A. Egges, X. Zhang, S. Kshirsagar, N. M. Thalmann. Emotional Communication with Virtual Humans. Multimedia Modelling, Taiwan. 2003.

Emotional communication with virtual humans

Arjan Egges, Xuan Zhang, Sumedha Kshirsagar and **Nadia Magnenat-Thalmann**
MIRALab, University of Geneva
Phone: +41 22 705 77 69
Fax: +41 22 705 77 80
Email: {egges,xuan,sumedha,thalmann}@miralab.unige.ch

Abstract

In this paper, we present our approach to modelling perceptive 3D virtual characters with emotion and personality. The characters are powered by a dialogue system that consists of a large set of basic interactions between the user and the computer. These interactions are encoded in finite state machines. The system is integrated with an expression recognition system, that tracks a user's face in real-time and obtains expression data. Also, the system includes a personality and emotion simulator, so that the character responds naturally to both the speech and the facial expressions of the user. The virtual character is represented by a 3D face that performs the speech and facial animation in real-time, together with the appropriate facial expressions.

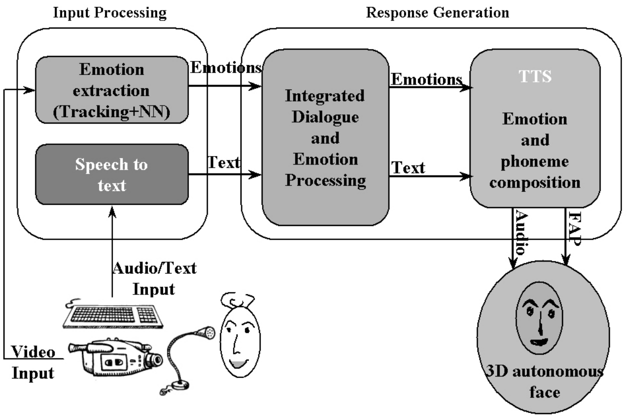


Figure 1: Facial Communication System

1 Introduction

With the emergence of 3D graphics, we are now able to create very realistic 3D characters that can move and talk. Further, multimodal interaction with such characters is possible as various technologies mature for speech and video analysis, natural language dialogues and animation. However, the behavior expressed by these characters is far from believable in a lot of systems. We feel that this problem arises due to their lack of *individuality* on various levels: perception, dialogue and expression. This paper describes our preliminary results in an experiment that tries to realistically connect personality and 3D characters not only on an expressive level (for example generating individualized expressions on a 3D face), but also on a *dialogue* level (generating responses that actually correspond to what a certain personality in a certain emotional state would say) and on a *perceptive* level (having a virtual character that uses expression data of a user to create corresponding behavior).

The idea of linking personality with agent behavior is not new. For example, [17] discusses influence of emotion on behavior in general, and [11] describes how personality and emotion can affect decision making. We do not intend to give a complete overview of ongoing research in behavior, emotion and personality, but we refer to a survey on the topic [20]. Our main goal is to create believable *conversational* agents that can interact with many modalities. We thus concentrate on emotion extraction of a real user, emotions in the dialogue systems and display of the resulting emotional behavior on a 3D face. In this paper, we explain our approach with the help of an integrated system for communicating with a virtual face. Figure 1 shows such a system, where input processing and response generation are the two main blocks. The perception of the facial expressions captured by a video camera results into extracting the emotional state of the user. The perception module uses a neural network trained on the facial expression data captured from several users. The speech input from the user is converted into text or the user types the text using a keyboard. Currently our system uses only the keyboard input, or there is a possibility of using available speech recognition software for speech-to-text. The response generation takes place in two steps. The dialogue system processes both text and emotions. The personality model is integrated with the dialogue system and appropriate "emotional text" is generated as a result. The "visual response" is generated from the text and emotions for the expressive speech animation of the 3D face in real time.

This paper is organized as follows. In the next section, we introduce the dialogue engine that we use to manage the conversation. Section 3 presents the perceptive part of our system. In Section 4, we discuss our system of personality, mood and emotional state. Section 5 presents the visual front-end that enables a user to actually *see* a virtual character with expressions corresponding to its internal state. Finally, we give some results and conclusions.

2 The Dialogue System

This section describes the architecture of our dialogue system and how dialogue can be integrated with other system parts, such as a 3D virtual human and a facial expression recognition system.

2.1 Introduction

One of the first attempts to create a system that interacts with a human through natural language dialogue, was ELIZA [27]. ELIZA is a system that generates standard answers, based on certain word combinations that it finds in a phrase entered by the user. A newer and more extended version of this approach is found in the ALICE program [1] that uses AIML (an XML based transcription language) to transcribe pattern-response relations. In general, we can distinguish three main approaches to dialogue modelling[1]:

Dialogue grammars This method defines sequences of sentences in a dialogue. Such a grammar can describe the whole dialogue from beginning to end, but these grammars can also be used to describe often occurring sequences in a dialogue, such as question-answer pairs. Dialogue specifications according to a grammar can be constructed using Chomsky rules or a finite state machine. In the latter case, state transitions often correspond to dialogue utterances.

