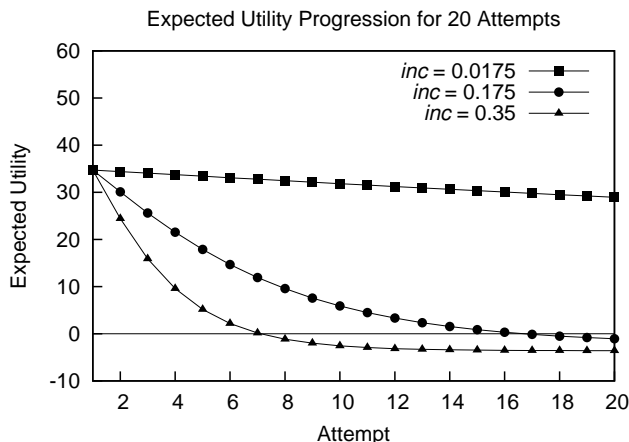


Figure 2: The expected utilities calculated at each attempt by agent for various values of inc , after an extended series of 20 failures and 100 decay cycles, demonstrating the role of affective states as memory.

that they can serve as *affective memory*, carrying the subjective estimates of the likelihood of success and failure ahead for a period after the events that modified the states. Returning again to the robot example, after a series of failures leading to the agent deciding not to attempt to ask directions again, the activation of A_N begins to decay. If, after some period of time, the agent is again faced with the choice of whether to ask for directions, any remaining activation of A_N will reduce the likelihood that it will choose to do so. In this way, the agent “remembers” that it has failed recently, and pessimistically “believes” that its chances of failing again are relatively high (e.g., because it has likely not left the noisy room it was in). Figure 2 shows the expected utility of asking for directions calculated by an agent 100 cycles after a series of failed attempts (e.g., Figure 1). The increased “pessimism” leads the evaluation to drop below zero earlier, potentially saving wasted effort on fruitless attempts.

2 Affect Representations in Architectural Components

We now show how the above decision-making process inspired by roles of human affect, where “affect states” are used to implicitly encode the history of positive and negative events from the agent’s perspective, can be incorporated into an architecture at the level of functional components, where each component maintains its own “affective state”. A primary determinant of the affective state of a component is its own performance, but in some cases the affective states of other functional components (e.g., those upon which it depends to function properly) or the occurrence of certain external events (e.g., a loud unexpected noise) can influence affect.

Specifically, we associate with each component of the architecture two state variables, one which represents *positive affect* (A_P), and the other which represents *negative affect* (A_N). A_P and A_N are reals in the interval $[0, 1]$ and define the “affective evaluation” of that component $a = f(A_P, A_N)$. Examples presented below define f as follows: $f(A_P, A_N) = 1 + A^{+2} - A^{-2}$. The value of a is used by the component when making decisions about how to perform its function.

A component’s affective state values can be passed on to other components to influence the calculation of their respective affect states. Associated with each affective state A_P is an increment variable inc^+ that determines how much a positive event changes positive affect. Specifically, success increases A_P by $\Delta A_P = (1 - A_P) \cdot inc^+$ (this update function ensures that A_P remains in the interval $[0, 1]$; failure updates A_N analogously). The value of inc^+ is computed based on the affective evaluation of connected components: $inc^+ = \sum_{i=1}^n w_i (A_i^{+2} - A_i^{-2})$, for $f(A_{P,i}, A_{N,i}) > 1$, where w_i is the weight assigned to the contribution of component i . Similarly, $inc^- = \sum_{i=1}^n w_i (A_i^{-2} - A_i^{+2})$, for $f(A_{P,i}, A_{N,i}) < 1$. Hence, positive affective evaluation of associated architectural components increases the degree to which positive outcomes influence positive affect A_P for a component, while negative affective evaluation of those components does the same for A_N . Affective states are also subject to regular decay, bringing their activations in the absence of triggering events back to their rest values (i.e., 0): $\Delta A_P = A_P \cdot dec$.

