# Creating Socially Interactive Non Player Characters: The μ-SIC System

Brian Mac Namee & Pádraig Cunningham
Machine Learning Group
Dept. Computer Science
University Of Dublin
Trinity College
Dublin 2
Ireland
E-mail: Brian.MacNamee@cs.tcd.ie

## ABSTRACT

A number of recent, highly successful games have shown that there is a demand for the personalities, moods, and relationships of Non Player Characters' (NPCs) to be made the focus of game-play. In order for this shift of focus to take place, agent architectures used to create NPCs must be augmented with models of these aspects of characters' personae. These models must then be used to drive characters' behaviour.

This paper will present the μ-SIC system, a component of an intelligent agent architecture designed for the creation of NPCs for computer games. μ-SIC uses a number of quantitative psychological models to simulate characters' personalities, moods and relationships. The values of these models are used as inputs to an Artificial Neural Network (ANN) which drives characters' social behaviour.

## INTRODUCTION

The success of games such as *The Sims* (thesims.ea.com) and *Black & White* (www.bwgame.com) have shown that there is a demand for the personalities, moods, and relationships of Non Player Characters' (NPCs) to be brought to the fore as the focus of game-play. In order to satisfy this demand, intelligent agent architectures must be augmented with relatively sophisticated models of these extra aspects of characters' personae. These models should then be used to drive characters' behaviour.

Research in psychology offers a number of solutions to model the aspects of characters' personae in which we are concerned. In order to use these models to drive character behaviour, connectionist AI techniques, and in particular Artificial Neural Networks (ANNs), can be used. An ANN can be trained to take a particular set of values for the aforementioned psychological models, and from these, determine which interaction a character should perform at a particular moment. The role the ANN plays in the μ-SIC system is to generalise smoothly from a set of handcrafted examples to cover the whole space of possibilities. This paper will describe the μ-SIC system which achieves these goals.

The purpose of the μ-SIC system is to choose which social interactions NPCs should engage in when placed within a virtual environment with other characters (both players and other NPCs). When a moment within a simulation arises in which a character is free to engage in an interaction, the μ-SIC system is queried with the character's personality and mood details, and also their relationship details to each of the other characters in the same location, who are available for interaction. From these queries a particular interaction with a particular character is chosen.

This paper will begin with an overview of a larger project of which the μ-SIC system is just one component. Following this, a description of the psychological models used by μ-SIC will be given. The actual implementation details of the system will be described next, along with a discussion of the troublesome data acquisition problem. Next, a short description of a simulation example which demonstrates the μ-SIC system will be described, along with an evaluation of the system's performance. Finally, some ideas on how the system might improved will be given.

## PROJECT OVERVIEW

Although games are becoming ever more engaging, it is still normal in current adventure and role-playing games for the behaviour of NPCs to be very simplistic. Usually, no simulation of an NPC is performed until the player reaches the location at which that NPC is based. When the player arrives at this location, NPCs typically wait to be involved in some interaction, or play through a pre-defined script. This can lead to very predictable, often jarring behaviour, and also frequently causes inconsistencies to arise. In order to overcome these limitations new models are required for implementing game characters.

Although such models have not been widely used in computer games, a number of architectures for creating realistic characters have been developed. For example, work led by Thalmann (Caicedo & Thalmann, 2000) and the Oz project (Mateas, 1997), which is based on interactive drama, have both developed virtual human architectures. As part of the TCD Game AI Project (Fairclough et al, 2001) the Proactive Persistent Agent (PPA) architecture (Mac Namee & Cunningham, 2001) is being developed for the express purpose of creating game NPCs, and overcoming the limitations typically associated with current computer game characters.

Agents based on the PPA architecture are *proactive* in the sense that they can take the initiative and follow their own goals, irrespective of the actions of the player. *Persistence* refers to the fact that at all times, all NPCs in a virtual world are modelled (at least to some extent), regardless of their location relative to that of the player. Although these two properties are considered an inherent part of the intelligent agent paradigm (Wooldridge & Jennings, 1995) they have mostly been ignored in agent architectures used in computer games (Mac Namee & Cunningham, 2001).

