BASIC: A Believable Adaptable Socially Intelligent Character for Social Presence

Daniela M. Romano, Gary Sheppard, James Hall, Adam Miller, Zhinan Ma
D.Romano@dcs.shef.ac.uk, ac04jh@sheffield.ac.uk, ac04gs@sheffield.ac.uk; ac04am@sheffield.ac.uk, ac04zm@sheffield.ac.uk
Computer Science, University of Sheffield
Regents Court, 211 Portobello Rd, Sheffield S1 4DP
United Kingdom

Abstract

Drawing inspiration from social science and psychology, a computational model of a personality model for a Believable, Adaptable, Socially Intelligent Character (BASIC) has been designed, implemented and tested to drive chimpanzees in a multi-agents scenario. The BASIC model can be customized to create different personalities that are able to trigger empathic responses in human spectators, otherwise known as social presence.

A multi-room event driven scenario, where queues propagate the social interactions amongst the characters, demonstrates the social interaction capabilities of the model embodied within the graphic visual representations. The system is efficient and can run on any mid-spec PC with over ten personalities being fully simulated.

The novelty of the approach lays in the factors combined in the personality model including mood, relationship-based memories, impulse based decision making, and gestural alteration through emotion. These are all steps towards the creation of virtual characters, able to elicit social presence.

Keywords--- **Social presence, multi-agents, synthetic personalities, empathy.**

1. Introduction

The creation of believable artificial characters has been the goal of many researchers in cross-disciplinary fields. Virtual characters are used with the aim of increasing the usability of the human-computer interaction in different ways, in particular in virtual environments to enhance the user experience and trigger social presence.

Biocca [1] defines physical presence, presence in virtual environments, self presence and social presence. Physical presence is the default sense of "being there". It is the basic state of consciousness that people attribute to the source of sensation from the physical environment. The sense of presence in virtual environments instead is a like a daydream in an imaginary environment [1][2]. Social presence has its roots in face-to-face interaction, and social interaction. Social presence is the sense of presence that is felt in mediated communication, where a user feels that a form, behavior, or sensory experience indicates the presence of another intelligence [1]. In this case, the amount of social presence is the degree to which a user feels access to the intelligence, intentions, and sensory impressions of another [1]. Biocca stresses that rather than seeing social presence as an insufficient replication of face-to-face communication, it should be seen as a simulation of another intelligence. Such simulation runs in the body and mind of the perceiver, and models the internal experience of some other moving, expressive body [1].

Furthermore [3] notes that social presence can be linked to a larger social context that includes motivation and social interaction. Social presence is the human ability to project oneself socially and effectively into a community [4].

In this paper we define achieving <u>*social presence*</u> as achieving *the illusion in the mind of the perceiver, that another intelligence exists in the environment. Such illusion is fostered by believable, to the perceiver internal experience, behavioral expressions of the character and its ability to engage in social interactions that trigger empathy in the human user.*

We present here a model and a realization of socially intelligent characters, able to adapt to the environment, with a tested ability to trigger emphatic reactions in the mind of the user.

Unlike other believable agents [5][6], the characters presented here express themselves only through their actions, gestures and facial expressions.

2. Background

Research on the link between realistic virtual characters and virtual presence has undergone some interesting advancement in recent years [7], but these are mostly based on characters as set actors in a preset scene.

2.1 Virtual personalities

The most common representations for modeling virtual humans in psychology are the OCC model of *Emotional State* [8] and *the Five-Factor Model of Personality* [9], both of which lend themselves to computer modeling [10][11][12]. Further work has proved that the full OCC model is not required for believable simulation of characters, and that a Restricted-OCC model, using only 10 of the original 22 emotions, can be reliably used [13]. Whilst not as accurate as the full model, the reduction in computational cost has proven enough to warrant the slight reduction in realism [10]. Kshirsagar [10]. also introduces *Mood* as a personality factor, suggesting that it is required to model slower changes of the current emotion; a concept that we found to be highly viable. Later works [12][14] also introduce memory to the personal state representation, indicating that current reactions and emotions are based on those that have passed in similar conditions. This improves the modeling of memory through the introduction of *Relationships*.

In addition to the emotional build-up of a character, it is also important to consider *Social Knowledge* and it's effects on the actions of an individual. Cervone [15] identifies several key areas of Social Cognitive Theory. They have been used in the personality simulation of our system. In order to co-ordinate social interactions, it is imperative for the agents to have some level of perception of their environment. [16] Rudomin et al. approaches this using a behavior map, but is somewhat limited in its domain. A more substantial event-driven model is proposed in [17] where events are fired into the environment for agents to react. The latter is the perception concept adopted here, as it is a well-proven approach as shown by similar techniques in modern day video games.

2.2 Character visualization

An important factor to consider in the visualization of interactive characters is the Uncanny Valley phenomenon [18], Figure 2.1, where slight inaccuracies in synthetic humans make the observer uncomfortable, reducing believability below what a less realistic representation would induce. An easy way to avoid all probability of being caught in this is to take a cartoon approach in the visual representation. Several systems have used this approach with good affect [14] [19] with minimal loss of user-empathy for the characters.

Considering the need for detailed control of the character a simple Forward Kinematical, skeleton based model is used.

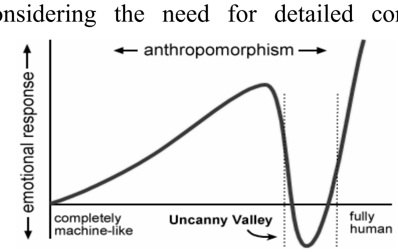


Figure 2.1 – The uncanny valley phenomenon

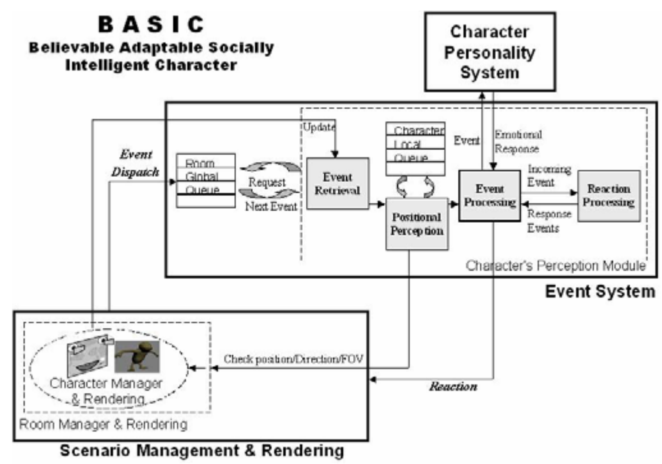
There are many approaches for facial animation from pseudo-bones to point-based manipulation [20]. This first version of the system has no natural language abilities, concentrating on the personality and behavioral simulations, consequently only basic expressions are required. For this purpose, a simplified muscle-based approach (loosely based on [21]) is adopted.

3. System overview

Here is a brief overview of the components that underpin the BASIC system, to aid the reader's understanding in subsequent sections, see figure 3.1.

The system is divided into three core components: Personality, Events System, Scenario Management & Rendering Engine. The primary focus of the research was on the level of believability created by virtual characters by using a dynamic model founded upon emotion, personality, mood, interpersonal relationships and impulse-based decisions. However, a model of this nature cannot inherently display its success without the other components.

Figure 3.1 – BASIC System Overview



The events system drives the interactions between the characters, enabling the passing of messages that inform one character of another's actions. These emotionally coded messages are placed in *queues* and processed by the character's *perception module* of they are directed to and those in the same room. The *event processing module* sends the interpreted event to update the internal status of the personality module receiving an emotionally coded response that is processed by the *reaction processing* sub-modules and consequently generates a reaction that drives the way the character responds and the scene is updated. The behavioral capabilities facilitated by the event system (in italics in figure 3.1) are scripted and external to the source code, to provide a more powerful data-driven simulation.

