**Christine Talbot**

**ITCS 6500 - Project**

**Due: 5/4/2010**

# Emotional Contagion CAS Model

## INTRODUCTION

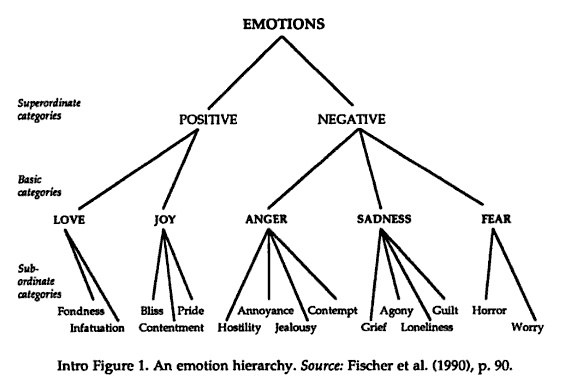
This project tries to model the concept of Emotional Contagion. By definition, emotional contagion theory is “a process in which a person or group influences the emotions or behavior of another person or group through the conscious or unconscious induction of emotion states and behavioral attitudes”, as suggested by Sigal G. Barsade . Emotions are used as a form of communication, and are not a one-way, one-shot type of influence. However, since mass hysteria isn’t a common phenomenon, there must be some sort of controls in place that “self-correct” the escalation of emotions .

Elaine Hatfield, et all describe the types of people that can “infect” others, as well as the ones that are most susceptible to “catching” emotions. The basic hypotheses for the types of people that can “infect” others include:

1. They must feel, or at least appear to feel, strong emotions
2. They must be able to express these strong emotions
3. They must be relatively insensitive to and unresponsive to the feelings of those who are experiencing emotions incompatible with their own.

Therefore, the “energy” of the emotion that is felt drives how likely a person is to infect others with their emotion .

What type of emotion you are feeling can make you more likely to infect others as well. Based on the model proposed by Fischer et al. below, we categorize the “buckets” of emotions. Within these categories of emotions, cognitive social psychologists argue that mood should affect how susceptible you are to “catching” someone else’s emotions. For instance, happy people should find it easier to pay full attention to others, and therefore can pick up on the clues as to the other person’s emotions. This would make them more vulnerable to catching the other person’s emotion .



However, people who are depressed, anxious, or angry may not be able to focus as much on the other person’s clues to their emotion. This would imply that they are less likely to be affected by someone else’s emotion . Along those lines, we can look at our everyday lives and notice that when we have the more extreme negative emotions, it is much harder to change our mood. Based on these observations, we find that what emotion you are displaying will affect your likeliness to “catch” emotions.

When we look at gender, we see a similar kind of effect on a person’s likeliness to “catch” emotions, or their openness. Stereotypically, men are less emotional than women, and women tend to be more comfortable about expressing their emotions. However, men may be less susceptible to “catching” emotions, perhaps because they tend to misinterpret emotions more than women . Based on this information, it appears that women are more open to “catching” emotions than men.

Hatfield et al also found that people who have power over others should be resistant to contagion and those that they control should be more vulnerable to “catching” emotions . From this, we infer that the dominance of the person in say a work environment, will affect how likely someone is to “catch” an emotion.

From this information, I decided to model these four basic components of emotion which may affect a person’s emotional contagion:

1. How strongly they feel an emotion (energy of the emotion)
2. What type of emotion they are feeling
3. How open they are to catching an emotion, with women being more likely to be open / receptive (openness of the person)
4. How dominant the person is compared to the person they are interacting with (dominance comparison)

I also chose to utilize a slight variation of the emotion hierarchy provided by Fischer et al, by adding a somewhat “neutral” emotion of “Ambivalence”. With these attributes, I tried to model the effects of emotional contagion on a somewhat “closed” society / group of people.