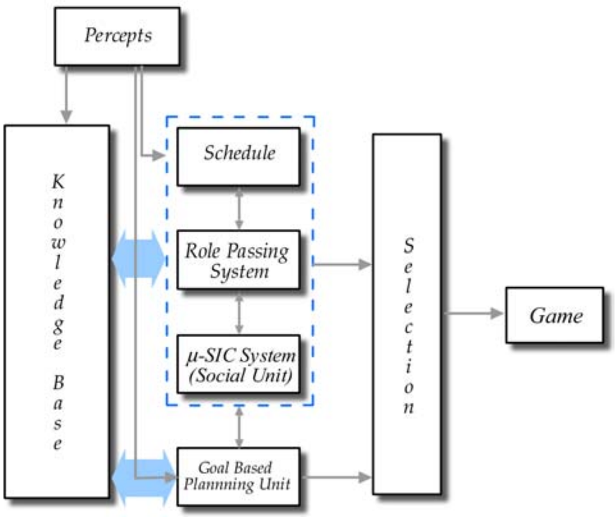


Figure 1: A schematic of the PPA architecture. This paper focuses on the μ-SIC system which implements agents' social capabilities

A schematic illustration of the PPA architecture is shown in figure 1. The architecture is a modular system made up of four key components. These are as follows:

**Schedule:** As NPCs created using the PPA architecture are expected to be persistent, they should appear to carry on ordinary lives as a game progresses. For example, an NPC might get up in the morning at home, go to work, go to the cinema after work and finally going for a drink in a local bar to wrap up the day. By instilling characters with a schedule this can be achieved without any intense computation or sophisticated data structures.

**Role-Passing System:** As mentioned previously NPCs in computer games are required to appear in a range of different situations, and in each one behave in very different ways. In the PPA architecture this is achieved using *role-passing* (Mac Namee et al, 2002). Role-passing begins with a basic agent capable of simple behaviours and driven by a small number of core motivations. Depending on the situation in which the agent is found, different roles are layered on top of this simple agent to drive its behaviour. The adoption of particular roles is driven by the agent's schedule.

To illustrate this technique, consider the NPC mentioned previously. This character would begin the simulation by adopting the AT-HOME role, adopt the OFFICE-WORKER role while at work, the CINEMA-PATRON role while attending the film and finally adopt the BAR-PATRON role while at the bar. The adoption of each of these roles would significantly change the behaviour of the character.

**Social Unit:** The social unit is used to control characters' social interactions and maintain inter-personal relationships between NPCs. The social unit is implemented using the μ-SIC system and is the subject of this paper.

**Goal Based Planning Unit:** NPCs should be capable of pursuing long term goals within a game. The role passing technique cannot achieve this alone and so a traditional planning unit is used.

## USING PSYCHOLOGICAL MODELS TO SIMULATE NPCS' PERSONAE

This section will describe the quantitative models taken from psychology which are used to model NPCs' personalities, moods and relationships. However, before discussing the models used, it is worth taking a moment to discuss the criteria used for selecting suitable models.

The first selection criterion is that the models chosen need not necessarily represent the current state of the art in cognitive science. Our goal is to create characters which behave plausibly at all times within a simulation, so models which achieve this are enough.

The second important criterion for model selection is that the models used should be as simple as possible. Ultimately, it is intended that the PPA architecture will be used by game designers to add NPCs to their games. If this is to be achieved, models must be simple enough that the designer can understand how they work, and more importantly, how changing a model's parameters might affect a character's behaviour.

In addition to the concern for usability, any system for use in games must be efficient both in terms of memory usage and the computation required.

