Modeling Emotions for Choosing Between Deliberation and Action

Darsana P. Josyula, Franklin C. Hughes, Harish Vadali, Bette Donahue
Department of Computer Science
Bowie State University, Bowie, MD 20715
darsana@cs.umd.edu

***Abstract*—Agents that operate in the real world have to make decisions on how long they can deliberate before they act. If the agent deliberates for too long, the agent may miss a deadline or the environment may change such that the preconditions for acting may no longer hold true. If the agent acts too quickly without proper deliberation, it may miss opportunities or may even perform the wrong action.**

Artificial agents with a metacognitive ability to monitor and influence its deliberation and action can potentially make intelligent decisions on when to stop deliberating and start acting. This research is motivated by the evidence that suggests the role of emotions in the cognitive activities of humans. This paper describes how a model of emotions can be applied to enhance the metacognitive capabilities of an artificial agent to choose between deliberation and action. The emotional model is illustrated within the domain of an air traffic control simulation system.

Keywords: Metacognition, Cognitive Modeling, Intelligent Agents, Fault Diagnosis, Modeling Emotions

I. Introduction

In the field of artificial intelligence agents are tasked with two main job functions: deliberation and action. Deliberation is the process by which agents plan what task to perform in order to accomplish current goals or create new goals. Action refers to the process by which agents actually perform each task in the plan.

In static environments, agents need not be concerned with how long the deliberation step takes provided they are not operating under a deadline, because there is no worry that the environment will change before the agent decides to act. Even if the environment is static, if an agent is working under a tight deadline, it needs to be cautious on how much time is spent deliberating since the time spent on deliberation will reduce the time available for action.

In dynamic environments, agents need to address the decision of how much time to spend thinking about acting versus acting. An agent that deliberates for too long may find that the presuppositions it had made no longer hold true. When it finally finishes with plan formulation and decides to act the world could be in a drastically different state causing the plan to break down or even move the agent further away from its goal. An agent that takes action too quickly may find that its decision to abbreviate the deliberation process has caused it to take actions that do not accomplish its goal in the most effective manner and could possibly hinder progress towards other goals that were not considered during plan formulation.

To address this issue of time management in the domain of robotic soccer, [1] presents a hybrid architecture with a reactive component, a deliberative component and an action selection component. The reactive component suggests an action based on the current state and some heuristics. The deliberative component is a Golog reasoning system that projects a number of candidate plans based on the current world model and returns the best plan for execution. The action selection component chooses between the reactive action and the Golog plan action. Thus, there is always an alternative action available if the Golog reasoning system cannot create a plan in time. The problem with this approach as a general solution is that it attempts to skirt the problem of having the system choose when to act and when to deliberate by providing a default action that is executed every time a well planned solution is taking too long to formulate.

Anytime algorithms [2] offer available solutions and their quality whenever the system needs them. They are iterative in nature and each iteration improves the quality of the solutions. The algorithm estimates the efficiency of a solution as a factor of the amount of time the algorithm has been running. Anytime algorithms are used in systems that need quick solutions sometimes and other times need well thought out solutions. However, they alone are not enough to create the kind of stable autonomous systems required for complex dynamic environments because, again, an anytime algorithm fails to make a decision on how much time can be spent on deliberation before acting. When an anytime algorithm is asked for a solution, it stops its calculations and automatically returns its solution. So, the decision on how long to deliberate is passed on to the process that invokes the anytime algorithm for a solution.

In human beings it is easy to observe evidence of a metacognitive component that uses rationality, logic or pragmatism to transit seamlessly from the deliberation phase to the action phase. Artificial intelligent systems have since attempted to copy this behavior using models of metacognition. These models (for example, [3], [4], [5], [6], [7], [8], [9]) are based on representing estimates of allowable deliberation and action time, monitoring the deliberation and action process, and making adjustments to the processes as deemed necessary. However, these models often overlook a key aspect to human reasoning: *emotion*.