Plan-based dialogues The idea of a plan-based approach for the listener is to discover the underlying plan of the speaker and respond properly to this. If such a plan is correctly identified, this approach can very well handle indirect speech acts. However, a plan-based approach does not explain *why* dialogue takes place: the dialogue manager has no explicit goal.

Collaborative dialogues A collaborative approach views a dialogue as a collaboration between partners in order to achieve a mutual understanding of the dialogue. This commitment of both participants is the reason that dialogues contain clarifications, confirmations and so on. Every partner has certain beliefs about the other partner and uses these beliefs to respond to him/her. An example of the application of such an approach is given in [25, 21].

2.2 Overview of our approach

The different approaches addressed in the previous sections have advantages and disadvantages. Dialogue grammars allow to write quick and simple dialogues, but their flexibility is low and they cannot retrieve very detailed information from natural language input. The collaborative approach requires a definition of desires and beliefs in a logical formalism. This also means in most of the cases that a logical interpreter of natural language should be included, such as for example the Core Language Engine [2].

Our system is based on the principle of **finite state machines** (FSMs). We can use them simply according to the dialogue grammar principle, but our system also provides a means to use them for a plan-based or collaborative approach. After we give a quick overview of the theory, we will present how we use the FSMs to create a flexible dialogue system.

2.3 Finite State Machines

A **deterministic** finite state machine (or deterministic finite automaton, DFA) is a quintuple $(Q, \Sigma, \delta, q_0, F)$ where:

- Q is a finite set of states;
- Σ a finite set called the alphabet;
- $q_0 \in Q$ the start state;
- F a subset of Q called the final or accepting states, and
- δ is the transition function, a total function from $Q \times \Sigma$ to Q.

A **nondeterministic** FSM is the same as a deterministic FSM, except for the transition function. A nondeterministic FSM has a transition function δ that is a total function from $Q \times \Sigma$ to $\mathcal{P}(Q)$, thus there can be more than one possible transition with the same input[2].

An extension to this FSM theory is FSMs that also generate output. They can do this in two ways: either output is associated with states (Moore machines) or the output is associated with transitions (Mealy machines). Another extension of nondeterministic FSMs is that they can make a transition without input being processed (this transition is called a λ-transition and the transition function δ is a total function from $Q \times (\Sigma \cup \lambda)$ to $\mathcal{P}(Q)$).

2.4 FSMs and Dialogue

In our system, conversations are modelled using Mealy nondeterministic finite state machines that can have λ-transitions and that generate output. The input alphabet consists of two values $\{0, 1\}$. The FSM output is directly linked with *actions* that can be extended by the user. The input that is received by an FSM is the result of an application of statements that either return 0 or 1 (false or true). From now on we will call these statements *conditions*.

One category of dialogue systems is the *dialogue grammar* (see Section 2.1). If we would use FSMs in the classic sense, then a transition in the finite state machine always corresponds to a step in the dialogue. In order to have this functionality, we need at least an action that can generate output and a condition that matches input to a certain pattern. In our XML definition, they can be defined as follows:

```
<condition type="input_match">
hello my name is *
</condition>

<action type="say">
hello how are you doing?
</action>
```

A very small dialogue system can now be constructed with three states q_0, q_1, q_2 where q_0 is the start state and q_2 is the end state. There would be two transitions, one from q_0 to q_1 (linked with the matching condition) and one from q_1 to q_2 (linked with the action that generates output). The action is linked with a module that provides the interface between FSM and user in the form of an output buffer. The condition is linked with a pattern matcher module and input buffer.

However, this approach gives us a very limited tool to create dialogue systems. What if we want to do more elaborate input processing? And what if we want to have a reasoning

[1] For a more complete survey, see [5].

[2] A good introductory textbook on formal language theory, which encapsulates finite automata, is [22].

engine at our disposal and the dialogue depends on the result of this reasoning process? How about topic-shifting and maintenance of dialogue history?

We believe that during the dialogue, different *kinds* of approaches are required. For example, having a small conversation about the weather required a lot less resources than a philosophical discussion about the meaning of life. Therefore we have developed a framework for building interactive systems that can include all of the functionalities possessed by the separate approaches. We define a dialogue system as a collection of small dialogue units; every unit handles a specific dialogue between the user and the computer. Every unit can use different resources to interpret natural language input, to reason if necessary, and so on. Dialogue units are defined using the FSMs. The dialogue system consists of a kernel of these FSMs that are running and a set of modules that can each perform specific tasks and that have an interface with the FSM kernel. For an overview, see Figure 2. The separate modules can be easily added to our system, together with XML readers for module specific conditions and actions, such as reading variables, check if a logical statement is valid, and so on. In sections 3 and 4, we will give examples of such modules and their corresponding conditions and actions.

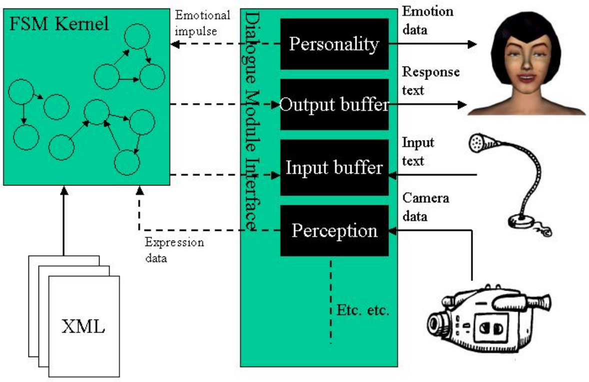


Figure 2: An overview of the dialogue system architecture.