The personality system of each character receives events as stimuli to its internal state, processing each event it receives with regard to the emotional feelings, mood, personality type, social cognitive factors, and memory of previous interactions (relationship) with the other agents. At each event/interaction, the model generate an emotionally coded response of a certain intensity that the perception module uses to judge what response (if any) should occur.

The scenario management & rendering is required to visually demonstrate the results of our research. A house party, described in section 7, with multiple rooms, each having a specific scripted emotional attribute, has been created for this purpose. The Object-oriented Graphics Rendering Engine (OGRE)[1] has been used for scene management and rendering, populating the environment with bespoken cartoon-type human-like chimpanzee characters, to represent the model without implying any real-world restrictions that a human form would impose.

4. Personality model

The overriding methodology behind the implementation of personality is the dogma discussed in [22] were three factors: *environment, people and behavior* are constantly influencing each other. Behavior is not simply the result of the environment and the person, just as the environment is not simply the result of the person and behavior. The reciprocal nature of the theory has been captured in the model that is described in the following sections. A simple, trait-only, introspective approach to behavior, which does not take into account outside influences, may be able to provide a personality across certain contexts, but in different situations, behavior must vary. Therefore the aim is to produce a character that behaves in a consistent, yet variable manner, across a variety of social contexts learning from its experience of the world, as suggested by [23][24] this is a fundamental capability for autonomous animated agents.

Through the implementation of these theories in a simulated environment that is rich enough to depict whether or not the personality model is successful, we aim to show and evaluate the way that social factors have an impact upon a character's behavior, and ultimately their personality.

4.1 Representations

There are five data structures used in this system based upon three theories. These are represented in Figure 4.1, with the incoming event and the reaction. Three data structures are inspired by the OCC emotional model [8][13] and used to represent emotions, memory of relationships, and mood. The OCC is also used to code the emotional weighting of an incoming event and outgoing response. The second theory used is the five-factor model of personality traits [9] driving the personality, the third; the social cognitive factors [15] driving the behavior in social situations.

The OCC model of emotion is represented by five variables representing one of the emotional opposite pairs (fear/hope, joy/distress, hate/love, anger/pride, and relief/disappointment), ranging between -1 and 1. In addition to the normal range, the value -2 has been used to indicate an extremely negative emotion (i.e. extreme fear) and +2 an extremely positive emotion (i.e. extreme hope). In such cases an additional animation is required as explained in sections 6.1 and 7.

The Five Factors Model (FFM) is used to represent the characters' consistent personality traits that we see as the inner personality of a character, on top of which mood and emotions are built and expressed. The factors represented are agreeableness, neuroticism, extroversion, conscientiousness, and openness. Each of these variables has a different impact on the personality and is represented as a set of values in the range 0 to1.

Social cognitive factors indicate the way in which the social context determines how the character will behave socially. These analyze the environment and other characters therein against six factors: social knowledge, personal goals and standards, reflection about oneself, affective experience, expectation, and self-regulatory skill. They vary the manner in which different characters will interpret an event. Each factor is stored as a variable that can range between 0 and 1, representing the characters' ability to perform each of these cognitive skills.

4.2 Implementation

The five internal representations described above combine into a single personality model that can be distinct for each character in the environment.

The personality system as been conceived as an onion-like model where each layer takes care of one aspect of the personality. Inner layers are more stable that the outer layers to changes, and are represented in a lighter color in Figure 4.1 displaying a section of the personality onion.

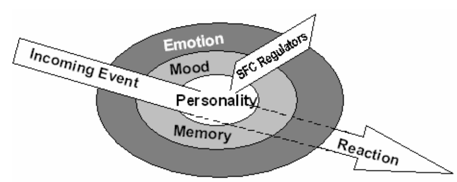


Figure 4.1 – A section of the personality onion with an incoming event passing through the layers (arrow)

The outermost layer is where emotions are formed. These vary easily according to the evolving events. Playing a game might make one happy, a discussion might make the character angry. At the second layer are placed the mood and the memory of relationships. Mood is considered as having a more permanent status than an emotion felt as a consequence of an event in the world. Mood and relationship are less subject to changes due to events in the world. Finally, the Inner layer is the personality coded by the Five Factors Model (FFM). Personality is less likely to change than emotion, memory of relationship and mood, and it is given to a character at creation, but is eventually shaped by the experiences of the world. The Social

[1] www.ogre3d.org

Cognitive Factors (SCF) are regulators that act across the three layers.

An incoming event is filtered through the three layers, see the arrows in Figure 4.1. An interpretation of the event is made at each layer according to the emotional impact, the relationship with the character generating the event, the mood of the character receiving the event and its personality. The incoming event is coded by OCC factors and the response to the event from the personality model is again a OCC factors combinations.

The following subsections define each of the layers and interrelations in detail.

4.2.1 Emotion and mood - The emotional state of a character in the personality model is represented by emotion and mood using the representation discussed above. Emotions and mood are respectively short to mid-term instances of a character's emotional state. The emotion is the more variable of the two, being heavily influenced by events, whereas mood is more involved in the interpretation of how an event is perceived and is a more permanent representation of emotional levels.

4.2.2 Memory - Memory is essential to the social cognitive theory of mind. Memory has an influence on how an event is interpreted, by scaling the values based upon social cognitive factors. Thus, characters that have had a previous experience with other characters will be influenced by their previous interactions. If the character following previous social interactions likes another, it will react in a more positive manner to the actions the latter performs, and vice-versa for negative past experiences. Memory gives you a preconception of the character, a guide as how to act in response to the event generated by it. Memory is implemented as a list of relationship mappings between a character and every other character that it has encountered in the world, see Figure 4.2. The relationship is represented using the OCC categories. Each time a new character is met, a new emotional set of response values is created and added to their memory of relationship. In addition, each time a character receives an event from another character, the memory is dynamically updated to refresh the opinion that the character has based upon the event received and past experience. Memory decays with time, so with the addition of new experiences the older, not updated records are deleted.

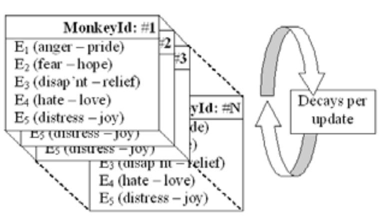


Figure 4.1 – Rapresentation of Memory)

4.2.3 Inner personality - The FFM variables describing the inner personality are given at birth and dynamically updated throughout the duration of the program (life of the character) depending upon the events received. In such manner the character's behavior is shaped by the environment, as described in section 4.3. These are the slowest to change due to external events.

Extroversion is a trait characterized by a keen interest in other people and external events, and venturing forth with confidence into the unknown [25]. In our system, it influences the weight of events in input compared with the mood and previous memory. It has an effect upon the magnitude of the reaction expressed.

Neuroticism is a dimension of personality defined by stability and low anxiety at one end as opposed to instability and high anxiety at the other end [26]. The model implements this as the speed at which the mood can be changed. A character with high neuroticism will quickly change emotion whereas a non-neurotic character will be more calm and stable.

Openness shows how willing people are to make adjustments in notions and activities in accordance with new ideas or situations [27]. The model implements this as a measure of how quickly your mood and FFM traits will changed based upon your perception of the world. A character with a low openness value will not change his personality as quickly as a character with a high openness value.

Agreeableness measures how compatible people are with other people or their ability to get along with others [27]. A character with low agreeableness is likely to have a much more adverse reaction to an event than a character that is highly agreeable.