Numerous psychological studies have shown the crucial role emotions play in cognition. Animals that exhibit more emotional behavior tend to be better suited for survival [10]. Emotion has been shown to alter the cognitive process in human beings allowing for different responses to the same problem based on different emotional states. [11] shows how cognition and emotion are intimately entwined. Human beings deprived of their emotional capabilities are often hampered in their cognitive functions [12]. The old model of human thought that places emotion at odds with cognition is being disputed. Indeed human thought may not be possible without the feelings associated with emotion [13]. This idea of emotions actually being a key component in human reasoning is our motivating factor to explore the plausibility of using an emotional model to guide the behavior of smart machines.

Previous uses of emotional modeling (for example: [14], [15] and [16]) in computer systems have largely been focused on the system being able to express emotions or recognize emotions in their users. Much of the research in this area is used in robotic companions such as Pleo—the dinosaur—or to a lesser extent its distant ancestor Furby. Pleo can build an entire personality over its life span and then can be reset to start the entire process over again. Pleo can even become bi-polar if its user teases it and refuses to feed it. This type of behavior may be pleasing in the realm of entertainment but not directly applicable for smart machines designed to run our infrastructure.

[17] presents an emotional robot that plays a game of tug of war. The robot uses its cognitive ability to analyze a person's demeanor and it updates its emotional state based on whether a person's actions are pleasing or displeasing to the robot. If a person's actions are pleasing to the robot, then it develops a friendly emotional state towards the person and hence moves closer to that person.

In this paper, we illustrate how an emotional component can alter the metacognitive processes in beneficial ways in the context of an air traffic control simulation that is already equipped with a metacognitive component.

II. Modeling Emotions for Improved Metacognition

A model of metacognition—MCL [3], [4], [5], [6]— that cycles through 3 different phases: (i) note anomalies, (ii) assess possible reasons for the anomaly's occurrence, and (iii) suggest an appropriate action, has been shown to be effective at handling complex, dynamic environments. MCL maintains expectations on how and when systems should respond as well as how and when the environment should change. These expectations are used to note anomalies that manifest as expectation violations. The anomalies are analyzed to guide appropriate responses into place. If the deliberation process is taking too long an expectation violation occurs; this in turn triggers a response. Examples of responses include act immediately or do more deliberation.

In the MCL model shown in Figure 1, the deviation of the observed values from expected values causes expectation violations. These expectation violations are indications of

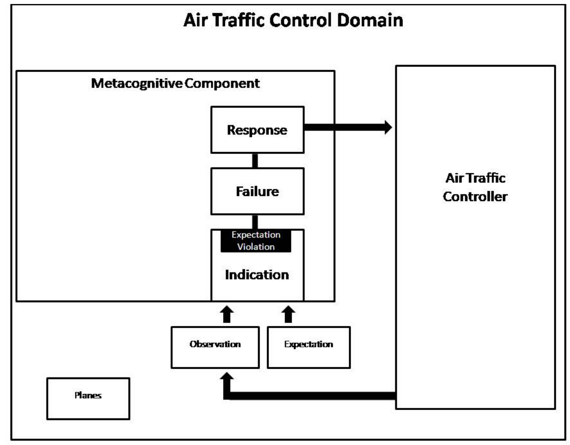

Fig. 1. Model of Metacognition

anomalies. One or more indications map to one or more failures and one or more failures map to one or more responses. The addition of an emotional component on top of this basic MCL will introduce another level of control that can provide a framework for decision-making as shown in Figure 2