The *affective goal manager* (AGM) prioritizes competing goals (i.e., those whose associated actions require conflicting resources) based on the expected utility of those goals and time constraints within which the goals must be completed. Each goal is assigned an *affective task manager* (ATM), which is responsible for action selection and dispatch. The AGM periodically updates the priority associated with each goal’s ATM. These goal priorities are used to determine the outcome of conflicts between ATMs (e.g., resource conflicts, such as when each wants to move in a different direction). A goal’s priority is determined by two components: its importance and its current urgency. The importance of a goal is determined by the cost and benefit of satisfying the goal. The affective evaluation a of the goal manager influences the assessment of a goal’s importance: $u = a \cdot b - c$. The resulting u is scaled by the urgency component g , which is a reflection of the time remaining within which to satisfy the goal: $g = \frac{Time_{elapsed}}{Time_{allowed}} \cdot (g_{max} - g_{min}) + g_{min}$, where g_{max} and g_{min} are upper and lower bounds on the urgency of that particular goal. The goal’s priority p , then, is simply: $p = u \cdot g$. When there is a conflict over some resource, the ATM with the highest priority is awarded the resource. This formulation allows goals of lower importance, which would normally be excluded from execution in virtue of their interference with the satisfaction of more important goals, to be “worked in” ahead of the more important goals, so long as the interrupted goal has sufficient time to satisfy the goal after the less important goal completes (i.e., so long as the urgency of the more important goal is sufficiently low).

The ATM uses affect states similarly to select between alternative actions in service to a single goal. Each potential action has associated with it (in long-term memory) affect states A_P and A_N that result from positive and negative outcomes in past experience with that action, along with (learned) inc^+ and inc^- that determine how further experience influences the affect state that determine how further experience influences the affect states. The ATM makes online choices based on the expected utility of a single attempt of an action, using $a = f(A_P, A_N)$ as an “affective estimate” of the likelihood of success for the attempt in the utility calculation $u = a \cdot b - c$. The alternative with the highest expected utility is selected in service of the goal associated with the ATM.

The effect of positive and negative affect, then, is to modify the benefit the agent expects to receive from attempting the action. That is, the AGM/ATM implements a decision making process that can operate without exact knowledge of the prior and conditional distributions. When both A_P and A_N are neutral (i.e., $A_P = A_N = 0$), the decision is based solely on a comparison of the benefit and the cost. However, given a history of outcomes, the agent may view the benefit more optimistically (if $A_P > A_N$) or pessimistically (if $A_P < A_N$), potentially leading it to make decisions

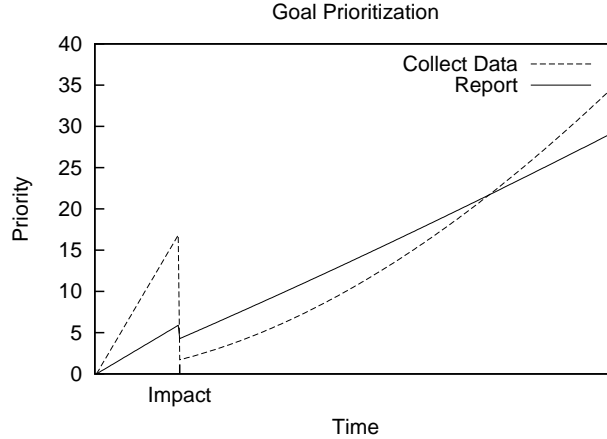


Figure 3: Priorities calculated by the affective goal manager for the goals *Collect Data* and *Report* during a sample run.

that differ from the purely “rational” decision strategy, as mentioned before.

The following two examples presented below focus on AGM and the ATM as they are currently implemented in our robotic architecture. For presentation purposes, simplified scenarios have been chosen to highlight the functionality and benefits of affect in decision-making.

2.1 Prioritizing Goals

The affective goal manager is responsible for prioritizing goals to determine the outcomes of resource conflicts. Priorities are recalculated periodically to accurately reflect the system’s affect states and time-related goal urgencies. In this example, the AGM maintains priorities for two goals, *Collect Data* and *Report*. The *Collect Data* goal requires a robot to acquire information about a region by moving through the environment and taking readings (e.g., for the purpose of mapping locations of interest in the region). There is a limited time within which to gather the data before the robot needs to return with the data. The *Report* goal requires the robot to locate and report to the mission commander once the information is collected *or* when something goes wrong.²

One approach to accomplishing these two goals would be to explicitly sequence the *Collect* and *Report* goals, so that when the former was achieved, the latter would be pursued. The appropriate response to problems could similarly be explicitly triggered when problems were detected. However, the AGM allows for a more flexible unified approach in which both goals are instantiated at the start and the AGM’s prioritization function ensures that the robot does the right thing at the right time. Figure 3 depicts the evolution of the two goals’ priorities throughout a sample run of this scenario. Initially, the AGM’s $A_P = 0$ and $A_N = 0$. The benefit associated with *Collect* (b_c) is 1800, while its cost (c_b) is 1200. The benefit associated with *Report* (b_r) is 200 and the cost (c_r) is 25. Both goals require the use of the robot’s navigation system, but only one may do so at a time.

²This example is taken from the hypothetical space scenario that we have repeatedly used in human-robot interaction experiments (Scheutz et al., 2006), see also Section 3.