### Personality Model

The first important factor of an NPC's persona which needs to be modelled is personality. This ought to allow the creation of characters with personality types, such as *aggressive*, *sociable*, *moody* etc. From the whole myriad of psychological models of personality available, Eysenck's two dimensional classification (Eysenck & Rachmann, 1965) has been chosen.

The Eysenck model plots personality across two orthogonal axes, *introversion-extroversion* and *neuroticism-stability*. From (Lloyd et al, 1984), the extrovert is said to be sociable, impulsive and open to new experiences, while the introvert is quiet, serious and prefers solitary experiences. The neurotic is contrasted with a stable person by suffering from tension and interpersonal difficulties. An illustration of the model, which shows the positions of a number of the possible personality types, is shown in figure 2.

It is worth noting that psychologists generally accept that two axes are not enough to accurately model the whole gamut of human personality types. Currently the most sophisticated models, such as the OCEAN (McCrae &

Costa, 1996) personality model, operate across five axes. However, the use of more axes was deemed overly complex for the purposes of game simulation, and the Eysenck model was chosen as it remains one of the most respected and well established personality models in psychological theory (Lloyd et al, 1984).

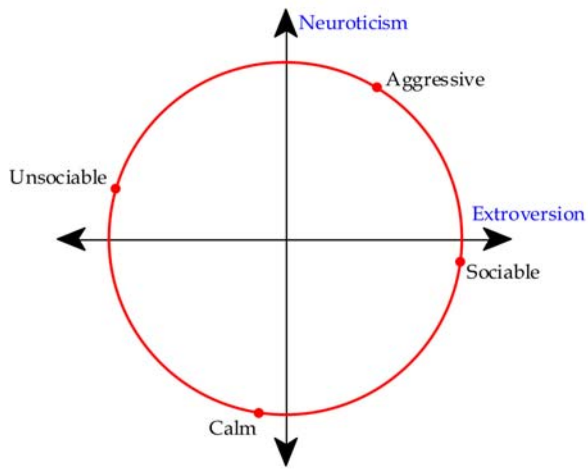


Figure 2: The Eysenck personality model which measures personality across the Introvert-Extrovert and Neuroticism-Stability axes. The positions of a number of personality types are shown

**Mood Model**

The second psychological model used, simulates a character's mood as it changes over time through interactions with other characters or players. Again, simplicity is key and a model (shown in figure 3) which works across two axes has been chosen. An agent's mood is measured according to *valance* and *arousal*, where valance refers to whether the mood is positive or negative, and arousal refers to the intensity of the mood.

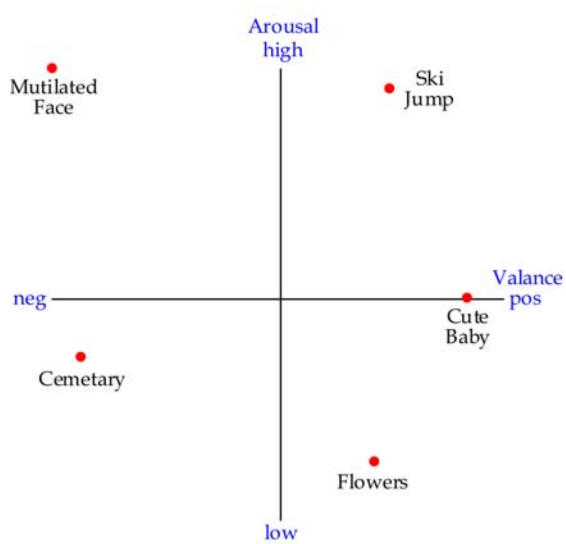


Figure 3: The Lang mood model which plots mood according to valance and arousal. Subjects reactions to different pictures are shown on the model

This model has been used in computing applications before (Picard, 1995), and is originally due to Lang (Lang, 1995). Over the course of Lang's work, this model was used in experiments wherein subjects were shown a number of pictures with their reactions to these pictures plotted according to the two axes. Some of these reactions are shown in figure 3.