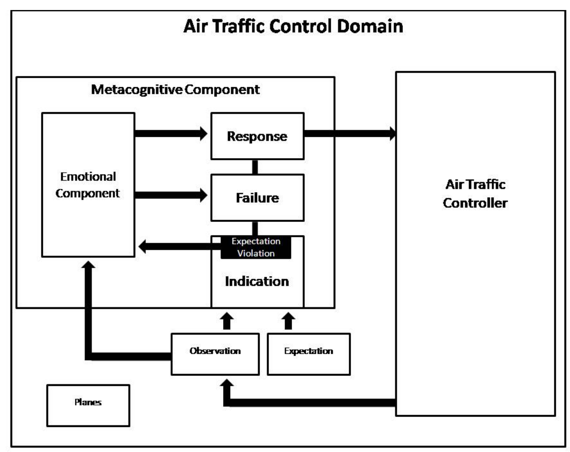

Fig. 2. Model of Metacognition with Emotional Component

During the assessment phase, the metacognitive component looks at all the failure nodes that are activated. The failure nodes have associated probabilities that are programmed in. The emotional model can assign probabilities to the failure nodes based on the current emotional state, thus allowing the system to weigh one failure node heavily in one circumstance and lightly in another.

In the guide phase, the metacognitive component searches through a set of possible responses and instantiates the highest utility response that corresponds to the type of failure. Which action is deemed to be the best is also a learned metric that could be altered for different emotional states; when stressed the best action may simply be the quickest, when relaxed the best action may be the slowest.

A. Emotional Model

Our emotional model for improving metacognition in artificial systems is based on Russell's circumplex model of affects [18], given in Figure 3. The circumplex model claims that all affective states are organised in a circular structure—a "circumplex"—in a two-dimensional plane with axes for degree of pleasure and degree of arousal. The circumplex pattern is confirmed by several different methods for characterising emotion words. Emotions arise from cognitive interpretations of neural sensations that are the product of two independent neurophysiological systems; these neurophysiological systems correspond to the pleasure axis and arousal axis in the circumplex model. The circumplex model of affects is consistent with many recent findings in cognitive neuroscience, neuroimaging, and developmental studies of affects [19]. The circumplex model has been used to study the development of affective disorders as well as the genetic and cognitive underpinnings of affective processing within the central nervous system.

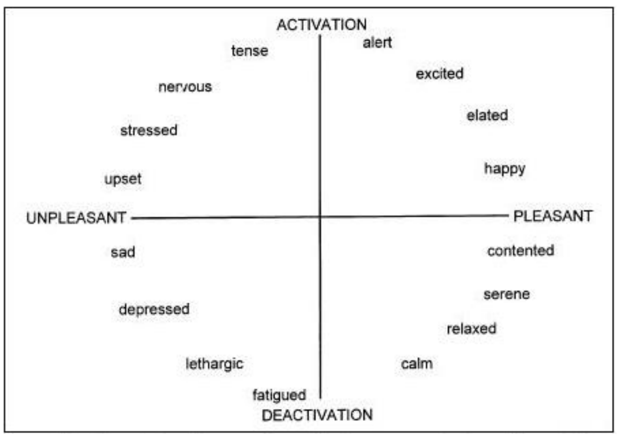

Fig. 3. Russell's Circumplex Model of Positive and Negative Affect

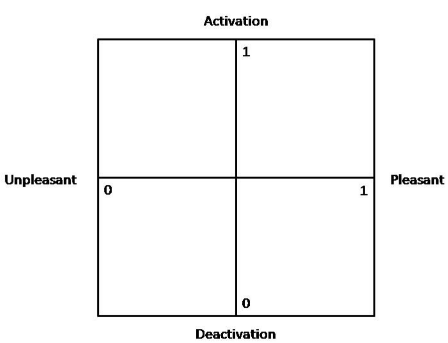

Fig. 4. Emotional Model adapted from Russell's Model

The emotional model that we developed is shown in Figure 4. In this model, the x axis represents the pleasure of the system and the y axis represents the system's activation. The numbers on the axes indicate the level or intensity of pleasure or activation and they range from a minimum of 0 to a maximum of 1.

The pleasure axis transitions are based on the number of expectation violations that occur and represent the system's feelings about its own performance. When no expectation violation occurs, the system is pleased with its performance and hence moves its state to the right on the pleasure axis. When expectation violations occur, the system is frustrated by its inability to quell the violations and hence moves to the left on the pleasure axis. The intensity of the expectation violation (the difference between the observed value and the expected value) decides how far to the left the system moves with respect to its current emotional state.