At the start, both goals have very low priorities due to the very low urgency (very little time had elapsed). *Collect* has a higher priority due to its greater net benefit ($b - c$); because the AGM's affect is neutral, there is no modification of the benefit component. As time passes, both priorities rise with the increasing urgency until an external event disturbs the system—the impact of an unknown object knocks out a sensor, causing a sharp increase in A_N for the AGM (this could be construed as a fear-like response to the impact event). The AGM output for time step 56 immediately preceding the impact was:

```
AGM A+: 0.0
AGM A-: 0.0
Collect PRIORITY 16.83
Report PRIORITY 5.89
```

Immediately following the impact event, the priorities have inverted:

```
AGM A+: 0.0
AGM A-: 0.5
Collect PRIORITY 1.71
Report PRIORITY 4.28
```

Both priorities were reduced due to the influence of A_N on the benefit component b , but because the reduction of b_c relative to c_c was so much greater than b_r relative to c_r , *Report* was given a higher priority. This allowed the robot to respond to the unexpected impact by seeking the mission commander, who would, presumably, be able to resolve the problem (e.g., by repairing the damage or redirecting the robot). Before the *Report* goal is achieved, however, the priorities were once again inverted (at time step 265), and *Collect* regained control of the navigation resources:

```
AGM A+: 0.0
AGM A-: 0.45
Collect PRIORITY 21.75
Report PRIORITY 21.73
```

This switch is attributable to the decay of A_N in the AGM. No further impacts (or other negative events) occurred and the impact did not cause a catastrophic failure, so negative affect was gradually returning to zero. This (in addition to rising urgency) caused the priorities of both goals to rise, but the priority of *Collect* climbed faster, so that it eventually overtook *Report* and the robot was able to continue pursuing its “primary” goal.³

2.2 Choosing between Alternatives

The affective task manager (ATM) component selects and executes actions on behalf of a goal, as priority allows. When an action completes, the ATM is also responsible for updating the affect states associated with the completed action (based on its completion status, success or failure), in

³Note that there is nothing explicit in the architecture that makes *Report* primary; it is simply the relative costs and benefits of the two goals that make it the preferred goal in the zero-affect state.

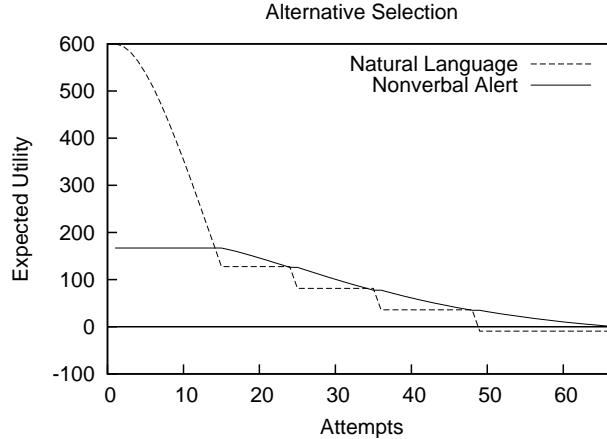


Figure 4: Expected utility of the two alternative actions *Natural Language* and *Nonverbal Alert* throughout a series of failed communication attempts.

addition to updating its own affect states. The following example is extracted from a sample run in which the robot has noticed a problem and needs to communicate it to a human user. There are two modes of communication available: *Natural Language*, in which the robot attempts to explain the problem using natural language, and *Nonverbal Alert*, in which the robot uses “beep codes” to try to convey the message. *Natural Language* has a greater benefit ($b_l = 1800$) than *Nonverbal Alert* ($b_a = 200$), due to the ability to communicate more information about the problem, but also has a greater cost ($c_l = 1200$ vs. $c_a = 25$). Based on past experience, *Nonverbal Alert* has $A_N = 0.2$ (perhaps because of poor results trying to communicate failures using this method). This sample run depicts a series of failed attempts to communicate the problem to the human user (Figure 4).⁴ At the beginning of the run, the ATM output is as follows:

```
Natural Language A-: 0.0
Natural Language UTILITY 600.0
Nonverbal Alert A-: 0.2
Nonverbal Alert UTILITY 167.0
```

The ATM selects *Natural Language* due to its higher expected utility. In the course of the next 14 attempts, u_l (the expected utility of *Natural Language*) falls, while u_a (the expected utility of *Nonverbal Alert*) remains unchanged:

```
Natural Language A-: 0.49
Natural Language UTILITY 173.70
Nonverbal Alert A-: 0.2
Nonverbal Alert UTILITY 167.0
```

After one more failure of *Natural Language*, $u_l < u_a$, so the ATM begins trying *Nonverbal Alert* instead:

⁴Because there are no successful attempts, A_P is not incremented for either action and remains zero throughout the run.