2.5 The Parallel FSM Algorithm

As our FSMs are concurrently running in one system, we need to extend the definition of an FSM in order to add extra features that influence how the kernel chooses the FSM and the transition that has to be made. From now on, we will use $\mathcal{F}$ to indicate a Mealy nondeterministic finite state machine with lambda transitions. We will now **extend** the definition of $\mathcal{F}$ by defining an additional set of parameters (p, c, s) for it. Parameter p is an integer value in the interval $[0, \rightarrow\rangle$, that defines the *priority* of this FSM in the total system. FSMs with higher priority will be selected for performing a transition first. The second parameter, c, specifies if an FSM is *cyclic* or not. An FSM is called cyclic if it automatically goes to the start state when it reaches an accepting state (especially useful in dialogues for repeating conversation parts). The final parameter, s, specifies whether or not an FSM is *strong*. When an FSM is strong, and it is selected to perform a transition, it obtains a temporary priority of ∞, until it can no longer make a transition. This is a feature especially useful for dialogue. In practice it means that when user and computer are involved in a certain conversation (modelled by one FSM), then the computer will always interpret a user reply as part of the current conversation, if possible.

Two of the three parameters have a direct effect on how FSMs are selected during run-time: parameters s and p. In the following, we assume that we have a set of currently running FSMs V. Also there is a currently selected FSM F_s. Finally, the algorithm uses a set W to store the tentatively selected FSMs in.

```
$W = \emptyset$

for all $F_i \in V$
  if $F_i$ can make a transition
    if $W = \emptyset$
      $W = W \cup \{F_i\}$
    if $\forall k \cdot F_k \in W : \; p_i \geq p_k$
      $\forall k \cdot F_k \in W : \; p_k < p_i$:
          $W = W \backslash \{F_k\}$
      $W = W \cup \{F_i\}$

if $F_s$ can make a transition and $F_s$ is strong
  $W = \{F_s\}$

if $W \neq \emptyset$
   select a random $F_r \in W$
   $F_s = F_r$
```

If an FSM can be selected by this algorithm, then this FSM is allowed to do a transition. When the system is started, all FSMs are set into their start state q_0. Then we start a thread that checks if an FSM can be selected for transition with a certain time interval.

3 The Perceptive System

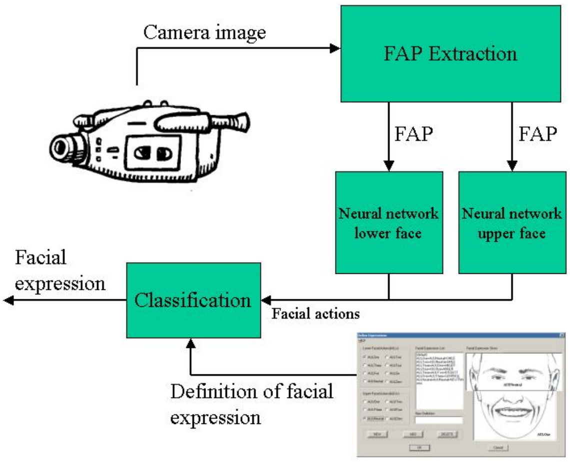
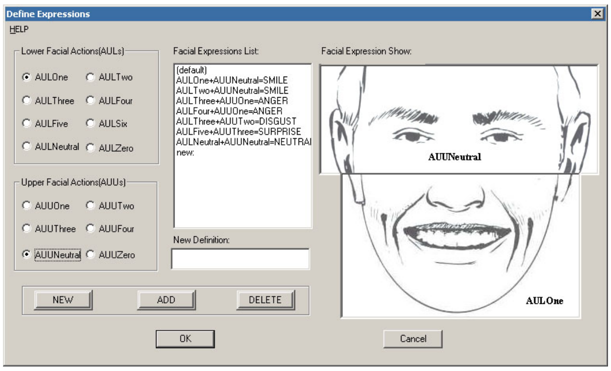

Camera image
FAP Extraction
FAP
FAP
Neural network lower face
Neural network upper face
Facial expression
Classification
Facial actions
Definition of facial expression

Figure 3: Overview of the perceptive system.

In this section, we introduce a novel approach to recognize facial expression. A real-time facial expression recognition system based on a set of facial features (eyebrows, eyes, mouth and nose) in a frontal-view face image sequence has been developed that can recognize human facial expression in real-time. Different from most facial expression recognition systems [19, 4, 7, 12, 26] that attempt to recognize a small set of template expressions, our system can recognize facial expressions that the users define *themselves*. In our system, facial expressions can be defined using several basic facial action units. This greatly enhances the flexibility of a facial expression recognition system in comparison to a system that only recognizes a few template expressions [16].

The user can define his/her individual facial expressions with a GUI by combining facial action units before recognition. The layout of this GUI is shown in Figure 4.