The activation axis transitions are based on the observations of the system and represent the system's feeling of stress. As the number of observables that the system has to deal with increases, the system becomes more stressed and hence its emotional state moves upward in the activation axis. As the number of observables decrease, the system can relax and hence its emotional state moves downward in the activation axis.

III. Test Domain

The testing domain is that of an air traffic control simulator that has two major components—(i) the ATC that monitors the traffic within a specified radar range and directs aircraft toward available approach paths, and (ii) the aircraft that fly towards the ATC monitored radar area, wait for direction from ATC for an approach path and use that approach path for landing.

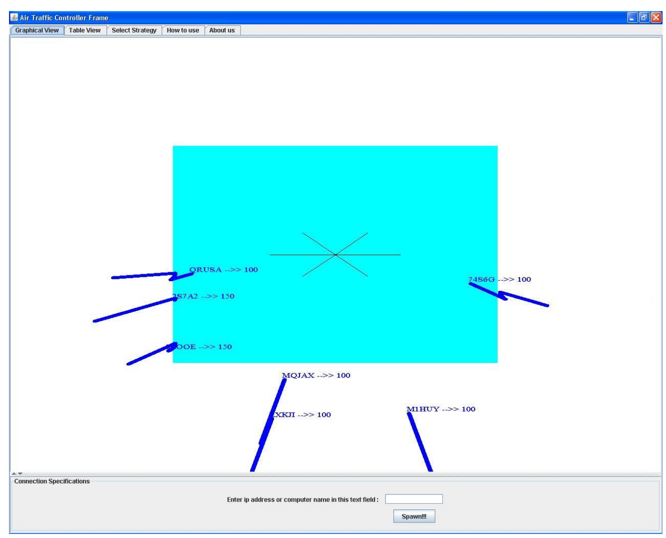

Fig. 5. Trajectories of multiple aircraft flying toward the ATC.

Figure 5 shows the GUI with multiple aircraft beginning their flights toward the ATC. The darkened center rectangle represents the ATC's radar range. The aircraft and their trajectories are represented by the thick dark lines, their current position is marked by the ID/Altitude label. The ATC is located at the center of the radar region. Each of the lines connected to the ATC's location represent one of six approach paths the aircraft must use. Although the number and location of these approach paths can change, shown here are the six default values.

The aircraft are spawned randomly in the region outside the ATC's radar range. The aircraft initiate a connection with the ATC upon creation and are issued a unique ID. Aircraft outside of the ATC's radar range fly under their own guidance until they cross into the area. Once an aircraft crosses the radar range it circles until the ATC consults its current strategy and communicates instructions. Communication between the aircraft and the ATC is accomplished by TCP/IP socket connections. All aircraft land at the ATC's location and must fly there through one of the available approach paths. Once the aircraft lands, its trajectory and current position are erased from the GUI.

The ATC has to deliberate on which approach path needs to be communicated to each aircraft that enters the radar range. This deliberation process needs to be of short duration so that the ATC can direct planes quickly to appropriate approach vectors. To facilitate this, the ATC has a repository of predefined approach path selection strategies and it directs planes based on the currently selected strategy.

The ATC also has a fail-safe collision avoidance mechanism that automatically maintains minimum safe distance zones around each aircraft. The ATC estimates the flight path within the radar range for all aircraft and communicates speed manipulation instructions to one or more aircraft should their predicted flight path intersect.

A. Metacognition in ATC

The metacognitive component of the ATC has various expectations for a number of system variables in the form of threshold values. It observes the ATC's state and if the value of an observable crosses an expectation threshold, it notes that an anomaly has occurred.