**Relationship Model**

The third model used (shown in figure 4) simulates agents' relationships with each other and players. The model has been used in a number of other entertainment projects, namely the Oz Project (Scott Neal Reilly, 1996), TALE SPIN (Meehan, 1976), and UNIVERSE (Lebowitz, 1985), and has its psychological basis in (Wish et al, 1976). Traditionally, four values are used to characterise the relationship of one character to another. These are the amount that a particular character likes another character, how physically attracted one character is to another, whether the characters are dominant or submissive towards each other and how intimate the characters are.

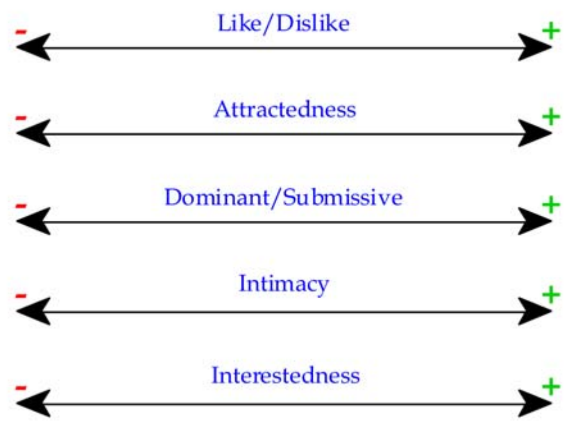


Figure 4: The relationship model used which plots a character's relationship to another character

To facilitate conversation, this model has been augmented with a value indicating how interested one character is in another. Conversation within the µ-SIC system is based on a very simple model in which each character has a list of subjects in which they are interested. When characters engage in a conversation they simply pass these subjects back and forth. Thus, characters are interested in one another if they share a number of common subjects of interest.

**IMPLEMENTING THE µ-SIC SYSTEM**

In order to use the psychological models just described to drive social behaviour, a technique is required which can take the current values of these models, and determine whether an interaction should be started, and if so which one. An ANN has been chosen to perform this task.

ANNs (Russell & Norvig, 1995) are a class of machine learning technique which is based on the manner in which neurons in biological brains operate. ANNs can be used to perform classification tasks in which a set of inputs

describing a particular problem case are presented to the network, which then outputs its class. In the µ-SIC system the inputs to the ANN are the current values of the models just described, while the outputs indicate which interaction should take place.

The structure of the ANN used by the µ-SIC system is shown in figure 5. The network used is a multi-layer perceptron (MLP) network, which uses just a single hidden layer. The network's input layer has nodes for the personality and mood of the character who is attempting to instigate an interaction, and their relationship to the current character being considered for interaction. The output layer has nodes for each of the possible interactions which the characters can engage in.

Before an ANN can be used to perform classification, it must be trained to recognise the different classes involved. Training a network involves presenting a number of known examples of the problem case to the network and adjusting the network's internals based on how well the network can recognise these training examples. In order to train the network used in the µ-SIC system the back-propagation of error (or more succinctly BackProp) algorithm (Bishop, 1995) was used.