Define Expressions

Lower Facial Actions(AULs)
● AULOne ○ AULTwo
○ AULThree ○ AULFour
○ AULFive ○ AULSix
○ AULNeutral ○ AULZero

Upper Facial Actions(AUUs)
○ AUUOne ○ AUUTwo
○ AUUThree ○ AUUFour
● AUUNeutral ○ AUUZero

Facial Expressions List:
(default)
AULOne+AUUNeutral=SMILE
AULTwo+AUUNeutral=SMILE
AULThree+AUUOne=ANGER
AULFour+AUUOne=ANGER
AULThree+AUUTwo=DISGUST
AULFive+AUUThree=SURPRISE
AULNeutral+AUUNeutral=NEUTRAl
new:

New Definition:

Facial Expression Show:

Figure 4: GUI used for facial expression definition.

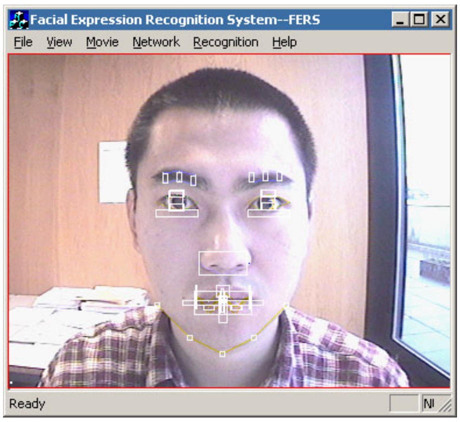
Our system consists of three parts. First, an automatic initialization of the position of the facial features is performed. Then, we can obtain MPEG-4 FAP data from these features in real-time with our tracking system (for more detail, see Section 3.2). From the FAP data, we recognize the facial action units for the upper and the lower face. Finally, a user-specific classifier determines the facial expression that is displayed (Section 3.3). Figure 3 shows an overview of the complete real-time perceptive system.

3.1 Initialization

The perception system automatically initializes the positions of all facial features (eyebrows, eyes, mouth and nose). Different from other systems that have to set initial positions of facial features by hand, the user can initialize the system by drawing a square to indicate the face region. The system then automatically locates the facial features (see Figure 5). Our system uses a 3D face model to match the face image. Through deforming the face model, edge and gray information around the facial features are used to calculate an energy level. This energy level then allows us to retrieve the proper positions of the facial features.

3.2 Facial feature tracking

In order to have a real-time facial expression recognition system, we need a real-time facial feature tracking system. However, many issues have to be resolved when developing such a system. One important problem lies in the variety of the appearances of individuals, such as skin color, eye color, beard, glasses and so on. In our system, the facial features and their associated information are set during an initialization phase that will solve the main problem in facial feature differences between people (see previous section). The tracking process is separated into two parts: mouth tracking and eye tracking. Mainly the edge and the gray level information around the mouth and the eyes are used during tracking. This work is based on the previous work by Goto *et al.* [10].



Facial Expression Recognition System--FERS

Ready

Figure 5: Initialization of the facial features.

Mouth tracking

The mouth is one of the most difficult facial features to analyze and track. Indeed, the mouth has a very versatile shape and almost every muscle of the lower face drives its motion. Furthermore, beard, mustache, the tongue or the teeth might appear sometimes and further increase the difficulty in tracking it. Our method is taking into account some intrinsic properties of the mouth:

1. upper teeth are attached to the head bone and therefore their position remains constant;
2. conversely, lower teeth move down from their initial position according to the rotation of the jaw joints;
3. basic mouth shape depends on bone movement.

From these properties it follows that the detection of the positions of hidden or apparent teeth from an image is the best way to make a robust tracking algorithm of the mouth shape and its associated motion.

The basic algorithm for tracking the mouth is that the system proceeds first with the extraction of all edges crossing a vertical line going from the nose to the jaw. In a second phase, the energy that shows the possibility of a mouth shape is calculated. Finally, among all possible mouth shapes, the best candidate is chosen according to a highest energy criterion. Figure 6 presents the gray level value along the vertical line from the nose to the jaw for different possible shape of the mouth and the corresponding detected edges.

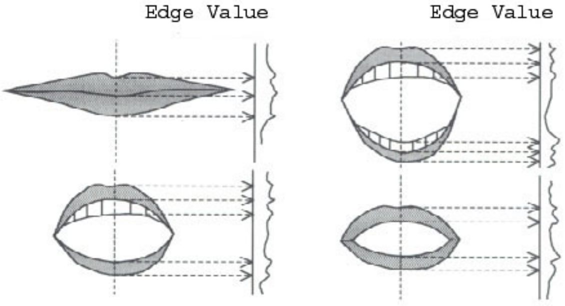


Figure 6: Edge configuration for possible mouth shapes.

Closed mouth In this case, the center edge appears strong, the other two edges appear normally weak, and teeth are hidden inside.

Opened mouth As shown in Figure 6, when teeth are present, the edges are stronger than the edge on the outside lips, or between a lip and the teeth or between the lip and the inside of the mouth. If the lips hide the teeth then of course the edge of the teeth is not detected.