The ATC makes its metacognitive component aware of the following observations: (i) aircraft circle times, (ii) flight speeds, (iii) aircraft locations, (iv) the number of times the collision avoidance mechanism is used under the current strategy, (v) the strategies that are available in its repository, and (vi) the performance metrics of each strategy. The expectations of the metacognitive component are as follows:

- Each flight will land within t seconds of coming into the ATC's radar region.
- The aircraft speed should match the speed the ATC assigns to it.
- The collision avoidance system should be used less than k number of times.
- The number of strategies that are available must be greater than x.

If an expectation violation occurs because there are not enough strategies available for choosing terminals, the metacognitive component will trigger the strategy creator to generate new test data in order to create a new strategy by applying various supervised learning algorithms. Once the ATC learns a new strategy, that strategy becomes part of the strategy repository.

If further expectation violation occurs, the metacognitive component can tell the strategy creator to increase the number of data points that it uses to generate training data in order to create a more robust strategy that provides more accurate results.

If an expectation violation occurs because of the circle time being too long or the collision avoidance mechanism being over worked, the metacognitive component will tell the ATC to change to another strategy. The new strategy could be a newly discovered one, one that has not been tried yet or one that has the best performance metrics. The metacognitive component can instruct the ATC to use a specific strategy it knows or delete strategies that do not perform well.

Without an emotional component, the metacognitive component blindly chooses the correct action to take based solely on reasoning based on expectation violations. The illustration in Figure 6 shows how the metacognitive component notices that one of the plane has deviated from its assigned path and triggers a corrective action to fix the situation. In this example, an anomaly was introduced for plane UVY9R using the plane's user interface to alter the flight path away from the nearest approach path to which it was assigned. The metacognitive component notes that the aircraft path is different from what was assigned to plane UVY9R. Upon detection, the metacognitive component recommends that the ATC resend the plane's approach path and then continues monitoring the environment. In this case, the new message from the ATC corrected the problem and the plane changed its flight path to reach the assigned approach vector.

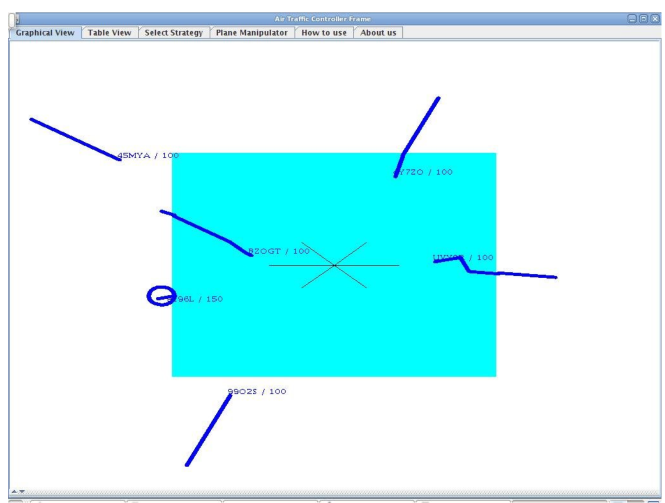

Fig. 6. MCL identified the path violation for plane UVY9R and issued a corrective action.

B. Emotion Processing in ATC

The emotional component of the ATC has access to the observations received by the metacognitive component and the expectation violations noted by the metacognitive component.

The current set of observations and expectation violations determines how emotional state transition occur in the ATC simulator.

Transitions in the activation axis are determined by the normalized values of observables like total number of aircraft with unassigned approach paths, the total number of times the collision avoidance mechanism was used and the flight durations of circling aircraft while there are free approach paths. As these values increase, the system becomes more stressed and hence moves upward on the activation axis. When the values decrease, the system can relax and move downward on the activation axis.

In the ATC system, the activation level can determine the probability of creating new strategies and the size of the training data set; a higher activation level translates to a lower probability for creating new strategies and smaller number of data points in the training set. The system should create new strategies with large training sets when it has the time to complete all the calculations required. When the system is overwhelmed with aircraft the likelihood of spending time and energy creating a new strategy should be lowered and if a new strategy is created it must use a smaller number of data points to save time. During these tense times the system must focus its energy on quickly adopting a strategy that works to some degree and making sure all aircraft land safely. Therefore, the system's location on the activation axis is determined by the values of the system observables such as flight circling duration, number of avoided collisions, number of planes and plane density.