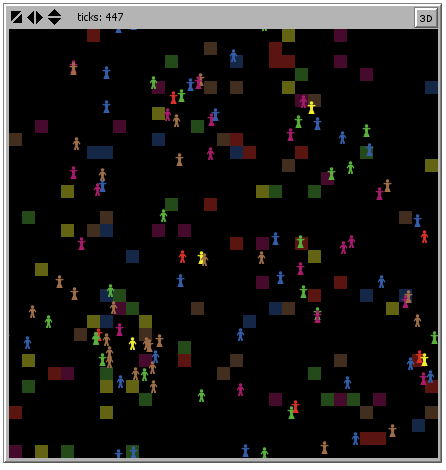
## HOW THE SYSTEM WORKS

Agents are initially created as either male or female. They are also given (randomly) different current emotions, energy of that emotion, a dominence / ranking within the society, and a scale of their openness to others' emotions. Women are given a more likely openness rating than men, based on the slider for the system. Patches are randomly given an emotion to represent a localized "event" that could impact a person's emotions.

As the system runs, the agents move around the world randomly, but will look for people with "similar" emotions within their visible range to move towards if possible, or will move towards the happiest people they can see. If they bump into another agent, OR land on a patch that is currently hosting a random emotion, they will be randomly impacted to change their current emotion because of that event or interaction.

Certain criteria can be turned on/off for determining the effects of an interaction / event, including dominence and openness. Global events can be invoked with a specific emotion to simulate things such as "9/11" or other society-wide events.

Their dominence vs the other agent's dominence rating will affect how likely they are to adjust to the other's emotion (the greater the difference of their dominence ratings & the higher the other agent's rating is, the more likely they are to align closely with the other agent's emotion). Their openness to change will also affect how likely they are to take on the other agent's emotions. And finally, the energy that the patch or other agent is evoking the emotion with will affect the likelihood of modifying their emotion.



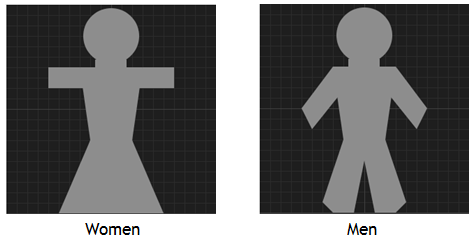
## CAS COMPONENTS

### Agents

* One type of agent – person, with an attribute to identify male or female gender
* # of agents is defined by user (slider) & helps to determine the size of the society
* % men vs women is defined by user (slider)

### Agent Attributes

* Openness
* How likely a person is to “catch” another person’s emotions
* Can be mutated over time based on slider settings
* Women can be made more open than men via a slider as well
* The effects of this attribute can be turned off via a toggle
* Current Emotion
* This is a range that is split into buckets to represent some “basic” emotions such as:
  + Joy
  + Love
  + Ambivalence
  + Sadness
  + Fear
  + Anger
* This can change based on interactions with other agents, landing on a patch with emotion, or when a global event is invoked for the system
* Each emotion is represented by a different color of the agents in the model
* Can be mutated over time based on slider settings
* Dominence Scale
* This determines if there is any hierarchy of people in the society
* Mimics the manager vs reportee kind of impact on emotion contagions
* Can be mutated over time based on slider settings
* The effects of this attribute can be turned off via a toggle
* Current Emotion Energy Level
* This varies with every step of the timer, or change of emotion
* Affects a person’s likeliness to impact another agent’s emotions
* Gender
* Differentiates the agents, primarily around the openness values of the agent
* Used to give the agents a different image:



### Patch Attributes

* Emotion
* This is a range that is split into buckets to represent some “basic” emotions such as:
  + Joy
  + Love
  + Ambivalence
  + Sadness
  + Fear
  + Anger
* This changes with each tick of the clock
* The number of patches with an emotion is controlled by a slider
* Each patch with an emotion is colored a grayed version of the agent’s coloring for the same emotion
* Simulates “local events” that can affect an agent’s emotions
* Emotion Energy Level
* This varies with every step of the timer, in conjunction with the patch’s emotion attribute
* Affects a patch’s likeliness to impact an agent’s emotions

### Agent Behaviors

* Each agent would be randomly generated (randomly set all of its parameters) & be assigned to a location within the environment
* Agents move towards other agents within their visibility (locality and peripheral vision range) who have the emotion closest to theirs, with some randomness included
* Upon interaction, based on their dominance, openness, current emotion, and energy level (if all turned on), the agents adjust their current emotion, with some randomness included
* The patch the agent lands on also affects the agent’s mood, with some randomness, if the patch has an event-emotion tied to it at that time
* Global “events” affect the agent’s emotions, with some randomness