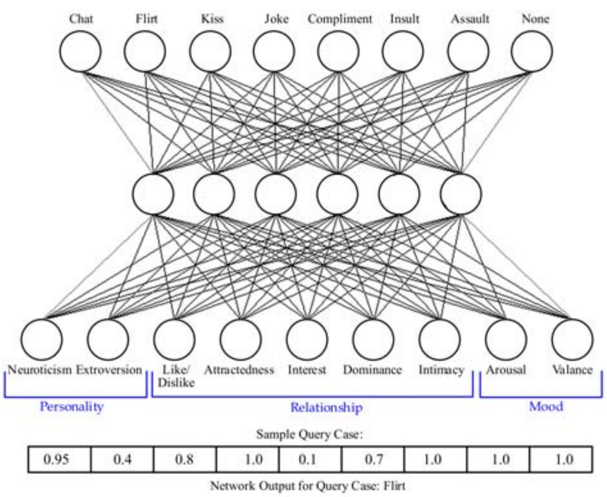


Figure 5: The structure of the ANN used within the µ-SIC system

For training, a data set describing the problem space must be acquired. Data acquisition is often a difficult problem in machine learning tasks, and is particularly so for the µ-SIC system, as there are no databases available which contain information on how people interact. Because of this it was required that an artificial data set was created. In order to do this a number of simulation situations were created and populated with characters whose personalities were set using the Eysenck model. Relationships between these characters were initialised and a group of people determined which interactions these characters would engage in as their moods changed over time. This resulted in a data set consisting of approximately 300 hand crafted data elements.

If an ANN were trained using just 300 data elements it is very likely that the network would *over-fit* to these examples. Over-fitting (Bishop, 1995) is a problem suffered by classifiers in which they too-closely learn the patterns contained in a small data set. If a classifier over-fits to a training set, the classifier will not generalise well to unseen query cases. So that this would not occur, the original data set was transformed by adding Gaussian noise to each of the samples to perturb the hand crafted elements and thus create a set of new data elements. This resulted in a new set of approximately 2000 training elements.

To determine the accuracy of the network a five-fold cross validation[1] was performed using the perturbed set of 2000 data elements. The system achieved an accuracy of 85% in this test indicating that the output of the network was both consistent and coherent. In addition to the cross validation test, the network was also tested on the original hand crafted data set, which was not used during training. The network achieved an accuracy of 84% on this data set.

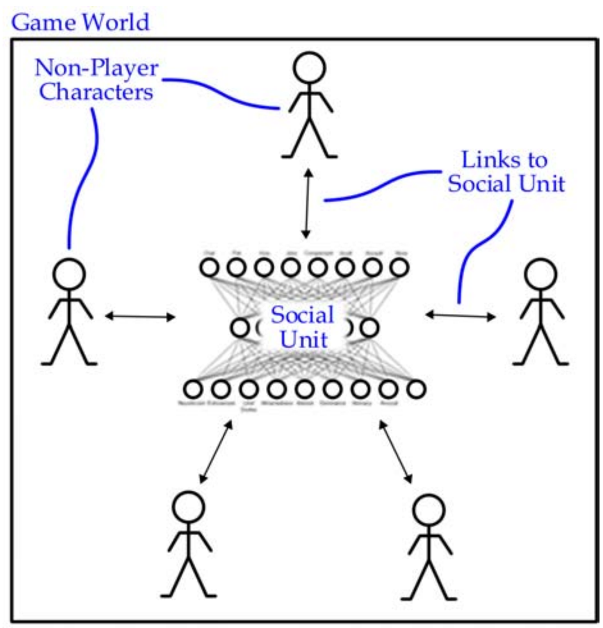


Figure 6: An illustration of how the µ-SIC system is incorporated into a virtual world

In addition to these high accuracies it is worth noting that when the system produces incorrect predictions, these are rarely significantly wrong. For example, the system may produce a FLIRT interaction rather than a KISS interaction, but will never produce an ASSAULT interaction instead of a KISS interaction.

Only one copy of the µ-SIC system is ever created and stored within the game engine when the PPA architecture is used to drive NPC behaviour. Whenever NPCs are free to begin an interaction they query this one copy of the µ-SIC system. For this reason the system can be considered

[1] Cross validation involves repeatedly training and testing a classifier on an initial data set. Each time the network is trained a portion of the data set is not used in training so as it can be used to test the network's performance on the data used in training. This way a realistic idea of the network's performance can be determined.

an oracle that advises NPCs on how to behave (see figure 6).

The reason that the system is implemented in this way is to minimise the storage requirements. If each agent were to store their own copy of the network then the amount of space used would quickly escalate as more and more agents were added to the simulation. By using just one copy of the network we avoid this massive storage requirement without changing the behaviour of the system.

