University of Memphis

University of Memphis Digital Commons

CCRG Papers

Cognitive Computing Research Group

1998

An Architecture for Emotion

T. McCauley

S. Franklin

Follow this and additional works at: https://digitalcommons.memphis.edu/ccrg_papers

Recommended Citation

McCauley, T., & Franklin, S. (1998). An Architecture for Emotion. Retrieved from https://digitalcommons.memphis.edu/ccrg_papers/148

This Document is brought to you for free and open access by the Cognitive Computing Research Group at University of Memphis Digital Commons. It has been accepted for inclusion in CCRG Papers by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khggerty@memphis.edu.

An Architecture for Emotion

Lee McCauley and Stan Franklin

Institute for Intelligent Systems
The University of Memphis
Memphis, TN 38152
{t-mccauley, stan.franklin}@memphis.edu

Abstract

The addition of emotions may be the key to producing rational behavior in autonomous agents. For situated agents, a different perspective on learning is proposed which relies on the agent's ability to react in an emotional way to its dynamically changing environment. Here an architecture of mind is presented with the ability to display adaptive emotional states of varying types and intensities, and an implementation, "Conscious" Mattie (CMattie), of this architecture is discussed. Using this architecture, CMattie will be able to interact with her environment in a way that includes emotional content at a basic level. In addition she will learn more complex emotions which will enable her to react to her situation in a more complex manner. A general description is given of the emotional mechanisms of the architecture and its effects on learning are explained.

Introduction

We have reached a point in the evolution of computing where the previously neglected phenomenon of emotion must be studied and utilized (Picard 1997). Emotions give us the ability to make an almost immediate assessment of situations. They allow us to determine whether a given state of the world is beneficial or detrimental without dependence on some external evaluation. For humans, emotions are the result of millions of years of evolution; a random, trial and error process that has given us default qualia and, often, default responses to common experiences. Unlike a reflexive action alone, however, emotions temper our responses to the situation at hand. Simple though that response may be it is this very ability to adapt to a new situation in a quick and non-computationally intensive way that has eluded previous computational agents. Our lives as humans are filled, moment-to-moment with the complex interplay of emotional stimuli both from the external world and from our internal selves (Damasio 1994). Here we'll describe a software agent architecture with the ability to display a full range of emotions and to learn complex emotions and emotional responses. In addition, the importance of emotions for learning in an environmentally situated agent is discussed. The learning of complex emotions is dependent on Pandemonium Theory, which will be described first.

Pandemonium Theory

This architecture is based on a psychological theory called Pandemonium Theory (Selfridge 1959) who applied it only to perception. Later, John Jackson presented it to the computer science community in an extended and more concrete form (1987; Franklin 1995) that makes it useful for control of autonomous agents.

In Jackson's version of Pandemonium Theory, the analogy of an arena is used. The arena consists of stands, a playing field, and a sub-arena. It is also populated by a multitude of "codelets," each a simple agent[1]. Some of the codelets will be on the playing field doing whatever it is they are designed to do; these codelets are considered "active." The rest of the codelets are in the stands watching the playing field and waiting for something to happen that excites them. Of course, what is exciting may be different for each codelet. The more exciting the action on the field is to any particular codelet, the louder that codelet yells. If a codelet yells loudly enough, it gets to go down to the playing field and become active. At this point, it can perform its function. Its act may excite other codelets, who may become active and excite yet other codelets, etc.

Which codelets excite which other codelets is not a random matter; each codelet has associations with other codelets that act much like weighted links in a neural network. The activation level of a codelet (a measure of how loudly it is yelling) spreads down the codelet's association links and, therefore, contributes to the activation level of the receiving codelet. In addition, these associations are not static. Whenever a codelet enters the playing field, the sub-arena creates associations (if they do not already exist) between the incoming codelet and any codelets already on the field. A strong output association and a weaker input association are created between the codelets currently on the playing field and the arriving codelet. The actual strength of the associations depends on

[1] Jackson uses the term "demon" where we use "codelet," a term borrowed from the Copycat architecture (Hofstadter and Mitchell 1994). We do this since not all of Jackson's demons may be so in the customary sense of the word as used in computer science.

a gain value that the sub-arena calculates. In addition to creating these new associations, existing association strengths between codelets on the playing field increase (or decrease) at each time step based on the gain value. Also, multiple codelets that have strong associations with each other can be grouped together, to create a single new codelet called a concept codelet. From the moment of their creation onward, these concept codelets act almost like any other codelet in the system. They differ in that the decay rate of their associations is less, and the amount of time that they spend on the playing field at any one calling is increased.