Conscientiousness refers to how much a person considers others when making decisions [27]. In our model this again is used to determine how a character will respond to events from the environment.

The FFM variables represent long lasting personality traits, and while they have an influence on the way the system behaves, their key function is to help interpret an event in a different manner depending purely upon the character itself. This method of perception allows certain aspects of events to be focussed on by one character whereas another character could derive something completely different from the same event, thus generating diverse characters.

4.2.4 Social cognitive factors – They influence the way in which emotional and personality values interact. These factors are given to the character when they are created and are static through their life.

Reflection is a variable used to determine the degree to which emotion influences mood. When the mood is updated the emotion is weighted by reflection upon oneself. A low reflection value causes a smaller effect in the mood based upon the emotional values, whereas a high reflection will instigate a strong influence (See Eq. 4.5-4.8)

Affective experience is used to determine how much past experiences (memory) affect the current mood. Whenever an event is received, the amount of change in mood, based on the memory of the events instigator, is weighted by this value. A high value means that the memory has a large influence on the mood, a low value means that memory is largely ignored. (See Eq. 4.4 & 4.9)

Social knowledge is used to determine how much past experiences affect the character's interpretation of incoming events and how comfortable the character is in the environment. Social knowledge is used as a weight for the amount of influence that the character's previous emotional state will have on generating a reaction to an incoming event. For an example of the influence of social knowledge, see equations 4.1 and 4.2

Self-regulatory skill is used to regulate the speed at which a characters emotions, mood and memory will return to neutral over time. A character with a high regulatory skill value will have better control over their emotions than a character with a low value. In the system, this is represented as a fall-off weighting (See Eq. 4.10).

Personal goals and standards represent how much a character's personality has influence on how events are interpreted. Higher a character's personal goals and standards, the more they will allow their FFM to influence their world view depending upon their personality. A character with no personal goals and standards will not allow their FFM to have such an effect. (See Eq. 4.3)

Expectation denotes what the character would belief to happen due to a particular course of action. In our model, this is represented as changes of the character's emotional state based on the events they instigate themselves. A higher value causes a greater influence of memory in this process.

4.3 Adaptation

Time and events change a character's given personality at birth. It can change due to events received either from another character or by being in a room, or when the personality is updated by a constant logic tick indicating the passage of time. In such manner the character adapts to the environment it lives in. To facilitate the creation of the personality system, a graphical user interface, the MonkeyBrain, was created at first to simulate all possible inputs from the environment and how they change while the application is running.

There was much discussion and decision making about how the various abstract factors were to be implemented in the system, this was the most crucial point that needed careful consideration. Of course, the best way to evaluate a hypothesis is to test it with the target audience in the target environment. Since the authors aimed at the creation of a general basic model, to be able to see the effects of the different hypotheses the MonkeyBrain personality viewer was created as a means of inspecting the interaction of all the different values in the system. This viewer has been be used to simulate, test, and fine-tune all the different events that were expected to be received by the personality in the course of simulating it in a virtual environment. In the creation of a project such as this, there were several attempts at getting a plausible personality model working correctly. The first approach taken linked the 10 emotional values in a static manner to the five personality factors, in a similar way to [12]. Their approach provides a way of influencing personality based upon the emotions that the character receives. For example, if the character receives a large number of events that elicit an emotional distress, one might expect the personality trait of extroversion to be reduced, and similarly for the other personality factors. We initially drew up the table 4.1 to consider what effects the different emotions would have had on personality, where A = agreeableness, C = conscientiousness, E = extroversion, N = neuroticism, O = openness and a positive, negative or neutral versus indicates how we thought they would interact (respectively +ve, -ve, 0).

OCC Factors	FFM Factors							
	A	C	E	N	O	Total +ves	Total –ves	Total 0s
Anger	-ve	-ve	+ve	+ve	0	**2**	**2**	**1**
Pride	0	-ve	0	-ve	+ve	**1**	**2**	**2**
Disapp.	-ve	0	-ve	0	-ve	**0**	**3**	**2**
Relief	+ve	0	+ve	-ve	0	**2**	**1**	**2**
Distress	0	0	-ve	+ve	0	**1**	**1**	**3**
Joy	+ve	+ve	0	-ve	+ve	**3**	**1**	**0**
Fear	0	0	-ve	0	-ve	**0**	**2**	**3**
Hope	0	+ve	0	0	0	**1**	**0**	**4**
Love	+ve	+ve	0	0	+ve	**3**	**0**	**2**
Hate	-ve	-ve	+ve	+ve	-ve	**2**	**3**	**0**
Total +ves	**3**	**3**	**3**	**3**	**3**	**15**		
Total -ves	**3**	**3**	**3**	**3**	**3**		**15**	
Total 0s	**4**	**4**	**4**	**4**	**4**			**20**

Table 4.1 – Potential Emotional/Personality Links

This first intuitive approach initially led to promising results. The major problem was that unless there were an equal number of emotional events received, the personality trait values tended towards the factor that is most represented. For example, joy/distress is a very simple emotional pair to identify in an event so many events will include a value of joy/distress in their emotional representation. This meant that, as the characters adapted over time, each character in the environment would up with the same personality. In Table 4.1, joy elicits three increases in the personality factors and one decrease, whereas a corresponding emotion of distress elicits one positive and one negative effect upon the personality model. This has the effect of disproportionately increasing the positive factors and they are not returned using the opposite end of the emotional scale.

Consequently the approach described in the section below was taken. Such approach does not statically link one emotion to one FFM factor. It was decided that the totals of the emotions should be considered and used such totals to change the personality traits. This meant that no matter which emotions were invoked, more often an equal change in both the FFM and the resulting interpretation of emotion could be examined over time.

The formulas that drive the personality adaptation are shown in the following sections. The way the weighting system was obtained is explained in section 4.3.3.

4.3.1 Receiving an event – When an event is received it is filtered through several steps, each having a sets of equations, as graphically shown in figure 4.1.

The first step interprets the affects of the event depending upon the FFM, social cognitive factors, mood, memory, and the current emotion. This step also uses a function that blends together the previous memories, mood, and emotion to return an instant reaction to the received event. This means that if the character encounters another it has never met before, memory plays no part in the equation. For each of the emotional affecters in the event, the following equations are used to update the personality model. If the character has no memory of previous interactions with the event source then Equation 4.1 is used, whereas if a relationship is existent then Equation 4.2 is used.

$$Ev' = Ev \cdot (\frac{SC_{SK}}{2})\omega + (Em + M)\omega$$

Equation 4.1 - Step one A function

$$Ev' = Ev \cdot (\frac{SC_{SK}}{2})\omega + (Em + M + Mem)\omega$$

Equation 4.2 - Step one B function

Where *Ev'* and *Ev* are the processed event and incoming event respectively, SC_{sk} is the social knowledge social cognitive factor in question, *Em* is the current emotional value, *M* is the current mood value, *Mem* is the current memory, and ω is a weighting used by the system depending upon the environment and is set empirically.

The second step uses the character's FFM values and the personal goals and standards factor of the SCF, to contribute to the characterization of an event, as shown in Equation 4.3. Four of the five FFMs have an influence upon the event by altering the values depending mostly upon neuroticism to vary the overall strength of the reaction. Personal goals and standards regulate the strengths further.

$$Ev' = Ev \cdot (FFM_N + \omega_1) \cdot (1 - FFM_O\omega_2) + (1 - FFM_C\omega_2) + (FFM_A - \omega_1) \cdot (\omega_3 + \frac{SC_{PGS}}{2})$$

Equation 4.3 – Step two function

Where (excluding terms previously defined), FFM_N, FFM_O, FFM_C and FFM_A are the neuroticism, openness, conscientiousness and agreeableness factors of the FFM respectively, and SC_{PGS} is the personal goals and standards regulator.