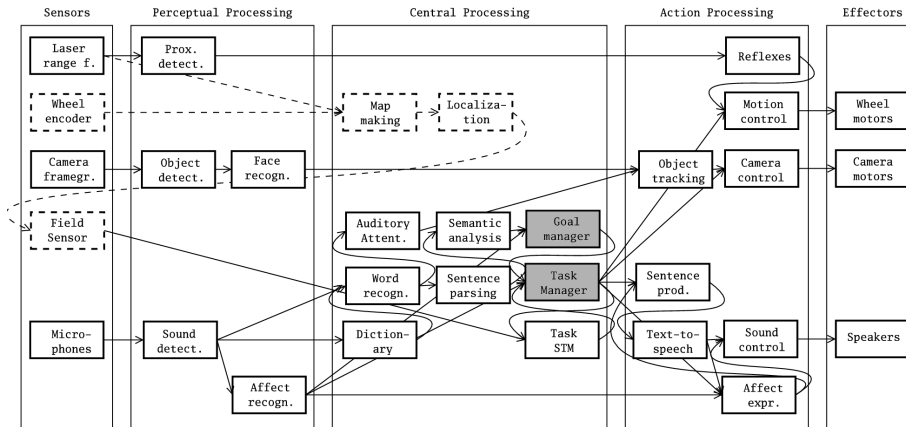


Figure 5: The (reduced) DIARC architecture as used in the human-robot interaction experiments.

Natural Language A-: 0.51
 Natural Language UTILITY 127.54
 Nonverbal Alert A-: 0.2
 Nonverbal Alert UTILITY 167.0

Nonverbal Alert is repeated through attempt 23, and u_a is reduced:

Natural Language A-: 0.51
 Natural Language UTILITY 127.54
 Nonverbal Alert A-: 0.47
 Nonverbal Alert UTILITY 130.96

After attempt 23, the increase in A_N for *Nonverbal Alert* causes its expected utility to fall below *Natural Language*, which is selected on attempt 24:

Natural Language A-: 0.51
 Natural Language UTILITY 127.54
 Nonverbal Alert A-: 0.50
 Nonverbal Alert UTILITY 125.84

Natural Language is attempted only once before the ATM switches back to *Nonverbal Alert*, and the cycle begins again, with the robot occasionally attempting *Natural Language* before reverting to *Nonverbal Alert*, producing the “stair-stepping” effect seen in Figure 4.

3 Human-Robot Interaction Experiments with DIARC

Here we briefly give an example of an application of DIARC for studying affective human-robot interactions (Figure 5 shows the relevant components of the architecture for the given task).

In (Scheutz et al., 2006), we reported an experiment that was intended to examine subjects’ reactions to affect expressed by the robot. Subjects were paired with a robot to perform a task in the

context of a hypothetical space exploration scenario. The task was to find a location in the environment (a “planetary surface”) with a sufficiently high signal strength to allow the team to transmit some data to an orbiting spacecraft. The signal strength was detectable only by the robot, so the human had to direct it around the environment in search of a suitable location, asking it to take readings of the signal strength during the search and to transmit the data once a transmission point was found. There was only one location in the room that met the criteria for transmission, although there were others that represented local peaks in signal strength; the “signal” was simulated by the robot, which maintained a global map of the environment, including all points representing peaks in signal strength. When asked to take a reading, the robot would calculate the signal based on its proximity to these peaks. The goal of the task was to locate a transmission point and transmit the data as quickly as possible; time to completion was recorded for use as the primary performance measure (see (Scheutz et al., 2006) for further details).

Subjects were asked to respond to a series of survey items prior to beginning the interaction with the robot, in order to gauge their preconceived attitudes toward robots (e.g., whether they would think that it was useful for robots to detect and react to human emotions or whether they thought that it would be useful for robots to have emotions and express them). They were given a chance to interact with the robot for a short practice period before the actual experimental runs were conducted. The subjects and the robot communicated via spoken natural language. In order to evoke affective responses from subjects (and to impose an artificial time limit of three minutes on the task), a simulated battery failure was used. There were three points at which the robot could announce problems related to the battery, depending on whether the subject had completed the task or not. One minute into the experimental run, the robot announced that the batteries were “getting low.” After another minute, it would follow with a warning that there was “not much time remaining” due to the battery problem. After three minutes (total), the robot would announce that the mission had failed.