The sub-arena performs the actual input and output functions of the system as well as most of the automatic maintenance functions. It calculates the gain, a single variable intended to convey how well the agent is performing. Jackson did not specify a mechanism for such an assessment. Surely the assessment must be both domain dependent and goal dependent. Since the gain determines how to strengthen or weaken associations between codelets, how this judgment is arrived at, and how the goal hierarchy is laid out is of considerable importance. The agent accomplishes goal directed behavior only by an *accurate* assessment of its moment to moment status. For humans there is a complex system of sensory labeling and emotional responses, tuned through evolution, which allows us to determine our performance, based on currently active goal contexts.

The current goal context of this system changes dynamically. It can be thought of as emerging from the codelets active at a given time. (How this happens will be described below.) Some high-level concept codelets can remain on the playing field for quite a long time and, therefore, influence the actions of the whole agent for that time. An example of such a high level codelet might be one that tends to send activation to those codelets involved in getting some lunch. Multiple goal contexts can be competing or cooperating to accomplish their tasks.

Emotions

One of the key components of Jackson's system is the gain. It is the gain that determines how link strengths are updated and, consequently, how well the agent pursues its intended goals. Therefore, it is of great importance to understand how the value of the gain is calculated at any given time, and how that value is used. One might view gain as a one-dimensional "temperature" as in the Copycat Architecture (Hofstadter and Mitchell 1994). The introduction of emotions into an agent architecture allows for a more sophisticated assessment of the desirability of the current situation.

A major issue in the design of connectionist models has been how systems can learn over time without constant supervision by either a human or some other external system. Ackley and Littman solve this problem for their artificial life agents by having those agents inherit an evaluation network that provides reinforcement so that its action selection network can learn (1992). In humans, emotions seem to play the role of the evaluation network. As well as affecting our choice of actions, they evaluate the results of these actions so that we may learn. Including emotions in an agent architecture could serve this same purpose.

This dilemma is solved in our architecture by the addition of emotion codelets whose resulting action is the updating of the gain value. The gain is not a single value; instead it is a vector of four real numbers that can be thought of as analogous to the four basic emotions, anger, sadness, happiness, and fear. It is possible that two more elements could be added representing disgust and surprise (Ekman 1992; Izard 1993). However, for our current purposes the four emotions mentioned should suffice. The agent's emotional state at any one time is, therefore, considered to be the combination of the four emotions. A particular emotion may have an extremely high value as compared to the other emotions, and, consequently, dominate the agent's overall emotional state, for example, anger. In such a case the agent can be said to be angry. It is important to note, however, that the agent will always have some emotional state whether it be an easily definable one such as anger or a less definable aggregation of emotions. No combination of emotions are preprogrammed; therefore, any recognizable complex emotions that occur will be emergent.

The value of an individual element (emotion) in the gain can be modified when an emotion codelet fires. Emotion codelets are a subset of dynamic codelets and, therefore, have preconditions based on the particular state or perception the codelet is designed to recognize. When an emotion codelet's preconditions are met it fires, modifying the value of a global variable representing the portion of the emotion vector associated with the codelet's preconditions. A two step process determines the actual value of an emotion at any one time. First, the initial intensity of the emotion codelet is adjusted to include valence, saturation, and repetition via the formula

where

x = the initial intensity of the emotion
v = the valence {1,-1}
x_0 = shifts the function to the left or right

The x_0 parameter will have its value increased when the same stimulus is received repeatedly within a short period of time. The effect of x_0 is the modeling of the short-term habituation of repeated emotional stimuli.

The second step in the process is that each emotion codelet that has fired creates an instantiation of itself with the current value for adjusted intensity and a time stamp. This new instantiated emotion codelet is like a static codelet in that it does not have preconditions and will only be active if other codelets activate it in the normal way. However, this new codelet is special because it will add its adjusted intensity value (not to be confused with activation) to the global variable representing its particular emotion based on the formula (modified from Picard 1997)

where

a = adjusted intensity at creation time
b = decay rate of the emotion
t = current time
t_0 = time at creation of the codelet

When y approaches zero the codelet will stop effecting the emotion vector. Even though the emotion codelet has reverted to acting like a static codelet it can still affect the emotional state of the agent if it becomes conscious. In such a circumstance the codelet will affect the emotional state of the agent using the previous formula adjusted for the new time of activation and with a degraded initial intensity. In this way, remembered emotions can re-effect the emotional state of the system.