Following this, in the third step, the event must again take into account the overall memory, this time considering the affective experience factor. This allows the model to represent how current experiences with other characters in the environment will affect the manner in which the character will perceive an event.

$$Ev' = Ev + (Mem_{Overall} \cdot SC_{AffEx}\omega)$$

Equation 4.4 – Step three function

Where (excluding terms previously defined), SC_{AffEx} is the affective experience of the character and $Mem_{Overall}$ is the overall memory.

Previously we included the memory in the reaction as a targeted relationship between the target and source. In this step, we affect the way that the character perceives the event based upon their collective memory, taking into account long-term memories.

The final step is to return the emotion to the perception as an instantaneous reaction to the event received. The OCC of the character is also updated as a result of the emotion received. Any changes to the personality model as a whole will take affect when the perception next calls the update personality function.

4.3.2 Personality update - With each invocation of the perception module, the personality is updated using the process described in this section. During the personality update, the FFM are updated based upon all the emotional values and a function is used for each of the five factors as shown in figure 4.5, 4.5, 4.7.

$$FFM'_C = (SC_{REF}\omega)FFM_E + \left[\frac{(1 - SC_{REF}\omega)}{2} + FFM_E \sum M\right] + \left[\frac{(1 - SC_{REF}\omega)}{4} \sum (M^2)\right]$$

Equation 4.5 – Conscientiousness update function

Where (excluding terms previously defined), FFM_E is the extroversion of the character and SCF_{REF} is the reflection upon oneself of the character. Conscientiousness tends to be increased when there are high levels of positive emotions in the mood, along with any extreme emotions. Additionally, the extroversion will influence the character to become more or less conscientious. The openness update function works as following. Depending upon the reflection upon oneself in the model, along with current openness, high values of extroversion and the rest of the FFM model will make the character more open to emotion. The same approach is used for agreeableness.

$$FFM'_O = (SC_{REF}\,\omega_1)FFM_O + (1 - SC_{REF}\,\omega_1)FFM_E \sum FFM$$
$$FFM'_A = (SC_{REF}\,\omega_1)FFM_A + (1 - SC_{REF}\,\omega_1)FFM_E \sum FFM$$

Equation 4.6 – Openness and agreeableness update functions

The final two factors in the model are extroversion and neuroticism. The way that these two factors are changed is based upon the sum of the square of emotional values of the mood. By considering these, it allows us to control the model so that the emotional and personality traits do not become too extreme in normal situations.

$$FFM'_N = (SCF_{REF}\omega_1)FFM_N + \left(\left(\frac{FFM_E(1-SCF_{REF}\omega_1)}{2}\right)\sum(M^2)\right)$$
$$FFM'_E = (SCF_{REF}\omega_1)FFM_E - \left(\left(\frac{FFM_E(1-SCF_{REF}\omega_1)}{2}\right)\sum(M^2)\right)$$

Equation 4.7 – Extroversion and neuroticism update functions

The next stage in the update of the personality is to update the mood. As shown in the equation 4.8 This is changed to reflect any changes that have occurred to the emotion, mood, and memory since last time the personality was updated.

$$M' = \left[\left(\frac{SC_{REF}\cdot\omega_1}{2}\right)(1+FFM_O\cdot\omega_2)M\right] + \left[\left(1-\frac{SC_{REF}\cdot\omega_1}{2}\right)(1-FFM_O\cdot\omega_2)Em\right]$$

Equation 4.8 – Mood update function

The reflection about oneself element of the SCF is used to determine how much of the new emotion is to be used in the update to the mood. Next, the mood must be influenced by memory. The average memory of the character is used weighted by affective experience. This changes the way the character feels based upon what experiences it has incurred in its lifetime as shown by the equation 4.9.

$$M' = M + (\omega \cdot Mem \cdot SC_{AffEx})$$

Equation 4.9 – Memory update function

Where (excluding terms defined above) SC_{AffEx} is the affection experience of the character.

Finally, the gradual decline of the characters mood due to time is represented. Depending upon the self-regulatory skill of the character the mood tends towards zero at a different pace using the following decay function.

$$OCC' = OCC(\omega + (1-\omega)\cdot SC_{REG})$$

Equation 4.10 – Mood function update

Where (excluding earlier terms), *OCC'* is the new mood and *OCC* is the current mood. The second decay step is then to update the emotion by substituting the self-regulatory skill of the SCF as the decay function, and the OCC values to those of the emotional state, into Equation 4.10. Finally, the memory is updated in a similar way. This time the decay function is used with the OCC' value set to the original memory, and each relationship's set of affecter values are reduced by the same percentage.

4.3.3 Weights used in the system - The Social Cognitive Factors (SCFs) of each character are set at run time, when a character is born. Characters with different SCFs respond to events, and evolve in different ways. In addition each equation has an associated weight used to vary the amount of overall influence that the calculation has in the engine. The SCFs allow us to change the behavior of the characters whereas the weights make sure that the variability of the behavior is believable and are used to calibrate the system. The weights are stored in an external file, avoiding hard-coding in the system and providing a way to tune the system without the need to re-code and recompile. Such file is called 'weights.ini'. The weights currently used have been determined with usability testing obtaining a finely balanced system. The weights descriptions are provided below covering their use. The most important weights were the values used in the update method. These were especially important, as initially it was not clear how quickly the personality was to be updated by the graphics engine and the perception module. Thus, the weights allowed to easily change the personality engine with respect to time. There are nine weights each with a different use within the system:

W_MEMORY: In the equations that concern memory, this weight varies how much previous encounters influence a reaction. This particular weight only influences equations that are used in the receive event process.

W_FFM: When an event is received, it is interpreted using the FFM. This weight has the effect of increasing or lowering the overall effect that the personality has on the interpretation of said events.

W_MOOD: When an action is received, the mood has an influence on how it is perceived. By increasing or decreasing this weight, the character will effectively become more moody.

W_MOOD_UPDATE: Every time the character is updated, there are several weights that are used to determine the amount of influence that each particular equation has in the module. This update is used to decide how much influence the mood has on the emotion every time the personality is updated.

W_FFM_UPDATE: Determined how much the FFM influences the mood every update.

Furthermore there are three regulatory weights. These are used to help determine how quickly a character's particular emotional traits return to a neutral value. The are: W_MOOD_REG: How fast the mood returns to normal over time. W_EMOTION_REG: How fast emotions return to normal over time. W_MEMORY_REG: How quickly memory deteriorates over time.

W_EVENT_RECEIVE: This decides how much of the reaction to actually return to the perception module. It represents how much the current emotion is affected by the event.

W_EXPECTATION: When you perform an event, the event has an influence on your personality. How much of an effect is influenced by the value of this weight.

5. Situation awareness

The personality model is a self-contained system that requires external events to evolve and produce emotionally coded responses. Such computational model has to be linked to a visual representation (scenario management & rendering) to show its effects and feed by an event system to inform & drive the character's social interactions by sending an event to a specific character or propagated it in a room. The event system represents the way the characters have a situation awareness. The event system is broadly composed by the scenario *rooms' global queues,* and the *character's perception module*.

Each character has a perception module that handles all incoming events providing events in response termed *reactions,* see Figure 3.1. To enable flexibility, customization and extensibility in the incoming-event/response, this relationship is scripted and independent of program compilation.

On a character update, the perception module retrieves all pertinent events from the global queue of the room in which the character is located. Each of these events is checked for *positional relevance* by checking relative position, direction and the field of view of the current character against the source of the event. If the source is a room, this test always returns success. The event is than processed though the *event* and *reaction processing* sub-modules. They inform the personality module of the new event updating its configuration accordingly and receiving an emotionally coded response that will update the scenario management & rendering.