We employed a 2x2 experimental design, with the first dimension, *affect expression*, being *affective* vs. *neutral* and the second, *proximity*, being *local* vs. *remote*. In the neutral affect expression condition, the robot’s voice remained affectively neutral throughout the interaction, while in the affective condition, the robot’s voice was modulated to express increasing levels of “fear” from the point of the first battery warning until the end of the task. Subjects in the local proximity condition completed the exploration task in the same room as the robot, whereas those in the remote condition interacted with the robot from a separate “control” room. The control room was equipped with a computer display of a live video stream fed from a camera in the exploration environment, along with a live audio stream of the robot’s speech (using the ADE robot infrastructure, we were able to redirect the robot’s speech production to the control station). Hence, the only difference between the two proximity conditions was the physical co-location of the robot and the subject. Most importantly, the channel by which affect expression is accomplished (i.e., voice modulation) was presented locally to the subject in both conditions—subjects in the remote condition heard the same voice in exactly the same as they would have if they had been next to the robot.

Subsequent analysis of the objective performance measure (i.e., time to completion) pointed to differences between the local and remote conditions with regard to the effect of affect. A 2x2 ANOVA for *time to completion* with independent variables *affect expression* and *proximity* showed no significant main effects ($F(1, 46) = 2.51, p = .12$ for affect expression and $F(1, 46) = 2.16, p = .15$ for proximity), but a marginally significant two-way interaction ($F(1, 46) = 3.43, p = .07$) due

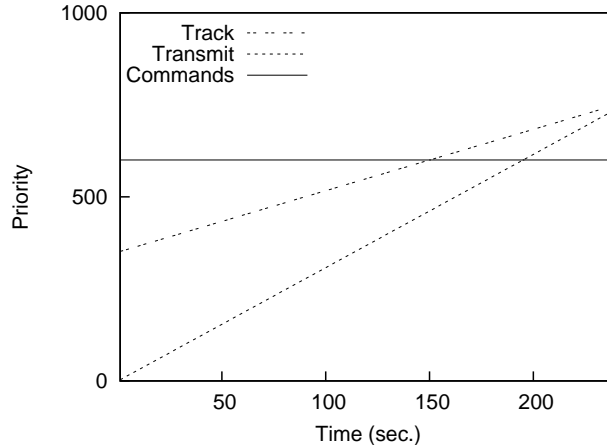


Figure 6: The priority evolution of the dynamic autonomy experiment with neutral starting affect.

to a performance advantage in the *local* condition for *affect* over *neutral* ($\mu = 123$ vs. $\mu = 156$) that was not present in the *remote* condition ($\mu = 151$ vs. $\mu = 150$). The difference in the local condition between affect and no-affect groups is significant ($t(22) = 2.21, p < .05$), while the difference in the remote condition is not significant ($t(16) = .09, p = .93$).

Affect expression provides a performance advantage in the local condition, but not in the remote condition. Given that the medium of affect expression (speech modulation) was presented identically in both proximity conditions, it seems unlikely that the remote subjects simply did not notice the robot’s “mood” change. In fact, subjects were asked on a post-questionnaire to evaluate the robot’s stress level after it issued the low-battery warning. A 2x2 ANOVA with *affect expression* and *proximity* as independent variables, and *perceived robot stress* from the post-survey as dependent variable and found a main effect on affect ($F(1, 44) = 7.54, p < .01$), but no main effect on proximity and no interaction.⁵ Subjects in the *affect* condition tended to rate the robot’s behavior as “stressed” ($\mu = 6.67, \sigma = 1.71$), whereas subjects in the *neutral* condition were much less likely to do so ($\mu = 5.1, \sigma = 2.23$). Hence, subjects recognized the affect expression as they were intended to, and the lack of any effect or interaction involving proximity indicates that both conditions recognized the affect equally well. This, combined with the results of the objective performance task, strongly suggests that affect expression and physical embodiment play an important role in how people internalize affective cues.

A currently still ongoing follow-up study examines dynamic robot autonomy and how affect expression, as described above, influences subjects’ responses to autonomy in the exploration task. The experimental setup is similar (Scheutz et al., 2006), but with an additional “distractor” measurement task included to induce cognitive load in the human team member concurrent to the exploration task. The measurement task consists of locating target “rock formations” (boxes) in the environment and “measuring” them (multiplying two two-digit numbers found on a paper in the box) to determine whether they were above a given threshold.

Dynamic robot autonomy is achieved via three goals in the AGM: *Commands*, which requires

⁵Two subjects had to be eliminated from the comparison since they did not answer the relevant question on the post-survey.