There can be multiple emotion codelets, each with its own pattern that can cause it to fire. The system is not limited to the firing of only one emotion codelet at any one time. The resulting emotional state of the agent, represented by the gain vector, is, therefore, a combination of the recent firings of various emotion codelets. Also, multiple emotion codelets can be included in concept codelets, thereby learning complex emotions that are associated with a higher level concept.

Learning via Emotions

It is also important to note how this emotional mechanism changes the way that learning takes place in the system. In most connectionist systems there is a desired output for every input vector. For our architecture, however, there is only desired input. An agent based on this architecture must be situated within an environment and, by its actions, be able to change its environment in a way that it can sense the change (Franklin and Graesser 1997). What this means for learning is that such an agent should be choosing its actions in such a way as to manipulate its environment so that the agent receives the greatest pleasure or avoids displeasure. This is different from the classic reinforcement scheme (Watkins 1989) where a simple positive or negative valence is returned to the system by the environment after an output is produced. Our system, which we call Unsupervised Internal Reinforcement, uses the set of internal emotion codelets to recognize pleasurable and non-pleasurable states of the environment.

Why is this method an advantage over standard reinforcement? For one, the judgement as to whether an output/action is correct is not dependent on some external judge. From the agent's point of view reinforcement, by its definition, can never be unsupervised because the agent is always dependent on this external evaluation. Secondly, in a reinforcement scheme a given output *b* for a given input *a* will always elicit the same reinforcement value. This method only allows the agent to react to its input while our method encourages the agent to manipulate its environment over time to maximize positive valence – pleasure. This allows for multiple positive environmental states as well as multiple paths possible to reach and maintain those states.

The real question for learning has, therefore, become one of how best to maximize pleasure at any one moment as opposed to minimizing the error. It seems fairly obvious that a minimization of error scheme is only useful for omniscient agents whose environment can be completely known either by the agent or by some external evaluation system. For situated agents in a more complex and dynamic environment, however, emotions serve as a heuristic that allows the agent to react to its changing situation in a quick and rational manner.

There has been a great deal of research that indicates that, for humans, emotion is one of, if not the, key element that brings about "rational" behavior (Adolphs 1996; Cytowic 1998; Damasio 1994). The definition of "rational" behavior is important for situated agents. Rational behavior is that behavior that avoids non-pleasurable states and/or pursues pleasurable states. As mentioned previously, emotions for humans have been adjusted and prewired over millions of years of evolution. Even so, many of the decisions that humans make in the course of our daily lives are based on our culture and on those complex learned emotions that are not prewired. How humans manage to learn these complex emotions and how these become coupled to actions is of paramount importance. A promising model is described by Juan Velásquez that is similar to Minsky's K-lines implementation (Velásques 1998; Minsky 1986). Velásquez's model involves associating an emotion to the particular sensory input that it evokes. This association then acts much like an inhibitory or excitatory behavior increasing or decreasing the likelihood that a particular action is chosen.

For our current implementation, the emotion codelets will effect the drives of the system which will, in turn, effect the behavior net (Franklin 1997). However, future work will attempt to determine if complex behaviors can be emergent without the use of explicit drive and goal generation modules.

Conscious Mattie

A version of the architecture described above is being implemented in a system called Conscious Mattie (CMattie). CMattie (Franklin 1997) is the next incarnation of Virtual Mattie (VMattie), an intelligent clerical agent (Franklin et al 1996; Zhang et al 1998; Song and Franklin forthcoming). VMattie's task is to prepare and distribute announcements for weekly seminars that occur throughout a semester in the Mathematical Sciences Department at the University of Memphis. She communicates with seminar organizers and announcement recipients via email in natural language, and maintains a list of email addresses for both. VMattie is completely autonomous, actively requesting information that has not been forthcoming, and deciding when to send the announcements, reminders, and acknowledgements without human intervention. No format has been prescribed for any type of email message to her. CMattie will occupy this same domain but will have a number of additions. For one, the underlying architecture for CMattie will be a version of the pandemonium architecture of Jackson, with modules for metacognition, learning, associative memory, and consciousness. One possible drawback to this domain with respects to the emotion component is that it may not be rich enough to require the emergence of complex emotions.

CMattie is designed to model the global workspace theory of consciousness (Baars 1988, 1997). Baars' processes correspond to codelets from pandemonium theory, and are also called codelets in CMattie. All actions are taken at the codelet level. Baars postulates goal contexts that correspond to higher level behaviors in CMattie. Some action selection is made at the behavior level, and then implemented by the appropriate lower-level codelets. Emotion codelets influence not only other codelets but behaviors as well.