Once this edge detection process is complete, the edge information is compared with data from a generic shape database and a selection of possible corresponding mouth shapes is done.

Eye tracking

The eye tracking system is responsible for the pupil tracking, the eyelid position recognition, and the eyebrow tracking. These three subsystems are highly interdependent. For example, if the eyelid is closed, the pupil position will be hidden, and it is obviously impossible to detect its position. In our first attempt, we considered a generic model of the eye [24], but the system was failing when the eye was closed, and it was difficult to obtain a stable result. We have improved this first method by:

1. calculating both the pupil positions;
2. calculating the eyebrow positions;
3. extracting eyelid positions with respect to their possible position, and
4. by checking for the presence of movement in any part of the feature tracking data.

When this analysis is finished, inconsistencies are checked again and a new best position is chosen if necessary.

For the pupil tracking, the same kind of energy functions used for the mouth are applied. In the eyebrow case, a vertical line goes down from the forehead until the eyebrow is detected. The eyebrow position is given by the maximum energy calculated with edge value and pixel value. After the center of the eyebrow is found, the edge of the brow is followed to the left and right to recognize the shape. This method is similar in the case of the closed mouth as described before, but with the advantage of using the pixel information. It is possible to estimate eyelid location after detecting the pupil and eyebrow locations. This estimation improves the detection of the correct eyelid position opposed to a possible wrong detection that may occur with a wrinkle.

FAP conversion

The movement information data of the facial feature points extracted by tracking is converted to MPEG-4 based FAPs (Facial Animation Parameters) [10]. A facial expression is represented in FAPs by giving the distance of every facial feature point from its original position in the neutral face. This neutral face is captured during the initialization phase. The FAP values are normalized using FAPU. For example, the standard FAPU mouth width (MWo) is 1024 pixels. When we have obtained a mouth width from the initialization phase of x pixels, and during tracking, a feature point on the mouth has moved y pixels, then the normalized FAP value is given by $\frac{1024 \cdot y}{x}$.

All movement data of facial feature points is converted into normalized FAP by using the FAPUs that correspond to the region of the face that the facial feature point is in. Eventually, we obtain a complete set of FAPs that represents a facial expression. Figure 7 shows the real-time tracking of the *surprise* expression.

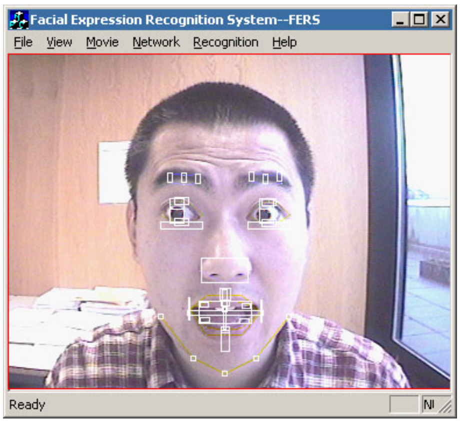


Figure 7: Real-time tracking of the *surprise* expression.

3.3 Recognition of facial expression

One important observation of human facial expressions is that they are never exactly the same for everyone. Building a facial expression recognition system that only recognizes template facial expressions will therefore not give very satisfactory results in most cases. A second problem of these kinds of systems is that they are not easily extended to recognize additional facial expressions. Recognizing someone's facial expression needs to be done by a system tailored to this particular user. Of course, building a different facial expression recognition system for everyone is not a solution to the problem. Our system can be tailored to any user, by letting *him/her* define his/her facial expressions.

According to the Facial Action Coding System (FACS) [9], facial expressions consist of several *facial action units*. These basic building blocks are the same for everyone. Facial action units can be either upper facial action units or lower facial action units (the categories are almost independent). So if we can recognize the upper facial actions and the lower facial actions [23], we will be able to recognize the facial expressions that are consisted by them. Different users can indicate the upper and lower facial actions that their expressions are constructed of, using the GUI shown in Figure 4. Based on this idea, we have designed our facial expression recognition system. The FAP data obtained by the facial feature tracking system is classified into four upper facial actions and six lower facial actions. This classification is done by two neural networks: one for the upper facial actions, and one for the lower facial actions. After a user has indicated which combinations of facial actions correspond to his expressions, we can easily determine the tailored facial expressions of this user from the facial actions obtained by the classified facial feature tracking data.

3.4 Interface with the dialogue system

The dialogue system requires an interface with the perceptive system in order to get the information. In our prototype system, the only information that the dialogue system gets, is the conceptual information: what is the current expression of the user? This data can be easily passed by a condition that can be described in XML as follows:

```
<condition type="user_expression">
joy
</condition>
```

4 Personality and Emotional State

In addition to the dialogue framework presented in Section 2, we have also developed a module to simulate personality and emotion, as well as an interface with the dialogue system. This section discusses the various features of this personality module.