Transitions in the pleasure axis are determined by the number of expectation violations that occur. In the ATC system, the pleasure level can determine the probability that the system changes its strategy for choosing approach paths. The system should change strategies when the current strategy is not performing well. However, every single expectation violation should not cause a strategy switch; for then the agent will be spending more time on deliberation than on action. When an expectation violation occurs, the emotional component pushes the system farther left on the pleasure axis making it more likely to cause a strategy switch rather than causing the switch for each violation.

C. Improved Metacognition by Processing ATC Emotions

In the beginning of the simulation, when the ATC starts no aircraft are spawned. Therefore, the default emotional state is high pleasure and low activation. When there are only a few planes in the air, the system experiences little if any observable changes or expectation violations so the ATC remains happy and relaxed. In this scenario, the impact of the emotional component is not apparent.

When traffic increases, circling time of aircraft and the number of times that the collision avoidance gets used correspondingly increase. With these changes, the system's emotional state in the activation axis will move upward, thereby increasing the chance that the system will create new strategies. Since the activation state is still in the lower hemisphere of the emotional model the number of training data points is large and the likelihood of choosing to create new strategies is high.

Suppose the ATC observes a high rate of expectation violations. If this is combined with low system observables, the ATC will move to a low activation, low pleasure state. In this state, the ATC is likely to change strategies frequently as directed by the displeasure it feels from having its current strategy fail to lower the tide of violations.

If the ATC is in a highly activated state when it decides to change strategies, then it is much more likely to use a stored strategy rather than spend time creating a new strategy as it must focus its energy on guiding the planes and not on mining large tables of data. The ATC will search through its strategy repository for the highest rated strategy and implement it until the environment changes in such a way that lowers its activation, raises its pleasure, or both. If the ATC finds a strategy that works very well under these circumstances it may enter a high activation, high pleasure state that lowers the chances of changing strategies. If instead the ATC never finds a good strategy but the tide of aircraft ebbs, the ATC could enter a low activation, low pleasure state where it will be very likely to change strategies and now that it is no longer stressed will choose to create new strategies with larger training data sets.

In short, the emotional component suspends, overrides or eliminates some of the standard metacognitive responses based on the emotional state of the system. For instance, if the emotional component is present, it would look at the system observables and the level of expectation violations to determine if it has time to start strategy development or if that option needs to be temporarily curtailed.

From time to time the system combs through its strategy repository eliminating strategies that have low evaluations and keeping strategies that perform well. Over time the emotional component and the metacognitive component working together will yield strategies to choose from and increase the overall efficiency of the operation.

IV. Conclusions and Future Work

In conclusion, the emotional model of the system helps the metacognitive component to decide when and how to deliberate and act. In addition, the model itself serves as a way to implement critical system functionality in terms that are easily understandable to users. In other words, we could have programmed this ability into the metacognitive component directly in the form of expectations of expectations but in so doing the program would have gotten much more complex and harder to understand. Using a separate emotional component, we were able to add the same functionality in an intuitive way. The relationship between the two components is straightforward and beneficial. Incorporating an emotional component also makes it a lot easier to drastically alter the behavior of the system merely by changing the transition functions or the metacognitive responses associated with each emotional state.

In the future we would like to expand our emotional model to encompass even more dimensions and act on even more of the system components. Emotional states do not just have to affect deliberation and action. Emotional states could act on input streams allowing the system to prioritize one input over another or ignore input that is only useful in another emotional state. Creating emotional models is a good way to add an intuitive level of control in a system that must be able to change its behavior as its environment changes.

V. Acknowledgements

We would like to thank NSF and NASA for the grants that supported this research. We would also like to thank the undergraduate students (Fassil Molla, Michelle Snowden, Jalissa Miles, Ahmed Kamara and Chinmnso Maduka) in the Robust Intelligence Lab at Bowie State University for their help in various aspects of this research.