## SIMULATION EXAMPLE EVALUATION

To evaluate the performance of the μ-SIC system a simulation example which takes place in a small town has been constructed. The simulation is visually represented through a very simple, two dimensional, cartoon style graphical environment. Key environments within the simulation include a bar, a restaurant, an office block and a police station.

The simulated town is populated by a range of NPCs, implemented using the PPA architecture. The simulation runs continuously in real-time with NPCs moving between the different locations in the simulation environment, adopting different roles based on their individual schedules.

Each of the characters in the simulation have been given a personality based on the Eysenck model, and are free to interact with one another (as determined by their adopted roles) at various times during the simulation. As the simulation progresses, characters' moods and relationships evolve based on the interactions in which they take part.

A range of screen-shots of this simulation are shown in figure 7. Each of the images illustrates how a different interaction can arise between two characters. The characters involved in the interaction in question are highlighted in each of the images.

Below each of the screen-shots, details of the character who instigates the interaction are shown, along with the details of their relationship with the other character involved in the interaction.

This information is also displayed in the information panel across the bottom of the screen-shot. On the left of this panel the character currently selected within the simulation is shown along with the character's name. In all of the images shown in figure 7 the character selected is the character who has instigated the interaction in question.

To the right of the character image the character's personality mood and appearance are displayed in the red and green bars. The personality bars show the character's extroversion on the top bar, and the introversion on the lower bar. The character's mood is shown with valence on the top bar and arousal on the lower bar. The appearance bar shows a measure of the character's appearance which is used to determine characters' attraction to each other.

Below these character details the character's current interaction along, along with the other characters involved in this interaction shown. Finally, to the right of the all of these details a list of the selected character's relationships is given. Relationships details are shown in the red and green bars in the order of Like/Dislike, Intimacy, Interestedness, Attractedness and Dominant/Submissive.

**Situation A**

Moe performs a CHAT interaction with Sean


**Personality:** Aggressive **Mood:** Fine

**Relationship**

**Like/Dislike:** Okay **Intimacy:** Low

**Interest:** Okay **Attractedness:** None

**Dom/Sub:** Low Submissive

**Interaction: Chat**

**Situation B**

Moe performs an ASSAULT interaction with George


**Personality:** Aggressive **Mood:** Angry

**Relationship**

**Like/Dislike:** Dislike **Intimacy:** Low

**Interest:** Okay **Attractedness:** None

**Dom/Sub:** Low Submissive

**Interaction: Assault**

**Situation C**

The Boss performs a COMPLIMENT interaction with Mary


**Personality:** Leader **Mood:** Good

**Relationship**

**Like/Dislike:** Okay **Intimacy:** Low

**Interest:** Okay **Attractedness:** High

**Dom/Sub:** Submissive

**Interaction: Compliment**

**Situation D**

Niamh performs an INSULT interaction with the Player


**Personality:** Thoughtful **Mood:** Bad

**Relationship**

**Like/Dislike:** Dislikes **Intimacy:** Medium

**Interest:** Quite High **Attractedness:** Low

**Dom/Sub:** Low Submissive

**Interaction: Insult**

**Situation E**

Bob performs a JOKE interaction with the Chief


**Personality:** Active **Mood:** Very Happy

**Relationship**

**Like/Dislike:** Likes **Intimacy:** Low

**Interest:** Okay **Attractedness:** None

**Dom/Sub: Equal**

**Interaction: Joke**

**Situation F**

Allison performs an FLIRT interaction with the Player


**Personality:** Social **Mood:** Okay

**Relationship**

**Like/Dislike:** Likes **Intimacy:** Low

**Interest:** Okay **Attractedness:** High

**Dom/Sub:** Equal

**Interaction: Flirt**

Figure 7: A range of screen-shots showing how the various interactions possible within the µ-SIC system arise as a simulation evolves

As the simulation progresses it successfully demonstrates the full range of interactions possible within the system. The simulation also demonstrates how relationships between the characters evolve as the simulation progresses, leading to different interactions. Related to this there is also an emergent property of the system through which characters will continually interact with characters they know well, leading to the emergence of groups. Finally, and possibly the most important feature of the system, is that it remains consistent throughout the simulation with all interactions being reasonably explained through the current values of the models used.