Events have a time-out value, as an event might be only relevant for a certain period of time. This is useful to cope with cases where an event arrives before one character group is ready to process it and after such period the event is might be no longer relevant. The time-out is handled by the room global queue as it will be explained in the following section.

5.1 Global, local queues and priority order

Two types of event queue are used by the system: a global queue for each of the room in the environment and a local queue for each character.

The room global queue is the collection point for all the events that are relevant to that room generated by any character and the room itself (case of character leaving or entering a room described in the following section). Events added to the global queue are not immediately available for retrieval by the characters in the room, but are buffered, to prevent a scenario whereby an event is not received by all relevant characters, as shown in figure 5.1.

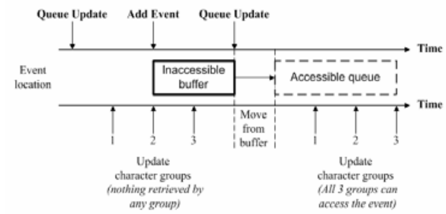


Figure 5.1 – Updating the global queue

If events were broadcast immediately, not all characters might have the chance to respond to them. Characters are updated in bursts to ensure that the rendering system is accessed frequently and with similar duration interim time differences. It must be noted that a room global event queue is effectively composed by a buffer and various separate sub-queues, one for each character that has registered with the room by entering in it. After the main loop of the program has performed updates on all the room's characters, it signals the event queue to update itself. Consequently all the events in the buffer are moved to an accessible area and reside there until the next queue update signal (broadcast), even if the timeouts of the events indicate event expiration. At the next queue update call all events that have timeout are removed. Broadcast signals are different from normal events because they have no specific target. When a broadcast signal is received by a room global queue, it inserts all relevant events into each of the character's room global sub-queues so that it is fairly processed by all characters.

Another type of queue exists in the perception module of each character, termed local queue.

A character local queue is coupled with the positional processing sub-module of a character and it is used as a temporary store for incoming events retrieved from the global queue that pass the positional tests. Local queues are far simpler than their global queue counterparts doing no special buffering.

Regardless of the type of queue, an order exists amongst the events based upon the priority value of each event. Priority values are: LOW, NORMAL, HIGH or CRITICAL and are assigned to an event at its creation in the script. The priority value of an event determines the insertion position of that event in the queues and consequently the processing order by the character. Events with the priority value CRITICAL have the ability to cease the processing of whatever event is currently being processed by the perception module to give relevance to the critical event. For example an explosion would be an event with a critical priority value, as it has to be reacted to immediately. Conversely, noticing that someone in the room is bored, unless relevant due to a particular relationship, would be an event with a LOW priority value.

5.2 The events types

All events have time-out value and priority value as described in the previous sections. There are seven different event types currently implemented in the system and listed below. They are implemented in an inheritance hierarchy, each deriving from an abstract base class “Event”, pointers to which are used throughout the system to facilitate polymorphism (examples found later in this section). The purposes of these events are as follows: *Null* - generated when nothing is happening externally to the character (there are no events to be processed). This is a tick of the time passing. *Gesture* - when a character performs a gesture this event informs the other characters in the same room. *Movement* - when a character begins to move to a certain location, in the current or a nearby room, this event is generated. *Mood* - used by rooms to affect the emotional status of characters residing inside it. *Delay* - allows a rest time before processing any other events. *Enter* - generated by a room when it detects that a character is entering the room, to inform all other characters in the room. *Leave* - Generated by a room when it detects that a character is leaving the room, to inform all other characters in the room.

All the events have the same field by defaults as they are inherited. Those are: event ID and animation names, timeout value, various booleans to affecting processing, values for enabling synchronicity of events, emotional affecters, priority value and destination type. Not all the fields are relevant to all event types, and the data content of each field differentiates an event from another, as it will be described in the following section.

5.3 Event processing

An event’s purpose is not just informing other characters that something is occurring, it is also used as a way to make the character firing an event do some processing. These events are termed *outgoing* events while they are *incoming* events when they are retrieved from the global event queues. The flow of the event processing system can be seen in Figure 5.2.

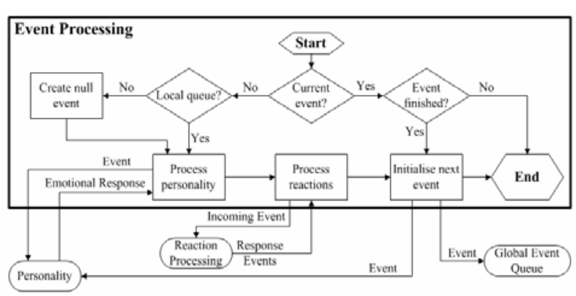


Figure 5.2 – Event processing procedure

Referring to Figure 5.2 if an event is currently being processed then it is checked to see if it is finished. The process() method is invoked and the boolean result examined. If the event has not finished then nothing more is performed in the event processing and the perception module has finished updating. Alternatively, if the event has finished the next event in the response list (if there is one) should be initialized according to its init(). Initialization and termination of events are explained in section 5.3.1. In addition, depending upon the boolean value of event fields named “global” and “personality”, the event is sent to the global event queue and personality model respectively.

In case there is no current event being processed by the perception module, the alternate path from the initial decision node is followed. A check is made to see whether there are any events within the local queue of the character and if there are any the first one is passed to the personality module for processing. If there are no events in the local queue then a *NullEvent* is generated and processed. Null events are explained in section 5.3.2 . Once an emotionally-coded response is received by the personality system, the event and the *emotional response (named incoming event* in Figure 3.1) are processed by the *Reaction Processing* sub-module that retrieve the scripted events/reactions and the first one returned is initialized.

Other important areas of the way the event system works require further explanations and are reported below.

5.31 Event initialization and termination - An event has polymorphic methods of initialization and termination checking. The standard behavior implemented in the abstract base class of the hierarchy: “Event” is an “init()” method with an empty body and a boolean “process()” method that always returns true. Most of the event classes need no extension over this default behavior however exceptions are the event types Delay, Gesture and Movement. Their “init()” and “process()” methods are depicted in Figure 5.3.

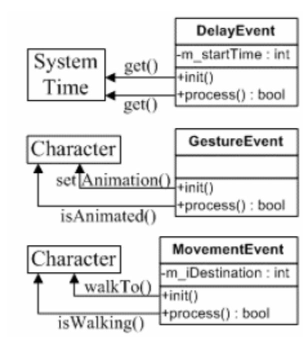


Figure 5.3 – Initialization and termination for delay, gesture and movement events

These methods are invoked for outgoing events only i.e. the response events from a reaction.

5.3.2 NullEvent – A null event (*NullEvent*) is generated when nothing is happening externally to the character. Its purpose is to provide scope for the programmer to script a reaction to nothing, that is say what the character should do when nothing else is going on. A good example of this would be a test for boredom; if the character is not being instructed to do anything, a null event is generated by the character itself and processed by its personality so that an emotional response can be obtained. If all the OCC category values lie within a certain distance from the midpoints then it could be deemed that the character is bored and trigger a boredom response e.g. walk to a random through the rooms until you meet someone else. Null events should never be sent externally from the perception module.