4.1 Introduction

When we talk about personality, we use various related concepts:

- Personality model: this is a model that—in our case—represents aspects of the personality as dimensions in a 5-dimensional space, according to the OCEAN paradigm [8];
- Emotional state: a set of emotions that are in a certain state of arousal (we use the OCC model [18] + two additional emotions *disgust* and *surprise*);
- Mood: the mood lies between the personality and the emotional state, and is either negative, neutral or positive $\{-1, 0, 1\}$; it is less fluent than the emotional arousal which can change very quickly [14];
- Emotional impulse: the emotional state is changed by giving it an impulse; the impulse works on a certain emotion with a certain level; the influence on the emotional state then depends on this impulse, the personality, the mood, and the previous value of emotional arousal;

Figure 8 gives a general overview of the components of the personality module that we have developed. Emotional impulses are defined by giving the emotion name and a level in the range $[-5, 5]\backslash\{0\}$[3]. An example is *joy+3* or *disappointment-5*. The emotional impulses are generated automatically by the FSM, depending on the dialogue state and the context. Positive values indicate an increased arousal; negative values decrease the arousal.

An emotional impulse is processed in two steps. First, the *mood* is updated using a Baseyian Belief Network. This step is explained in more detail in Section 4.2. The second step calculates the influence of the emotional impulse on the *emotional state*. This calculation is performed by an *affector* (see Section 4.3).

4.2 Updating the mood

We use a Bayesian Belief Network (BBN) for probabilistic mood processing. A BBN is a directed acyclic graph. Each node in this graph represents a state variable with mutually exclusive and independent possible states. The directed links represent the influence of the parent node on the child node. For each child node, a Conditional Probability Table (CPT) defines the probability that the child node assumes different states for each combination of the possible states of the parent nodes. [3] have previously used BBN for personality modelling. However, they did not use mood in the processing, and used personality to directly affect the emotional states.

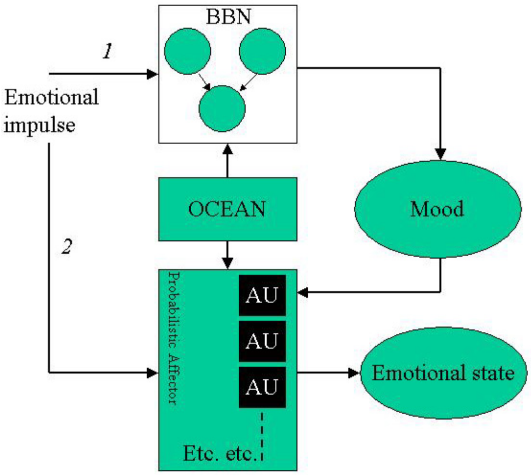


Figure 8: An overview of the personality system.

Figure 9 shows our BBN that has two parent nodes and a child node indicating the current mood, the response mood (or the expected mood) and the changed mood respectively. The response or expected mood is decided by the emotional impulse from the dialogue FSM. This simply indicates a positive mood if a particular response is expected to be said in a positive emotion (*happy, hopeful etc.*) or a negative mood if the associated emotion is negative (*sad, disappointed etc.*). The CPT for the mood BBN defines the probability of next mood given the current mood and the response mood. As a result, we get the probability of the next mood being positive, neutral or negative. The mood having the highest probability is selected as the possible candidate for mood change. We perform a random test using the probability value for the selected mood assuming a normal distribution. The computed next mood is is subsequently used for updating the emotional state.

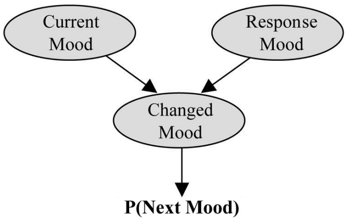


Figure 9: BBN for Mood Processing

We have one BBN for each factor of the personality, and while combining the factors, a weighted average of the corresponding CPTs results into the desired BBN definition of the new personality.

4.3 Affectors

An affector defines how an emotional impulse influences the emotional state. It has to take certain parameters into account, such as personality and mood, as well as the current emotional state. In this section we explain how the affector behaves with respect to the mood and personality and how the history of the emotional changes are reflected in the emotion appraisal.

[3] A value of zero is not allowed since this would mean no impulse at all.

We express emotional arousal in a percentage 0-100%. Suppose we give an emotional impulse Ex where E is the emotion and x is the level. A simple affector could then calculate the arousal change for emotion E in the emotional state as follows[4,5]:

$$\Delta E = 20x \qquad (1)$$

Using this affector, the impulse *joy+3* would result in an emotional arousal percentage increase of $+60$. However, this affector does not take into account the personality or mood of an individual. We have designed another affector that defines an *Affection Unit* (AU) for each emotion. Each AU is designed to update one emotion E. It consists of a set of parameters that are used to process an emotional impulse Ex.

The change of emotional state depends on two factors: the new mood (see Section 4.2) and the personality. Since emotional arousal is expressed as a percentage in our system, the personality module calculates the desired *change of this percentage*, which we will call ΔP from now on. This value consists of two partial changes (one for the mood and one for the personality)

$$\Delta P = \Delta P_M + \Delta P_P \qquad (2)$$

where ΔP_M is the change that depends on the mood and ΔP_P is the change that depends on the personality.