### Environment

* Open environment that introduces the impacts of outside events that can affect emotions via a button and landing on a patch with an emotion
* Random mutation is allowed for within the system and is controlled by sliders
* Locality is applied by visibility of only the agents within a certain range and “peripheral” vision of the person, controlled by sliders
* No resources to consume

### Energy

* Energy of the system is the moods of the agents

### Information

* The information being shared within the system is primarily around moods & emotions

### Feedback Loop

* The feedback loop is how high on the emotional scale the agent is

### Fitness Functions

* Agents – how positive or negative the agent’s mood is
* System – the cumulative view of all the agents’ moods

This ends up being a complex system because the quality of each agent’s decisions affects the whole, but it is uncertain whether it will improve the overall system until it is run. In other words, it is non-linear: the sum of the parts do not make up the whole – you can’t predict what you’ll end up with based on looking at each agent individually.

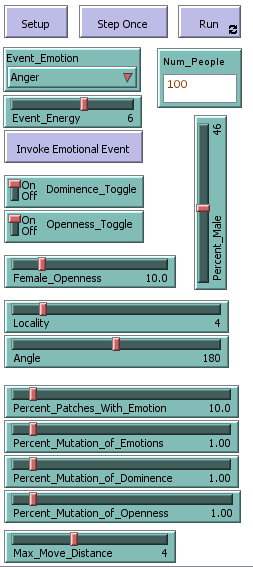
The agents also adapt to their surroundings and learn from each other while interacting and moving throughout the environment. They adjust based on local events, global events, and each other.

The agents also share information when they interact, vying for the more positive emotions. They self-organize by their movements within the system.

I have not really been able to identify any emergence with this system, perhaps because of the way I’ve applied emotions to the agents when they interact. I did notice that I could not get the similar emotions to cluster together as I had thought they might, and that agents appeared to follow a bell-curve for their emotions.

Based on the above details, I believe this to be a complex adaptive system.

## HOW TO USE IT



Setup button - resets the world & creates the agents, their attributes, and patches, initialized

Step Once button - runs one tick of events

Run button - runs step continually

Num\_People - identifies how many people to create in the world (only applied when you click Setup)

Event\_Emotion - drop down to choose the emotion to incur when you click the Invoke Event button (only applied upon clicking Invoke Event button)

Event\_Energy - slider to choose a value from 0-10 for how strongly the emotion should be evoked when you click the Invoke Event button (only applied upon clicking Invoke Event button)

Invoke Event button - Invokes an event of Event\_Emotion & Event\_Energy to all the agents in the system

Percent\_Male - slider to choose percentage of male vs female agents to create (only applied when you click Setup)

Dominence\_Toggle - turns on / off the effects of the dominence trait in the system for applying emotions upon interactions & events

Openness\_Toggle - turns on / off the effects of the openness trait in the system for applying emotions upon interactions & events

Percent\_Patches\_With\_Emotion - determines what percentage of patches should represent a local emotional event (is applied with each tick)

Percent\_Mutation\_of\_Emotions - determines what percentage of agents at each tick should randomly "mutate" their emotion, regardless of whether there is any interaction or event incurred

Locality - determines how far away an agent can see, in order to look for another agent with similar emotions to move towards

Angle - determines the visible arc in front of a person (ie peripheral vision) that they can see, in order to look for another agent with similar emotions to move towards

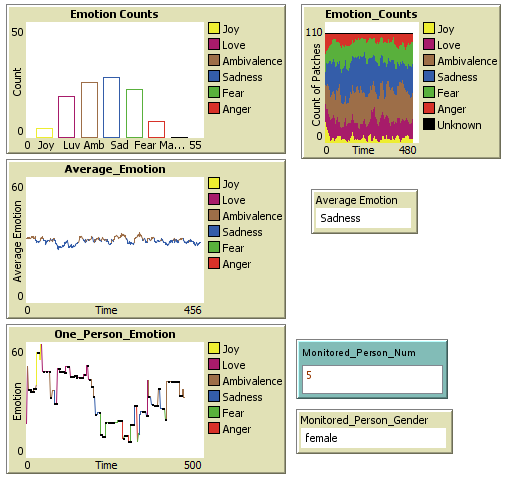
Max\_Move\_Distance - maximum distance that an agent can move on one tick (although it is a random amount up to that distance)