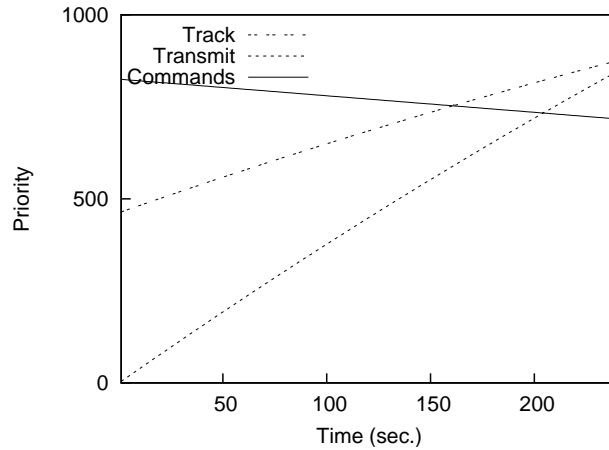


Figure 7: The priority evolution of the dynamic autonomy experiment with positive starting affect.

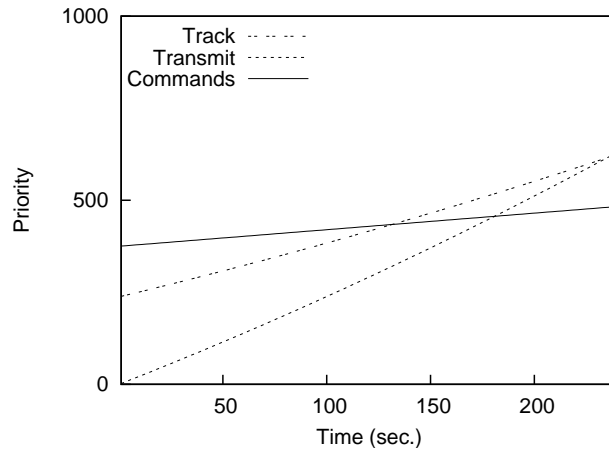


Figure 8: The priority evolution of the dynamic autonomy experiment with negative starting affect.

the robot to obey commands from the human team member, *Track*, which requires the robot to locate and stay with the transmission location, and *Transmit*, which requires the robot to gather and transmit data about the measurements. The priorities and costs were chosen to allow the tracking and transmission goals to overtake the commands goal at specific times. Obeying commands is originally given the highest priority, so that the other goals cannot acquire the resource locks for motion commands, etc. Hence, the autonomy condition starts out exactly as the non-autonomy control condition, with the robot taking commands from the subject related to searching the environment for the transmission location. Then, for example, when the tracking goal's priority surpasses the command goal's (Figure 6), the robot will no longer cooperate with commands that interfere with the robot's autonomous search for the signal peak. These transitions occur at approximately 150 seconds into the task for the tracking goal and 195 seconds for the transmission

goal. This assumes that both A_P and A_N are in their rest states. Figure 7 shows the evolution of priorities for a case in which the robot begins the task with $A_P = .25$ and $A_N = 0$ (e.g., as might be the case if the robot had recently detected positive affect in the voice of the human team member).⁶ The elevated positive affect leads to an “optimistic” assessment of the benefit of following commands (relative to taking over and searching for the transmission location, for example), so the point at which the other goals take over is pushed back (by about ten seconds in either case). An analogous example of the impact of negative affect ($A_P = 0$ and $A_N = .25$) is shown in Figure 8, which shows the “pessimistic” assessment hastening the takeovers by tracking and transmission by approximately 20 and 15 seconds, respectively.

The experimental design includes the *Autonomy* dimension and the *Affect Expression* dimension. This design allows us to explore the degree to which subjects are willing to accept robot dynamic autonomy, and how affect expression on the part of the robot influences the acceptability of autonomy. For example, it seems likely that the robot’s expression of stress as a part of normal speech interactions will provide subjects with some context explaining *why* the robot has stopped following commands, which could facilitate acceptance. We are currently conducting experiments and analyzing the results, having completed the first phase of experiments in a remote condition. As reported in (Scheutz & Schermerhorn, 2009), even without affect expression, subjects are positive with regard to dynamic autonomy in a robotic teammate, to the extent that they even characterize the robot in the autonomy condition as more cooperative than the robot in the non-autonomy condition, despite the fact that the autonomous version disobeyed in the later phase of the task, whereas the non-autonomous version obeyed throughout. We are currently analyzing the remaining data to determine if and how affect expression alters the picture.

4 Related Work

While different forms of affective and deliberative processes (like reasoning or decision-making) have been in simulated agents (e.g., El-Nasr, Yen, & Ioerger, 2000; Elliott, 1992; Gratch & Marsella, 2004), most robotic work has focused on action selection (e.g., Moshkina & Arkin, 2003; Murphy, Lisetti, Tardif, Irish, & Gage, 2002; Parker, 1998; Scheutz, 2002) using simple affective states, often times without explicit goal representations. Yet, complex robots (e.g., ones that work with people and need to interact with them in natural ways (Scheutz et al., 2007)) will have to manage multiple, possibly inconsistent goals, and decide which to pursue at any given time under time-pressure and limited resources.