5.5.4 Synchronous events – A synchronous event is one that requires another character to perform the same or a different event that needs to happen at the same time in the scene. The only synchronous events implemented in our current version of the system are some gesture events. For example, if two characters were to shake hands see Figure 5.4, the animations of each should happen at precisely the same time to look believable. The synchronous gesture events use the event type's polymorphic initialization and "process()" methods. Unlike a normal gesture event, a synchronous gesture event does nothing inside its initialization method; instead, it defers setting of the gesture animation in the character until the process method is called. However, this only occurs if the target of the gesture event also has a synchronous event currently being processed. At every update a check is made against the target, as soon as the target indicates that it is processing a synchronous event, the animation is set. Two fields are used to indicated that an event is synchronous: the *sync* field that is a Boolean indicating whether or not the event is synchronous, and the *syncTimeout* field determines how long the event processing must wait before terminating the event. If synchronous events are performed, they terminate in the same way as normal gesture events.

5.4 Reaction Processing and scripting

Breaking down a human response to a certain occurrence into its constituent actions it makes it clear how such response can be described as a of pre-specified events. For example if a friend appears one might wave and to move near to talk to him/her. Thus a character appearance is a valid input event generated by the room it enters. In this case a plausible response sequence for a friendly character in the room is to wave, walk towards the new comer, shake hands and smile to such character. Conversely, an unfriendly character in the room other might walk towards the new comer, shake hands at fist, but then start arguing.

A character reaction-processing sub-module allows a series of events to be performed in turn, as a consequence of the emotional response input (see Figure 3.1). Each event of the sequence starts only when the previous one has finished. A *reaction* for the character is the incoming-event in input that requires a response, plus a list of conditional event list responses. Conditions for responses are in the form of five ranges, one for each of the OCC paired categories, and it can hold values in the continuous range from -1 to +1, indicating the normal rage of the emotion, or the extreme values -2 and +2 indicating an extreme emotion.

The incoming-event/reaction relationship used in the system are defined in scripted text, external from the source code of the actual program. This caters for customizable behavior and enhances the user's ability to provide a flexible interaction system within the simulation. In the script the incoming-events are defined as an angle-bracketed event type header, such as <GestureEvent>, followed by a list of fields and their values. The reactions are similarly defined using a <Reaction> header and then at least one pairing of a set of OCC value conditions and a list of response event IDs (corresponding to those in the scripted event file). This pairing can be repeated for as many different conditions and their associated responses as deemed appropriate.

6. Visualizing emotion

Irrespective of the quality of emotional modeling involved, it is of limited use unless this can be portrayed to the user in a believable manner. This does not necessarily mean photo-realistically, as it is clear that character empathy can be present in highly unrealistic, but believable context – Disney/Pixar films. In this section, are discussed the steps taken to empathetically portray the emotional state of the characters (or at least those that can be observed from visuals alone) within the social simulation.

Visual expression can be classified into three groups: facial expressions, postures, and gestures. The first two represent the pure emotion of the subject and are discrete, such as can be obtained from a photograph. The actual personality of the person cannot be determined in such a forthright manner, but it is portrayed through the actions and gestures performed by the character over time. Such actions/gestures are recalled from pre-stored animations rather than generated through a procedural language like in [5] [6].

6.1 Facial expressions

Due to the cartoon-like nature of our scenario, chosen in the wish to avoid the uncanny valley, facial expressions are not generated by a deformable 3D mesh, but are rendered in 2D to video memory before being used as a texture for the 3D head model. Not only does this make the rendering of the face considerably easier, it also creates a layer of abstraction so that the body model can be changed (to a human for example) without having to reform the facial mesh or rendering routine.

The definition of the face is based on Minimal Perceptible Actions [21]. Due to the 2D nature of the implementation the number of variables can be considerably reduced without losing the range of achievable expressions. We use only the following 11 facial expression factors: eyes open/closed, eyebrows raised/lowered, eyelid/eyebrow tilt, pupils horizontal, pupils vertical, mouth width, mouth smile/frown, mouth open/closed, where the last three have been defined separately for the left and right side of the face.

Prior to the model of each chimpanzee being rendered, a stack of facial features are positioned through reference to these values as shown in Fig. 6.1. Each of the 10 OCC variables has its own Expression Factor Set (EFS) representing the expression for that emotion. In addition to these there is also an eleventh EFS representing the neutral expression.

Figure 6.1 – The face stack.

Each emotional scale is, as indicated earlier, within the range -1 to 1, plus the extreme values –2 and 2 representing extreme static emotion. Beyond a magnitude of one, the values have no affect on the facial visual emotion that are therefore clamped, but when an extreme level is reached (value –2 or 2) the character performs an animation to further express the feelings. The EFS blending is done through a weighted average variation from the neutral expression (E = Expression).

$$E = \frac{E_n(1 - W_{\max}) + \sum E_x W_x}{(1 - W_{\max}) + \sum W_x}$$

Equation 6.1 – Expression function

Where E_n is the neutral EFS, E_x is the x^{th} emotional EFS, W_x is the x^{th} OCC value, and W_{max} is the maximum weighted OCC value.

6.2 Postures & gestures

The body of the chimpanzee is rigged with an Forward Kinematics (FK) skeleton for animation. Each factor of the OCC has its own animation frame defining the body posture for the emotion. At run-time, these are combined using a weighted average to produce a posture (A =animation) relative to the current emotion.

$$A = \frac{p \sum A_x W_x}{\sum W_x}$$

Equation 6.2 – Animation function

Where A_n is the x^{th} posture animation, W_x is the x^{th} OCC value, and p is the overall weighting of the total posture. The posture strength (p) remains constant unless the character is allocated a gesture to perform in which case it is reduced to allow blending with the gestural animation. Whilst this produces some hint of the emotion being experienced (besides the actual choice of gesture) the definition is small. This results in unrealistic movement, as real gestures are very heavily affected by the current mood of the instigator.

To better represent the emotional state of a character we introduce the concepts of a Gestural Affection Table, see Table 6.1. This identifies that depending on your current emotional mood, the speed and magnitude of your movements are altered. For example, an unhappy person makes small slow movements, while an angry person will be much more expressive in both magnitude and speed. These values are again averaged across the current OCC values for the character, and the animation altered by the resulting amount.

Emotion	Magnitude	Speed
Joy	+100%	+10%
Distress	-80%	-50%
Hope	-	-
Fear	-80%	+25%
Relief	+60%	-45%
Disappointment	-50%	-45%
Pride	+60%	-20%
Anger	+50%	+40%
Love	-	-
Hate	-40%	+35%

Table 6.1 – Gesture affection table

The table values have been generated through usability tests with four users, speeding up or slowing down and changing the amount a character moved during an animation until a level that was believed to be realistic for the emotion was being shown. Assumptions based on our everyday experience guided the testing, were for example anger produces a more expressive (thus higher speed and magnitude) display of energetic expressions than another emotions such as distress and disappointment. Furthermore the animations have been tested as described in section 7.

7. Scenario adaptation and impulse decision making

The prototype system created is described in this section. A scenario was defined along with a set of events and reactions to test the personality model. The code itself it is flexible and can be tailored towards any number of real-world uses. Characters in the world are born as a 'blank page' representing an average personality, with no relationships memories or personality tendencies. The characters' personality can be easily be set with sliders on the control pads for mood, FFM, SCF, current emotion and relationship.

The world can be populated with an unlimited number of characters. The currently scenario created contains a neutral foyer with three attached rooms designed to instigate emotional change to those that reside in them: a disco (inducing joy), an haunted room (inducing fear), and a rather dismal pub where the bar is closed (inducing anger).

The adaptation system of the character is such that if a character is staying for a prolonged time in one of the rooms that induce a specific emotional aura having its mood modified, it reaches ad extreme emotional state performs an extreme gesture, to illustrate its status, and decides itself (with an impulse based decision) to return to the neutral foyer. At the return to the foyer the character's personalities it is altered accordingly to its past experiences.