References

[1] F. Dylla, A. Ferrein, and G. Lakemeyer, "Acting and deliberating using golog in robotic soccer," in *AAAI Technical Report WS-02-05*, 2002. [Online]. Available: www.aaai.org

[2] N. Vlassis, R. Elhorst, and J. R. Kok, "Anytime algorithms for multiagent decision making using coordination graphs," in *Proceedings of the International Conference on Systems, Man and Cybernetics*, 2004.

[3] M. L. Anderson and D. R. Perlis, "Logic self awareness and self improvement: the metacognitive loop and the problem of brittleness," *Journal of Logic Computation*, vol. 14, pp. 03–44, 2004.

[4] M. Schmill, D. Josyula, M. L. Anderson, S. Wilson, T. Oates, D. Perlis, and S. Fults, "Ontologies for reasoning about failures in AI systems," in *Proceedings from the Workshop on Metareasoning in Agent Based Systems at the Sixth International Joint Conference on Autonomous Agents and Multiagent Sytems*, 2007.

[5] M. L. Anderson, M. Schmill, T. Oates, D. Perlis, D. Josyula, D. Wright, and S. Wilson, "Toward domain-neutral human-level metacognition," in *Papers from the 2007 AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2007. [Online]. Available: http://www.ucl.ac.uk/commonsense07/papers/anderson-et-al.pdf

[6] M. L. Anderson, S. Fults, D. P. Josyula, T. Oates, D. Perlis, M. D. Schmill, S. Wilson, and D. Wright, "A Self-Help Guide for Autonomous Systems," *AI Magazine*, 2008.

[7] M. T. Cox, "Perpetual Self-Aware Cognitive Agents," in *Metacognition in Computation: Papers from 2005 AAAI Spring Symposium*, M. Anderson and T. Oates, Eds. Menlo Park, USA: AAAI Press, 2005, pp. 42–48.

[8] A. Raja and V. Lesser, "A framework for meta-level control in multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 15, no. 2, pp. 147–196, 2007.

[9] S. Fox and D. Leake, "Using introspective reasoning to refine indexing," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, I. C. Mellish, Ed. San Mateo, CA,USa: Morgan Kaufmann, 1995, pp. 391–397.

[10] D. Keltner, "Darwin's touch: Survival of the kindest," *Psychology Today*, February 11 2009.

[11] L. Berkowitz, *Causes and Consequences of Feelings*. Cambridge, England: Cambridge University Press, 2000.

[12] A. Damasio, *Descartes Error: Emotion, Reason and the Human Brain*. G. P. Putnam's Sons, 1994.

[13] J. Lehrer, "Hearts and minds," *The Boston Globe*, April 29 2007.

[14] S. Brave, C. Nass, and K. Hutchinson, "Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent," *International Journal of Human-Computer Studies*, vol. 62, no. 2, pp. 161–178, February 2005.

[15] J. Klein, Y. Moon, and R. W.Picard, "This computer responds to user frustration: Theory, design, and results," *Interacting with Computers*, vol. 14, no. 2, pp. 119–140, 2002.

[16] R. W. Picard and J. Klein, "Computers that recognise and respond to user emotion: theoretical and practical implications," *Interacting with Computers*, vol. 14, no. 2, pp. 141–169, 2002.

[17] D. G. Cooper, D. Katz, and H. T. Siegelmann, "Emotional robotics: Tug of war," in *Association for the Advancement of Artificial Intelligence Spring Symposium on Emotion, Personality and Social Behavior*, 2008.

[18] J. A. Russell, M. Lewicka, and T. Niit, "A cross-cultural study of a circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 57, no. 5, pp. 848–856, 1989.

[19] J. Posner, J. A. Russell, and B. S. Peterson, "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development and psychopathology," *Development and Psychopathology*, vol. 17, pp. 715–734, 2005.