For every emotion E, we have defined a constant value λ_E that defines how the mood influences the change of the emotional state for emotion E. Given the new mood (which is either -1, 0 or 1) we calculate a percentage change ΔP_M of emotion E as follows:

$$\Delta P_M = M\lambda_E \qquad (3)$$

We also define a set of parameters θ that define how personality influences the emotional arousal. For each dimension of personality according to the OCEAN model, $\theta_{E,p}$ defines a weight of personality factor p for emotion E. For example, $\theta_{joy,O}$ is the weight of personality factor *O* for the *joy* emotion. The contribution ΔP_P to the change in emotion by the personality model is calculated as follows:

$$\Delta P_P = \sum_{i\in\{O,C,E,A,N\}} \frac{W_i\theta_{E,p}}{100} \qquad (4)$$

where W_i denotes the percentage of each dimension of personality. Using Equation 2, we then obtain a ΔP. If this value is greater than 1, we define $\Delta P = 1$; if it is smaller than 0, we define $\Delta P = 0$. We use ΔP as a chance in a probabilistic system with a normal distribution where $\mu = \Delta P \cdot 20 \cdot x$. We then do a random test every time an emotional impulse is given with the calculated chance ΔP. The percentage that we thus obtain, is the change of the percentage of arousal of the corresponding emotion.

4.4 Decay of emotional arousal

Over time, the emotional arousal decays and will become zero again. The shape of the decay curve depends among others on the personality and the mood of an individual. In our current implementation, the decay is linear over time, where a constant value is used to control the speed of the decay. In the future, we intend to explore more elaborate methods of decaying emotional arousal that take into account the personality and the current mood of the individual.

4.5 Interface with the dialogue system

The personality module is interfaced with the dialogue system through conditions and actions that can be specified in XML. The FSMs from the dialogue system can generate an emotional impulse by specifying the following action:

```
<action type="emotional_impulse">
fear+3
</action>
```

Also, an FSM can check if an emotional state is below or above a certain threshold by specifying the following condition:

```
<condition type="above_threshold">
<subject>fear</subject>
<value>40</value>
</condition>
```

When the dialogue system has to utter a response, data from the emotional state is used to *tag* the output text. These emotional tags are passed along with the response to the visual front-end.

5 Visualization

Our graphical front-end comprises of a 3D talking head capable of rendering speech and facial expressions in synchrony with synthetic speech. The facial animation system interprets the emotional tags in the responses, generates lip movements for the speech and blends the appropriate expressions for rendering in real-time with lip synchronization. Facial dynamics are considered during the expression change, and appropriate temporal transition functions are selected for facial animation. The snapshots of the facial expressions used by our system are shown in the Figure 10.

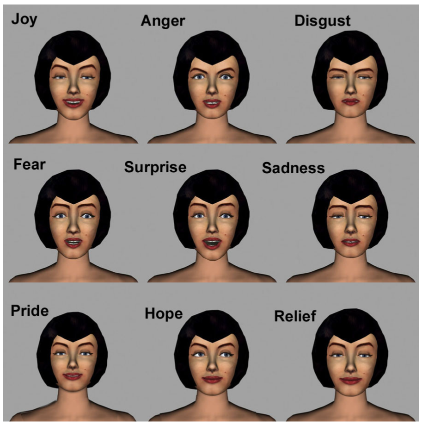


Figure 10: Various Expressions for the Virtual Human

We use MPEG-4 Facial Animation Parameters as low level facial deformation parameters. The details of the deformation algorithm are explained in [13]. However, for defining the visemes and expressions, we use the Principal Components (PCs) as described by Kshirsagar *et al.* [15]. The PCs

[4] We assume that an emotional impulse for emotion E does not influence any other emotion than E itself.

[5] By choosing the value of 20 in the equation, the maximum percentage change is automatically 100% since the level of an emotional impulse is in the interval $[-5, 5]$.

are derived from the statistical analysis of the facial motion data and reflect independent facial movements observed during fluent speech. They are used as high level parameters for defining the facial expressions and visemes. The use of PCs facilitates realistic speech animation, especially blended with various facial expressions. The main steps incorporated in the visual front-end are the following:

1. Generation of FAPs from text: For this we use available text-to-speech (TTS) software that provides phonemes with temporal information. The co-articulation rules are applied based on the algorithm of Cohen *et al.* [6] that is adopted for use with the PCs.
2. Expression blending: The dialogue system outputs expression tags with the text response. Each expression is associated with an intensity value. An attack-sustain-decay-release type of envelope is applied for the expressions and it is blended with the previously calculated co-articulated phoneme trajectories. This blending is based on observed facial dynamics, incorporating the constraints on facial movements wherever necessary in order to avoid excessive/unrealistic deformations.
3. Periodic facial movements: Periodic eye-blinks and minor head movements are applied to the face for increased believability. In future, these movements should be extracted also from the dialogue module, considering their relation with the semantics and context. Periodic display of facial expression is also incorporated, that depends on the recent expression displayed with the speech, as well as the mood of the character.

6 Conclusions and future work

We have developed a framework for building flexible dialogue systems that are easy to extend with additional functionalities. The dialogue is entirely defined in XML and is thus easy to transfer and read by others. We also have developed a computational model of personality and affection. Furthermore, we have integrated the system with a robust facial expression recognition system. In order to demonstrate the system, we are currently developing a simple application of a virtual acting tutor. The tutor asks the user to perform certain facial expressions and it responds to the user by giving an appraisal of the quality of the facial expression. The tutor has a personality of its own and it becomes angry or sad when the user fails to make a correct expression. The emotional state and mood of the tutor are expressed in a 3D face, that also performs the speech synchronized with a text-to-speech engine.