If three or more chimpanzees are created within the central foyer, they automatically interact with each other using the events/responses scripted. In our prototype for example they wave, move closer to each other, shake hands, and then wander off into one of the three rooms around the foyer. Upon re-entrance to the foyer the characters are at a heightened emotional state, they do not quite act as cordially as at the start, while anyone else in the room, being in a normal state, waves at them. This wave induces a response according to the emotional state of the character. For example, if the character has been in the angry room it may return to the foyer and start slapping other characters for extreme anger, if it was in the disco it dances and gives the other characters a hug. Coming back from the haunted room a chimpanzees will cower in fear in response to a wave.

This short series of actions and reactions shows that the model drives the character to vary in response to their environmental situation. Depending upon what situations they have found themselves in, their behavior changes as a result. The way that their behavior changes, is consistent between different rooms, and changing the make up of their FFM and SCF characteristics further changes the manner in which their behavior varies.

8. Perception testing

To evaluate the quality of the graphical effort in portraying the emotional modeling, two tests have been performed, named static and dynamic emotion conveyance, to determine how successful the characters are at conveying empathy, thus social presence, in the viewers. The facial emotions, postures and gestures have been initially modeled based on both the six basic human facial expressions of emotions and the cartoon expressions, posture and gestures observed watching various Disney/Pixar type of characters. The expressions have been experimented with until all authors agreed on the emotion they conveyed and ready to be tested by a bigger audience. Description and results of such tests are presented below.

8.1 Static emotion conveyance

A first test evaluates static expressions and postures. Thirty volunteers where asked to participate to the test, they were all undergraduates in the computer science department, of mixed gender.

For this test the ten emotional extremes of the system (anger, pride, love, hate, joy, distress, fear, hope, relief, and disappointment) along with four mixed emotions (hope/pride, disappointment/distress, joy/relief, and fear/anger) and the neutral expression as a control, have been used. The aim was to evaluate how well the static emotions are conveyed by the character first by the facials alone, and then the impact of the inclusion of body posture on the emotion identification task.

The expressions were presented along with the five emotional pairs with a five-point scale (extreme negative emotion, moderate negative emotion, neutral emotion, moderate positive emotion, extreme positive emotion) between each of them. The user was asked to identify which of the emotions that the expression was trying to convey. The user records this by marking on each line the appropriate emotion that they think is displayed in the image of the character's face.

In the second phase of the test the volunteers were shown once more the same facial expressions (in a different order), but this time the still picture incorporated the body posture into characters emotional portrayal. The participants were again given the five 'sliders' and asked to rate the perceived emotion conveyed by each of the pictures.

In analyzing the results, an error distance measure has been used to see how far the users' perception of the emotion was, from the emotion the character was trying to convey was considered. The absolute amount that the user differed from the correct result for each of the emotions was collected. Then for each facial expression, an average of distance that the users misclassified the emotion was calculated. These were then collated and are presented in graphical form in Figure 8.1.

8.2 Dynamic emotion conveyance

A second test was performed to determine the effect of animating the different postures and expressions. The aim was to see if dynamic changes in the appearance of the character make the identification of the emotion easier. Twenty participants, once more undergraduate in the computer science department, watched the animations of one of the character in isolation on a 17" screen and were asked to identify the emotion pair being displayed from a set list. The experimenter drove the character's animation.

At first the character was moved from a neutral state to both extremes of an emotional pair, for three of the five emotional pairs, where when an extreme level is reached the character performs an animation to further express their feelings.

Second the volunteers were asked to identify which of the ten extreme emotions was being portrayed. Ten test events were created. Each of these represented one of the ten extremes of emotion, of which four were presented to each user. The results were evaluated on a purely hit or miss based method. If the user identified the correct emotion then a hit was recorded, a miss otherwise.

8.3 Discussion

8.3.1 Static emotion conveyance - The distance error shows that that overall, the inclusion of body posture makes the classification of emotions a great deal easier than just a facial expression on its own on a still pose. In most circumstances, the inclusion of body posture reduced the distance error found between the emotion that the system was presenting and the emotion that was perceived by the participants. The results are shown below in Figure 8.1.

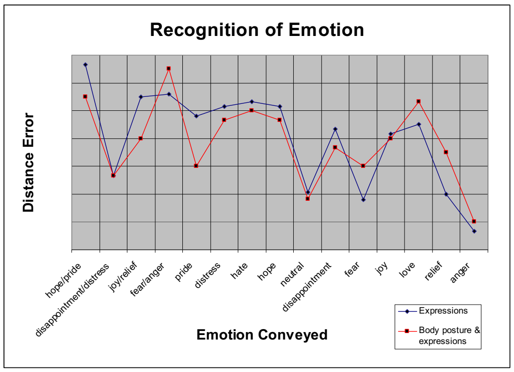


Figure 8.1 – Increase in recognition of emotion when postures are used.

The most surprising results are those that present a larger distance error, such as those of fear/anger, fear, love, and relief. What this suggests is that the user has become confused by the increased amount of information in the picture that obviously does not correspond to their personal beliefs as to how the emotion should be conveyed in an expression and body posture. However, in other cases a significant improvement in recognition has occurred.

In [Coulson04] a survey of static body postures was performed where an identification of the most appropriate static body posture that represents a particular emotion is found. If we take one of our strongest postures, fear, and compare it the posture provided in the paper (Figure 6.3) we can identify that several of the crucial elements are apparent – the outreached arms for example, although not as horizontal are significantly different to the neutral posture in our model. Although, in our model the posture was never meant to convey the full meaning of the emotion, simply a guide as to what the emotion should be. [28] results echo a similar point – the body posture can be very difficult to interpret as an emotional representation. For some emotions, with specific people, there would be no perceptible difference in posture; it can all be conveyed by the face. In contrast, there are some emotions, and some people, who are very expressive with their body movements, so will convey much more information. We feel that the key is to be consistent. If a consistent emotion is portrayed in a certain way then it will be understood given the context of the expression, the actions that resulted in that emotion, and the previous manner in which the character has acted.

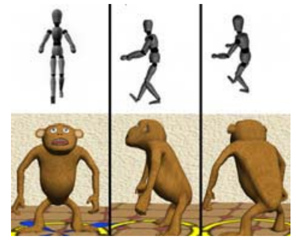


Figure 8.2 – Fear posture.

8.3.2 Dynamic emotion conveyance - In the second round of evaluation the dynamic emotions provided some interesting results. The overall correct classification of emotions improved significantly. If we examine Figure 6.4 we can see that many of the emotions are identified correctly in over 70% of cases.

The major problems occur in the identification of distress and hope. This is due to the vague nature of hope and the misclassification of distress, often with fear, although fear itself is very often identified correctly. One thing to take into account is that our character will produce an action when their emotional levels reach a certain point. This action then becomes the dominant feature of the emotion. For example, in relief, the character wipes his hand across his brow. This is a well known and obvious mannerism for someone who is relieved. Yet, if we examine some of our weaker results such as hope. The action when hope is to be displayed is very vague and perhaps does not convey the emotion of hope across very well.

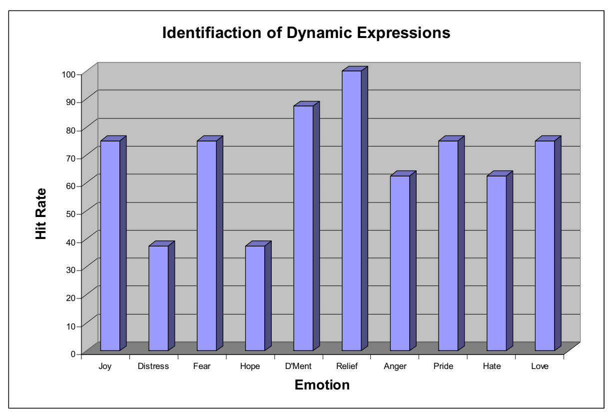


Figure 8.3 – The identification rate of dynamic expressions of emotion.