CMattie also has an associative memory capability based on a sparse distributed memory mechanism (Kanerva 1988). A new perception associates with past experiences including actions and emotions. These remembered emotions activate emotion codelets that, in turn, influence current action selection. Thus, CMattie will produce actions, at least partially based on emotional content, and appropriate for the active goal contexts. This is quite in keeping with current research on human decision making using emotions (Cytowic 1998; Damasio 1994).

What sorts of emotional reactions can be expected of CMattie? She may experience fear at an imminent shutdown message from the operating system. She may be annoyed at having reminded an organizer twice to send speaker-topic information with no reply. She may be pleased at having learned the new concept "colloquium."

What effects might these have? Fear may result in CMattie's ignoring another message being processed in her perceptual workspace in favor of quickly saving files. Annoyance may result in a more sharply worded reminder. Pleasure may reinforce her learning activities. All of these influences will be brought about by increasing activation or association or both.

Will CMattie be aware of her emotions? Yes and no. The yes answer results from past emotions appearing in one part of the focus, and of the present emotion in another part. Consciousness routinely shines its spotlight on the focus. Hence, CMattie "experiences" these emotions. The question could also be asked if CMattie would be aware of what emotion she was "experiencing"? Put another way, does the spotlight of consciousness ever shine on the emotion mechanism or on an emotion codelet? Here the answer is no, not because it would be difficult to make it happen, but because we've found no justification for doing so.

Future Work

Some have argued that emotions are the primary force behind rational, goal directed behavior in humans and likely in animals (Cytowic 1998; Damasio 1994; Picard 1997). Cytowic goes so far as to suggest that consciousness, itself is a form or result of emotion (1998). Even at a neurological level, emotion may have a strong influence on which drives are of greatest importance and on which goals we pursue (Cytowic 1998; Damasio 1994; Panksepp 1995). It is our conjecture that emotions are not separate from drives and goals, but that drives and goals are the two ends of a continuum describing the pursuit or avoidance of particular states, which elicit emotional content. What we normally consider drives would occupy the broad end of the continuum while goals are considered to be specific. As an example, consider the general drive to satisfy hunger. A particular goal that would serve that drive might be going to the Burger King™ to get a Whopper™. While many computational systems have used explicit drive and goal modules to account for this behavior, it can also be explained using only remembered emotions influencing behaviors. For this example there would be a remembered satisfaction of no longer being hungry and the enjoyment of the sandwich. These particular emotions influencing the behaviors enact the pursuit of the specific goal. The general group of emotions that come into play when the sensory input occurs that coincides with hunger causes what we consider to be a drive to satisfy hunger. We are not discounting the difference between the terms "drive" and "goal" but the notion that they are separate modules.

An experimental system is being planned that will use emotions to a much greater extent than CMattie. This system will essentially be a connectionist architecture using Jackson's Pandemonium Association Engine (1987). Emotions will serve as the motivations, the drives, and, ultimately, the goals of the system. The purpose of such an experiment is to find out if complex actions can emerge from a system that has only a relatively small set of primary emotion elicitors and no explicit drives or goals.

Related Work

There has been a surge recently in the number of computational and theoretical models in which emotions play a central role (Breazeal 1998; Dahl and Teller 1998; Sloman 1987; Ushida, Hirayama, and Nakajima 1998; Velásques 1997, 1998). These systems have a fair amount of overlap both with each other and with our model. Most notable is the use of the six generally agreed upon primary emotions. Also, these models, along with our own, presume that secondary or learned emotions will be some combination of the primary emotions. The key difference between these approaches and ours lies in the connection of emotions to action selection.

In addition, several systems have been developed that attempt to determine the emotional state of the human user via facial and gesture recognition among other things (Elliot 1992; Reilly 1996). These systems do not internally model emotions. Instead, they represent the emotional state of the user in such a way as to effect the actions of the agent.

Conclusions

The mechanism described here addresses several current issues with the modeling of emotion. It describes a way that complex emotions can be built from basic innate emotions and how all these emotions, innate and basic, effect the constantly changing state of the agent. In light of the fact that all autonomous agents must, ultimately, be situated within an environment, this mechanism proposes a paradigm of learning that should be considered for such systems. CMattie is just such a system. Although her domain may be limiting, she will "experience" emotional content and react to emotional stimuli. CMattie will learn from her experiences and pursue goals reinforced by the emotional valence that the result of those goals elicit.

Acknowledgements

Supported in part by NSF grant SBR-9720314 and by ONR grant N00014-98-1-0332 with help from the other members of the Conscious Software Research Group including Art Graesser, Ashraf Anwar, Myles Bogner, Scott Dodson, Derek Harter, Aregahegn Negatu, Uma Ramamurthy, Hongjun Song, and Zhaohua Zhang.