The two closest affective robotic architectures in terms of using emotion (a form of affective state) for internal state changes and decision-making on robots are (Murphy et al., 2002) and (Breazeal, Hoffman, & Lockerd, 2004). In (Murphy et al., 2002) emotional states are implemented with fixed associated action tendencies (e.g., HAPPY–“free activate”, CONFIDENT–“continue normal activity”, CONCERNED–“monitor progress” and FRUSTRATED–“change current strategy”) in a service robot as a function of two time parameters (“time-to-refill” and “time-to-empty” plus two constants). Effectively, emotion labels are associated with different intervals and cause state transitions in a Moore machine, which produces behaviors directly based on perceptions and emotional states. This is different from the explicit goal representation used in our architecture,

⁶Note that the lines curve slightly due to the built-in decay of affect states.

which allows for the explicit computation of the *importance* of a goal to the robot (based on positive and negative affective state), which in turn influences goal prioritization and thus task and action selection.

The architecture in (Breazeal et al., 2004) extends prior work (Breazeal, 2002) to include natural language processing and some higher level deliberative functions, most importantly, an implementation of “joint intention theory” (e.g., that allows the robot to respond to human commands with gestures indicating a new focus of attention, etc.). The system is intended to study collaboration and learning of joint tasks. The mechanisms for selecting subgoals, subscripts, and updating priorities of goals are, however, different in our affective action interpreter, which uses a dual representation of positive and negative affect that is influenced by various components in the architecture and used for the calculation of the importance, and consequently the priority, of goals.⁷

5 Conclusion

In this chapter, we introduced the idea of integrating affect representations and processing mechanisms throughout a robotic architecture based on psychological evidence that affect permeates the human cognitive system. We present the specific mechanisms integrated in our DIARC architecture, with focus on DIARC’s goal and task managers. We showed with several examples that these mechanisms can lead to effective decisions for robots that operate under time, computation, and knowledge constraints, especially given their low computational cost and knowledge requirements. As such, they can improve the functioning and level of autonomy of social robots. Moreover, we also demonstrated that DIARC can be used for systematic empirical studies that investigate the utility of affect mechanisms for social robots. Specifically, we described results from human-robot interaction experiments where affect expression by the robot in the right context could significantly improve the performance of joint human-robot teams. We also pointed at the potential of DIARC and its affective goal and task management mechanisms for further investigations of the interactions between affect and robot autonomy. Current experiments suggest that these interactions will be particularly important for robots that have to collaborate with humans.

While DIARC has already proven its robustness and applicability in real-world settings, it is still very much “work-in-progress”. We are investigating criteria for situations in which good values for some of the parameters (i.e., the increment and weight values in the affect update equations) can be found. We are also examining ways of making these parameters dependent on goal and task contexts, thus allowing for multiple context-dependent values (which can be learned using reinforcement learning techniques) to overcome the shortcomings of a single value.

References

- Blaney, P. H. (1986). Affect and memory: A review. *Psychological Bulletin*, 99(2), 229–246.
- Bless, H., Schwarz, N., & Wieland, R. (1996). Mood and the impact of category membership and individuating information. *European Journal of Social Psychology*, 26, 935-959.
- Breazeal, C. (2002). *Designing sociable robots*. MIT Press.

⁷The details for reprioritization of goals were not provided in (Breazeal et al., 2004).