If we further consider the problem of misclassification of results we can conclude that certain emotions are classified incorrectly more often than could be expected due to simple error. The misclassification has been described pictorially in Figure 6.4. Here it can be seen that the emotions anger and hate are placed close to one another, this shows that anger and hate are often mistaken for each other. Similarly, emotions such as joy, love, and hope are often misclassified as each other by the user. In contrast, emotions such as disappointment and hate are rarely mistaken for each other in our experiments are a distance apart.

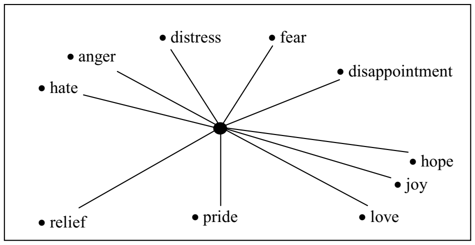


Figure 8.4 – Common emotional misclassification

In conclusion, our results show that although the actual classification of emotion into their specific type is quite poor, the users do recognize an emotion in a similar way, or at least an element of the emotion is recognized. The most common mistake of a user is to try and identify too many types of emotion in the one being portrayed.

9. Conclusion and applications

A model is capable of expressing the personality's current emotional level through facial expression, body posture, and by performing gestures has been created and embodies into emotionally believable chimpanzees. The emotional levels of the character are displayed by taking the novel approach of mapping the OCC emotional values to a facial texture generator, and reinforcing them with posture and gestures. The model's final expression of emotion has been validated using a series of user tests and we are satisfied that our simulation is successful in conveying emotions in a way that can be understood in a meaningful manner.

The personality model takes the approach of combining the consistent personality traits represented by the five factor model [9] and the environmental and behavioral regulators that are the social cognitive factors [Cervonne99] using an emotional model represented using a reduced version of the OCC model of emotion [8]. The OCC model is used to represent short-term emotion, long term mood, and a relationship based memory. This provides a means of representing time-dependent personality adaptation.

The system is event driven to provide situation awareness and the characters use their personality model to adapt to the environment and decide their actions following a scripted variability in response to given events depending upon their emotional levels and personality. The scripting system makes the characters respond to events that are not constrained to one particular context, making our system potentially applicable in many areas such as training, behavioral pattern analysis, and computer games.

References

[1] Biocca, F. The Cyborg's Dilemma: Progressive Embodiment in Virtual Environments. Journal of Computer Mediated Communication, 3/2. 1997

[2] Slater, M., & Usoh, M.. Representations systems, perceptual position, and presence in immersive virtual environments. Presence, 2/3, 221-233. 1993.

[3] Tammelin, M. From Telepresence to Social Presence: The Role of Presence in Network-Based Learning Environment. In Tella,S.(ed.) Aspects of Media Education: Strategic Imperatives in the Information Age. Media Education Centre. Department of Teacher Education. University of Helsinki. Media Education Publications 8. 1998

[4] Rourke, L., Anderson, T., Garrison, D.R., & Archer, W.. Methodological issues in the content analysis of computer conference transcripts. International Journal of Artificial Intelligence and Education, 12. 2001

[5] Loyall, A.B, Bates, J. Personality-rich believable agents that use language. International Conference on Autonomous Agents, *Proceedings of the first international conference on Autonomous agents*, Marina del Rey, California, United States, 106-113. 1997.

[6] Rickel, J., Marsella, S., Gratch, J., Hill, R., Traum, D., Swartout, W. Towards a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, 2002.

[7] G. Papagiannakis, H. Kim, N. Magnenat-Thalmann. Believability and Presence in Mobile Mixed Reality Environments, *IEEE VR2005 Workshop on Virtuality Structures*. 2005

[8] Ortony, A., Clore, G., & Collins, A. The Cognitive Structure of Emotions. Cambridge University Press. 1988

[9] Digman, J. M. Personality structure: Emergence of the five-factor model. Annual Review of Psychology, 41, 417-440. 1990

[10] Kshirsagar, S. Virtual Humans Personified, *Proceedings Autonomous Agents Conference (AAMAS)*, 356-359. 2002

[11] Dugdale, J., El Jed, M., Pallamin, N., Pavard, B. Modelling Character Emotion in an Interactive Virtual Environment. *AISB Symposioum, 29 March, 1 April,Leeds, UK..* 2004

[12] Romano, D.M., Wong, A.K.L. Personality Model of a Social Character. *Empathy in Human Computer Interaction Workshop, HCI 2004, 6 Sept., Leeds, UK*. 2004

[13] Ortony, A. On making believable emotional agents believable. Trapple, R. P. ed.: Emotions in humans and artefacts. MIT Press, Cambridge, USA. 2003

[14] Louchart S., Romano D.M., Aylett R., Pickering J. Speaking and Acting - Interacting Language and Action for an Expressive Character. *Intelligent Virtual Environments and Virtual Agents Workshop*. Mexico City, Mexico. 2004

[15] Cervone, D, Social-Cognitive Theories and the Coherence of Personality. In D. Cervone & Y. Shoda (Eds.), The coherence of personality: Socialcognitive bases of consistency, variability, and organization, 3-37. 1999

[16] Rudomín, I., Perez, F., Millan, E. Groups and Crowds with behaviors specified in the environment. *Intelligent Virtual Environments and Virtual Agents Workshop*. Mexico City, Mexico. 2004

[17] Delgado-Mata, C. Characterising Emotion Signalling and Affected Group Behaviour in Virtual Mammals. *Intelligent Virtual Environments and Virtual Agents, IVEVA 2004, ITESM CCM, Mexico, CEUR Workshop*, 97. 2004

[18] Mori, M.. The Buddha in the Robot. Charles E. Tuttle Co. ISBN 4333010020. 1982

[19] Paiva A., Dias J., Sobral D., Aylett, R., Zoll C., Woods, S. Caring for Agents and Agents that Care: Building Empathic relations with Synthetic Agents. *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 194-201. 2004

[20] Parke, F.I.. A Parameterized model for facial animation, IEEE Computer graphics and applications, 2/9, 61-68. 1982

[21] Kalra P, Mangili A, Magnenat Thalmann N, Thalmann D. 3D Interactive Free Form Deformations for Facial Expressions, *Compugraphics, Siesembra, Portugal*. 1991

[22] Glanz, K., Rimer, B.K. & Lewis, F.M. Health Behavior and Health Education. Theory, Research and Practice. San Fransisco: Wiley & Sons. 2002

[23] Blumber, B. Learning in character: building autonomous animated characters that learn what they ought to learn. In O. Balet, G. Subsol, and P. Torguet (Eds.): ICVS 2001, LNCS 2197, 113–126. 2001.

[24] Blumberg, B., Downie, M., Ivanov Y.A., Berlin, M., Johnson, M., P., Tomlinson, B. Integrated learning for interactive synthetic characters. *SIGGRAPH 2002:* 417-426

[25] Ewen, R. B. Personality: A topical approach. Mahweh, NJ: Erlbaum. 1998.

[26] Pervin, L. A. Personality: Theory and research (5th ed.). New York: Wiley. 1989.

[27] Popkins, N. C. The Five-Factor Model: Emergence of a Taxonomic Model for Personality Psychology, http://www.personalityresearch.org/papers/popkins.html. 1998.

[28] Coulson, M. Attributing Emotion to Static Body Postures: Recognition Accuracy, Confusions, and Viewpoint Dependence. Journal of Non-Verbal Behavior, Kluwer, 28/2. 2004.