References

Ackley, David, and Littman, Michael 1992. Interactions Between Learning and Evolution. in Christopher Langton et al., ed., *Artificial Life II,* 487-509. Redwood City, Calif.: Addison-Wesley.

Adolphs, R. Et al. 1996. Neurophychological Approaches to Reasoning and Decision-Making. In: Damasio, A., et al. Eds. *Neurobiology of Decision-Making*. Berlin: Springer-Verlag.

Baars, Bernard 1988. *A Cognitive Theory of Consciousness*, Cambridge: Cambridge University Press.

Baars, Bernard 1997. *In the Theater of Consciousness*, Oxford: Oxford University Press.

Breazeal, Cynthia 1998. A Motivational System for Regulating Human-Robot Interaction. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence,* 54-61. Menlo Park, Calif: AAAI Press.

Dahl, Hartvig, and Virginia Teller 1998. Emotions Just Are Cognitions, In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, 267-272. Mahwah, New Jersey: Lawrence Erlbaum.

Damasio, A. R. 1994. *Descartes' Error,* New York: Gosset/Putnam Press.

Ekman, P. 1992. An Argument for Basic Emotions. In: Stein, N. L., and Oatley, K. Eds. *Basic Emotions*, 169-200. Hove, UK: Lawrence Erlbaum.

Elliot, C. 1992. The Affective Reasoner: A Process Model of Emotions in a Multi-Agent System. Ph.D. Thesis, Institute for the Learning Sciences, Northwestern University.

Franklin, Stan 1995. *Artificial Minds.* Cambridge, MA: MIT Press.

Franklin, Stan 1997. Global Workspace Agents. *Journal of Consciousness Studies*, 4 (4), 322-234.

Franklin, Stan and Graesser, Art 1997. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Intelligent Agents III,* Berlin: Springer Verlag, 21-35.

Franklin, Stan, Art Graesser, Brent Olde, Hongjun Song, and Aregahegn Negatu 1996. Virtual Mattie—an Intelligent Clerical Agent. *AAAI Symposium on Embodied Cognition and Action*, Cambridge MA.

Hofstadter, D. R. and Mitchell, M. 1994. The Copycat Project: A model of mental fluidity and analogy-making. In Holyoak, K.J. & Barnden, J.A. (Eds.) *Advances in connectionist and neural computation theory*, Vol. 2: Analogical connections. Norwood, N.J.: Ablex.

Izard, C. 1993. Four Systems for Emotion Activation: Cognitive and Noncognitive Processes. *Psychological Review* 100(1):68-90.

Jackson, John V. 1987. Idea for a Mind. *SIGART Newsletter*, no. 181 (July):23-26.

Kanerva, Pentti. 1988. *Sparse Distributed Memory*. Cambridge. Mass.:MIT Press.

Minsky, M. 1986. *The Society of Mind*. New York: Simon & Schuster.

Panksepp, J. 1995. The Emotional Brain and Biological Psychiatry. *Advances in Biological Psychiatry*, 1, 263-286.

Picard, Rosalind 1997. *Affective Computing*, Cambridge MA: The MIT Press.

Reilly, S. 1996. Believable Social and Emotional Agents, Technical Report, CMU-CS-96-138, School of Computer Science, Carnegie Mellon University.

Selfridge, O.G. 1959. Pandemonium: A Paradigm for Learning. In *Proceedings of the Symposium on Mechanisation of Thought Process.* National Physics Laboratory.

Sloman, Aaron 1987. Motives, Mechanisms, and Emotions. *Cognition and Emotion*, 1, 3: 217-33.

Song, Hongjun and Stan Franklin 1998. *Action Selection Using Behavior Instantiation*. Forthcoming

Ushida, H., Y. Hirayama, and H. Nakajima 1998. Emotion Model for Life-like Agent and Its Evaluation. In

Proceedings of the Fifteenth National Conference on Artificial Intelligence, 62-69. Menlo Park, Calif: AAAI Press.

Velásquez, J. 1997. Modeling Emotions and Other Motivations in Synthetic Agents. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. Menlo Park, Calif: AAAI Press.

Velásquez, J. 1998. When Robots Weep: Emotional Memories and Decision-Making. *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 70-75. Menlo Park, Calif: AAAI Press.

Zhang, Zhaohua, Stan Franklin, Brent Olde, Yun Wan and Art Graesser 1998. Natural Language Sensing for Autonomous Agents. In Proceeding of IEEE Joint Symposia on Intelligence and Systems, 374-81. Rockville, Maryland.