In the future, we intend to improve the affection of personality and mood on emotional arousal, as well as the decay of the emotional arousal. We will perform experiments with different dialogue systems and personalities to validate the results of our research. Also, the facial expression recognition system can be further improved by automatically detecting the face region and by taking into account the head movement during the tracking process.

7 Acknowledgement

This research was partly funded by the European RTN project MUHCI (Multi-modal Human Computer Interaction) with project number HPRN-CT-2000-00111. For more information about this project, see the project home page *http://www.muhci.org/*.

References

[1] AIML webpage. http://alice.sunlitsurf.com/, November 2002.

[2] Hiyan Alshawi, editor. *The Core language engine*. MIT Press, 1992.

[3] G. Ball and J. Breese. Emotion and personality in a conversational character. In *Proceedings of the Workshop on Embodied Conversational Characters*, pages 83–84 and 119–121, October 1998.

[4] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal Computer Vision*, 25(1):23–48, 1997.

[5] Gavin E. Churcher, Eric S. Atwell, and Clive Souter. Dialogue management systems: a survey and overview. Technical report, University of Leeds, February 1997.

[6] M. M. Cohen and D.W. Massaro. *Modelling co-articulation in synthetic visual speech*, pages 139–156. Springer-Verlag, 1993.

[7] J. F. Cohn, A. J. Zlochower, J. J. Lien, and T. Kanade. Feature-point tracking by optical flow discriminates subtle differences in facial expression. In *Proceedings International Conference Automatic Face and Gesture Recognition*, pages 396–401, 1998.

[8] P. T. Costa and R. R. McCrae. Normal personality assessment in clinical practice: The NEO personality inventory. *Psychological Assessment*, (4):5–13, 1992.

[9] P. Ekman and W. V. Friesen. *Facial Action Coding System: Investigators Guide*. Consulting Psychologists Press, Palo Alto, CA, 1978.

[10] Taro Goto, Marc Escher, Christian Zanardi, and Nadia Magnenat-Thalmann. MPEG-4 based animation with face feature tracking. In *CAS '99 (Eurographics workshop)*, pages 89–98. Springer, September 1999.

[11] Michael Johns and Barry G. Silverman. How emotions and personality effect the utility of alternative decisions: a terrorist target selection case study. In *Tenth Conference On Computer Generated Forces and Behavioral Representation*, May 2001.

[12] S. Kimura and M. Yachida. Facial expression recognition and its degree estimation. In *Proceedings Computer Vision and Pattern recognition*, pages 295–300, 1997.

[13] S. Kshirsagar, S. Garchery, and N. Magnenat-Thalmann. *Deformable Avatars*, chapter Feature Point Based Mesh Deformation Applied to MPEG-4 Facial Animation, pages 33–43. Kluwer Academic Publishers, July 2001.

[14] S. Kshirsagar and N. Magnenat-Thalmann. A multilayer personality model. In *Proceedings of the second International Symposium on Smart Graphics*, pages 107–115. ACM Press, June 2002.

[15] S. Kshirsagar, T. Molet, and N. Magnenat-Thalmann. Principal components of expressive speech animation. In *Proceedings Computer Graphics International*, pages 59–69, 2001.

[16] J. J. Lien, T. Kanade, J. F. Cohn, and C. C. Li. Automated facial expression recognition based on facs action units. In *Proceedings International Conference Automatic Face and Gesture Recognition*, 1998.

[17] Stacy Marsella and Jonathan Gratch. A step towards irrationality: Using emotion to change belief. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, July 2002.

[18] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[19] M. Pantic and L. M. Rothkrantz. Automatic analysis of facial expressions: the state of the art. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(12), 2001.

[20] Paul Piwek. An annotated bibliography of affective natural language generation. Technical report, University of Brighton, July 2002.

[21] M. D. Sadek, P. Bretier, and F. Panaget. ARTIMIS: Natural dialogue meets rational agency. In M. E. Pollack, editor, *Proceedings 15th International Joint Conference on Artificial Intelligence*, pages 1030–1035. Morgan Kaufmann Publishers, 1997.

[22] Thomas A. Sudkamp. *Languages and Machines: an introduction to the theory of computer science*. Addison Wesley, 1994.

[23] Y. L. Tian, T. Kanade, and J.F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 2001.

[24] M. Tistarelli and E. Grosso. Active vision-based face authentication. *Image Vision Computer*, 18(4), 2000.

[25] David R. Traum. Conversational agency: The TRAINS-93 dialogue manager. In *Proceedings of the Twente Workshop on Langauge Technology: Dialogue Management in Natural Language Systems (TWLT 11)*, pages 1–11, 1996.

[26] M. Wang, Y. Iwai, and M. Yachida. Expression recognition from time-sequential facial images by use of expression change model. In *Proceedings International Conference Automatic face and Gesture Recognition*, pages 324–329, 1998.

[27] J. Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.