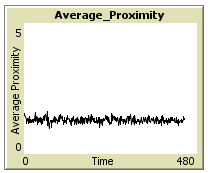
Female\_Openness - Used to determine how much more likely women are to be open than men (only applied when you click Setup)

Percent\_Mutation\_Dominence - determines what percentage of agents at each tick should randomly "mutate" their dominence, to simulate rises & falls of hierarchy

Percent\_Mutation\_Openness - determines what percentage of agents at each tick should randomly "mutate" their openness, to simulate changes due to experiences

## GRAPHS





Monitored\_Person\_Num - who # of the person you want to monitor the emotions for (will clear graph if you change this mid-run)

Monitored\_Person\_Gender - gender of the person with who = Monitored\_Person\_Num

One\_Person\_Emotion Graph - graph of the monitored person's emotion over time - vertical pink lines are global events that are occurring, and black dots are when the agent is interacting with another agent

Average\_Emotion Graph - graph of the average emotion of all agents over time - vertical pink lines are global events that area occurring

Average\_Emotion - shows the name of the average emotion of all agents currently

Emotion\_Counts Graph - shows a bar graph of all agents bucketed into their emotions

Average\_Proximity Graph - shows how close agents are to each other (closest agent is used)

## THINGS TO NOTICE

If you increase the visibility range for the agents, you'll notice that they flock together. Lower visibility ranges do not cause any flocking.

Also, if you invoke an event, not all people are affected, it may take several invocations in order to get everyone to align with that event's mood.

A single person's emotions can vary drastically & spike from one extreme to the other. This may be due to the patches having events & affecting the person.

If you set Locality = 7 & Angle = 360, you can get clusterings of people (I got 4 clusters based upon the point in time where I made the change from a previously started run). This clustering merged into 3 groups after running a little longer with the same settings & when run long enough, incurred a single clustering of agents. However, this occurred primarily when a bug was in the system which spiked people to the extreme emotions.

Looking at the distribution of emotions in the system, you'll notice they follow a relatively bell-shaped distribution, meaning we have more people with middle-ground to neutral emotions most of the time.

## THINGS TO TRY

Turn on / off the dominence or openness attributes from affecting the agents.

Invoke different global emotional events (in rapid sequence, differing emotions, etc)

Change the locality / angle of visibility for each agent (Angle = 360 means they can see all around them, Locality = 26 means they can see the entire board)

Modify the world to allow or disallow wrapping

## THINGS TO DO DIFFERENTLY

Utilize multiple emotion buckets per agent instead of a single range, allowing people to feel more than one emotion at a time, but with an overall “dominant” emotion. This might help alleviate the problem with the bell-curve for the overall emotion buckets.

Not force agents to move toward the most similar agents based on only the emotion. This may need to be based on all the attributes.

Utilize a different algorithm for applying emotions. Currently, the system just checks for dominance, openness, emotion, then energy & applies only one of them independently. Using a weighted algorithm to apply all of the attributes in one calculation would be a more realistic model. I struggled with finding the right algorithm and tried a few different methods, keeping the simplest to implement.

## EXTENDING THE MODEL

Might want to try allowing opposing emotions

Different application of emotions which applies all components in one decision

Use the energy of the emotion to trigger amount of change

Allowing for the reverse emotion to occur

Add graph for how similar the closest person is to each agent.

Might add ability to color agents based on dominence, openness, energy, OR emotion with a click of a button

Allow patches to be colored (or not) based on a toggle

Highlight the person being monitored based on a toggle

Allow people to move towards closest person, and allow them to move past them if moving towards them

Add a graph for how similar (on average) each agent is to their closest neighbor

## NETLOGO FEATURES USED

Use of in-cone for providing only locality within peripheral & in-front of person visibility.

Manually plotting "histogram" to show bar charts over time

Ability to track one person's emotions over time

## RELATED MODELS

None

## REFERENCES

[1] “Emotional contagion - Wikipedia, the free encyclopedia.”

[2] B. Parkinson, A. Fischer, and A.S.R. Manstead, *Emotion in social relations*, Psychology Press, 2005.

[3] E. Hatfield, J.T. Cacioppo, and R.L. Rapson, *Emotional contagion*, Cambridge University Press, 1994.