One drawback of the system is that at times characters can become stuck repeatedly performing the same interaction. The reason for this is that, although an interaction will always cause some changes to the characters mood and relationship details, these changes can sometimes be quite minor. If this is the case, then the next time the μ-SIC system is queried for an interaction, the inputs will still be very similar to the last time and so the system might well produce the same answer. This is fine for something like a CHAT interaction, as the subjects might well change between interactions, but can look a little disconcerting when the repeated action is a JOKE. A solution to this problem will be suggested in the following section.

It has been previously mentioned that the μ-SIC system runs in real-time. Often, the use of an ANN within a real-time system gives rise to efficiency concerns from developers. However, this is a wholly unjustified reaction. Although it is true that training an ANN is an intensive task (training the ANN within the μ-SIC takes approximately 20 minutes[2]), this is not true of querying a network. A query to the μ-SIC system takes approximately eight micro-seconds, which can be considered negligible within a game engine. Furthermore, characters only query the μ-SIC system when their roles allow them to socially engage, so the frequency of queries to the system is in the region of several queries per minute.

## CONCLUSIONS

The purpose of this work is to develop a system which can be used within a larger agent architecture to allow NPCs within computer games perform social interactions with other NPCs or players. These interactions ought to be based on the characters' personalities, moods and inter-personal relationships and remain consistent at all times throughout the game. The system achieves this by simulating these aspects of a character's persona using quantitative models from psychology. These models are used as inputs to an ANN which determines which interactions the character should engage in at a particular time. This ANN has been trained with a data set generated from a small set of hand crafted interactions. The μ-SIC system successfully performs a comprehensive range of social interactions based on the data set produced, and a simulation example has been created to demonstrate this.

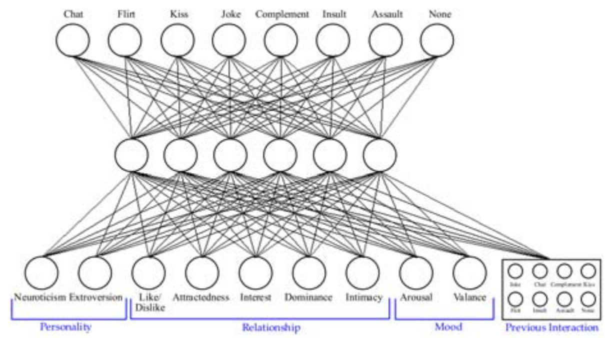


Figure 8: An extended version of the ANN used with the μ-SIC system. An extra input is added to the network to indicate the last interaction in which the character was engaged

Although the system is quite successful in its present state, one addition to the system has been identified which could improve it considerably. As was mentioned previously, characters can become stuck repeating the same interaction over and over. In order to overcome this problem an extra set of inputs could be added to the network to indicate the interaction in which the character has just been involved. An illustration of this new extended network is shown in figure 8.

By adding these extra inputs to the network, context would be explicitly added to the model. This would allow interactions to evolve through different interaction modes without the repetitive behaviour that is sometimes a feature of the current system.

The major drawback to this extension to the model is that an order of magnitude more data is required for training. As previously discussed, data acquisition is a difficult issue for the μ-SIC system as no databases exist to model the problem. However, it is hoped that by using the techniques described previously a selection of hand made data samples can be created and a data set can be generated by perturbing these samples.

Figure 9: A screenshot of the ALOHA system showing a number of characters engaged in conversation at a table in a bar

At present the μ-SIC system is being demonstrated within a very simple graphic environment. Interactions are shown

---

[2] Timings refer to the system running on a PC using a Pentium 4 2GHz processor with 512 MB of RAM.

simply through speech bubbles above the character's heads. This is suitable for development purposes, however for true evaluation of the believability of the system, a more elaborate graphical representation is required. This will lend a higher level of believability to the appearance of the characters and so users will automatically raise their level of expectation of the behaviour of the characters.