- Breazeal, C., Hoffman, G., & Lockerd, A. (2004). Teaching and working with robots as a collaboration. In *Proceedings of aamas 2004*.
- Brick, T., Schermerhorn, P., & Scheutz, M. (2007, October/November). Speech and action: Integration of action and language for mobile robots. In *Proceedings of the 2007 iee/rsj international conference on intelligent robots and systems* (p. forthcoming). San Diego, CA.
- Brick, T., & Scheutz, M. (2007, March). Incremental natural language processing for hri. In *Proceedings of the second acm ieee international conference on human-robot interaction* (pp. 263–270). Washington D.C.
- Clore, G., Gasper, K., & Conway, H. (2001). Affect as information. In J. Forgas (Ed.), *Handbook of affect and social cognition* (p. 121-144). Mahwah, NJ: Erlbaum.
- Elliott, C. (1992). *The affective reasoner: A process model of emotions in a multi-agent system*. Unpublished doctoral dissertation, Institute for the Learning Sciences, Northwestern University.
- El-Nasr, M., Yen, J., & Ioerger, T. (2000). Flame – fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3(3), 219-257.
- Gratch, J., & Marsella, S. (2004). A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, 5(4), 269–306.
- Kahneman, D., Wakker, P., & Sarin, R. (1997). Back to Bentham? Explorations of experienced utility. *Quarterly Journal of Economics*, 112, 375-405.
- Moshkina, L., & Arkin, R. (2003). On TAMEing robots. In *Ieee international conference on systems, man and cybernetics* (Vol. 4, p. 3949 - 3959).
- Murphy, R. R., Lisetti, C., Tardif, R., Irish, L., & Gage, A. (2002). Emotion-based control of cooperating heterogeneous mobile robots. *IEEE Transactions on Robotics and Automation*, 18(5), 744-757.
- Parker, L. E. (1998). Alliance: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 220–240.
- Schermerhorn, P., Kramer, J., Brick, T., Anderson, D., Dingler, A., & Scheutz, M. (2006). Diarc: A testbed for natural human-robot interactions. In *Proceedings of AAI 2006 robot workshop*.
- Schermerhorn, P., Scheutz, M., & Crowell, C. R. (2008, March). Robot social presence and gender: Do females view robots differently than males? In *Proceedings of the third acm ieee international conference on human-robot interaction*. Amsterdam, NL.
- Scheutz, M. (2001). The evolution of simple affective states in multi-agent environments. In D. Cañamero (Ed.), *Proceedings of AAI fall symposium* (pp. 123–128). Falmouth, MA: AAI Press.
- Scheutz, M. (2002). Affective action selection and behavior arbitration for autonomous robots. In H. Arabnia (Ed.), *Proceedings of the 2002 international conference on artificial intelligence* (p. 6 pages). CSREA Press.
- Scheutz, M., & Andronache, V. (2004). Architectural mechanisms for dynamic changes of behavior selection strategies in behavior-based systems. *IEEE Transactions of System, Man, and Cybernetics Part B*, 34(6), 2377-2395.
- Scheutz, M., Eberhard, K., & Andronache, V. (2004). A parallel, distributed, realtime, robotic model for human reference resolution with visual constraints. *Connection Science*, 16(3), 145–167.
- Scheutz, M., McRaven, J., & Cserey, G. (2004). Fast, reliable, adaptive, bimodal people tracking

- for indoor environments. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1340–1352).
- Scheutz, M., & Schermerhorn, P. (2009). Dynamic robot autonomy: Investigating the effects of robot decision-making in a human-robot team task. In *Under review for the 4th acm international conference on human-robot interaction*.
- Scheutz, M., Schermerhorn, P., Kramer, J., & Anderson, D. (2007, May). First steps toward natural human-like HRI. *Autonomous Robots*, 22(4), 411–423.
- Scheutz, M., Schermerhorn, P., Kramer, J., & Middendorff, C. (2006). The utility of affect expression in natural language interactions in joint human-robot tasks. In *Proceedings of the 1st acm international conference on human-robot interaction* (pp. 226–233).
- Scheutz, M., Schermerhorn, P., Middendorff, C., Kramer, J., Anderson, D., & Dingler, A. (2005). Toward affective cognitive robots for human-robot interaction. In *Aaai 2005 robot workshop* (pp. 1737–1738). AAAI Press.
- Sloman, A., Chrisley, R., & Scheutz, M. (2005). The architectural basis of affective states and processes. In J. Fellous & M. Arbib (Eds.), *Who needs emotions? the brain meets the machine*. New York: Oxford University Press.

Terms

DIARC The Distributed Integrated Affect Cognition and Reflection architecture, a control architecture for social robotics that integrates affective mechanisms into reasoning processes.

Positive Affect A DIARC system state that reflects past positive experiences (e.g., successful completion of a task), either overall (for the whole system) or for individual system components.

Negative Affect A DIARC system state that reflects past negative experiences (e.g., task failure or an alarming event), either overall (for the whole system) or for individual system components.

Affective Evaluation The use of system affect states to influence the utility calculation when making decisions. Positive affect tends to produce more “optimistic” evaluations, while negative affect tends to produce more “pessimistic” ones.

Affective Memory Affective states associated with DIARC components (e.g., procedural knowledge) that “keep track” of past positive and negative experiences with that component.

Affective Goal Manager The DIARC component responsible for goal selection and prioritization. The AGM uses affective evaluations of goal utilities when determining which goals have highest priority and hence receive control of system resources.

Affective Task Manager The DIARC component responsible for task selection and execution. The ATM references affective memories associated with potential actions when choosing which ones to schedule to accomplish a task.