To achieve this, the entire PPA architecture is being incorporated into the Adaptive Level Of Detail Human Animation (ALOHA) system developed by the Image Synthesis Group from Trinity College (Giang, 2000; Mac Namee et al, 2002). A number of screenshots of this system (which already incorporates the role passing unit of the PPA architecture) are shown in figures 9 and 10.

Figure 10: A screenshot of the ALOHA system showing an entire bar scene

**REFERENCES**

Bishop, C.M., "Neural Networks for Pattern Recognition", Clarendon Press, Oxford, 1995.

Caicedo, A. & D. Thalmann, "Virtual Humanoids: Let them be autonomous without losing control", The Fourth International Conference on Computer Graphics and Artificial Intelligence, 2000.

Eysenck, H.J. & S. Rachman, "The Causes and Cures of Neuroses", London: Routledge and Kegan Paul. 1965.

Fairclough, C., M. Fagan, B. Mac Namee & P. Cunningham, "Research Directions for AI in Computer Games", *In D. O'Donoghue (Ed.), Proc. of the 12th Irish Conference on AI and Cognitive Science*, (pp 333 – 344), 2001.

Giang, T., R. Mooney, C. Peters, C. O'Sullivan, "ALOHA: Adaptive Level Of Detail for Human Animation. Towards a new Framework", Eurographics 2000, Short Papers Programme.

Lang, P. J., "The Emotion Probe: Studies of motivation and attention", A study in the Neuroscience of Love and Hate. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 1995.

Lebowitz, M., "Story-Telling as Planning and Learning", *In Poetics*. Vol. 14. No. 6. December 1985.

Lloyd, P., A. Mayes, A.S.R. Manstead, P.R. Meudell & H.L. Wagner, "Introduction to Psychology: An Integrated Approach", Fontana Paperbacks, 1984.

Mac Namee, B. & P. Cunningham, "Proposal for an Agent Architecture for Proactive Persistent Non Player Characters", *In D. O'Donoghue (Ed.), Proc. of the 12th Irish Conference on AI and Cognitive Science*, (pp 221 – 232), 2001.

Mac Namee, B., Dobbyn, S., Cunningham, P. and O'Sullivan, C., "Men Behaving Appropriately - Integrating the Role Passing Technique into the ALOHA System", *In (R. Aylett & L. Cañamero Eds.), Proceedings of Animating Expressive Characters for Social interactions, Symposium of the AISB'02 Convention*, (pp 59 – 62), Imperial College, London, 2002.

Mateas, M., "An Oz-Centric Review of Interactive Drama and Believable Agents", Technical Report CMU-CS-97-156, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 1997.

McCrae, R. R., & P. T. Costa Jr., "Toward a new generation of personality theories: Theoretical contexts for the five-factor model", *In J. S. Wiggins (Ed.), The five-factor model of personality: Theoretical perspectives* (pp. 51-87). New York: Guilford. 1996.

Meehan, J., "The Metanovel: Writing Stories by Computer", Research Report #74. Computer Science Department, Yale University. New Haven, CT. 1976.

Picard, R.W., "Affective computing", Perceptual Computing TR 321, MIT Media Lab, 1995.

Russell, S. & P. Norvig, "Artificial Intelligence: A Modern Approach", Prentice Hall, 1995.

Scott Neal Reilly, W., "Believable Social and Emotional Agents", Ph.D. Thesis. Technical Report CMU-CS-96-138, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 1996.

Wish, M., M. Deutsch & S. Kaplan, "Perceived Dimensions of Interpersonal Relations". *In Journal of Personality and Social Psychology*. Vol. 33. No. 6. April 1976.

Wooldridge M. & N. R. Jennings, "Intelligent Agents: Theory and Practice". *The Knowledge Engineering Review*, 10(2):115--152, 1995.