ParleE: An Adaptive Plan Based Event Appraisal Model of Emotions

The Duy Bui, Dirk Heylen, Mannes Poel, and Anton Nijholt

University of Twente
Department of Computer Science
The Netherlands
{theduy,heylen,mpoel,anijholt}@cs.utwente.nl

Abstract. We propose ParleE, a quantitative, flexible and adaptive model of emotions for a conversational agent in a multi-agent environment capable of multi-modal communication. ParleE appraises events based on learning and a probabilistic planning algorithm. ParleE also models personality and motivational states and their role in determining the way the agent experiences emotion.

1 Introduction

In this paper, we describe ParleE, a quantitative, flexible and adaptive model of emotions for an embodied conversational agent situated in a multi-agent environment that can engage in multi-modal communication. ParleE is developed in order to enable the conversational agent to respond to events with the appropriate expressions of emotions with different intensities.

Research in our Parlevink group at the University of Twente has focused on human-computer interaction in virtual environments ([7], [8]). Our aim is to build believable agents for several application areas: information, transaction, education, tutoring and e-commerce. For embodied agents to be believable, the minds of agents should not be restricted to model reasoning, intelligence and knowledge but also emotions and personality. Furthermore, it is necessary to pay attention not only to the agent's capacities for natural language interaction but also to its non-verbal aspects of expression. We propose ParleE for this purpose.

ParleE is built upon existing computational models of emotions: Elliott's Affective Reasoner [3], Velásquez's Cathexis [11], El-Nasr et al's FLAME [4] and Gratch's Émile [6]. Like many of these and other models, ParleE generates emotions based on Ortony et al.'s theory of appraisal [9]. ParleE appraises events based on learning and a probabilistic planning algorithm. We also pay attention to the integration of personality and motivational states into ParleE.

Section 2 will describe the requirements for ParleE and compare how existing models suit these requirements. Section 3 will discuss the ParleE model. Section 4 will show how to calculate the intensity of emotions. The model of personality will be discussed in Sect. 5. Finally, the learning components and an illustration of the model are presented in Sects. 6 and 7 respectively.

M. Jarke et al. (Eds.): KI 2002, LNAI 2479, pp. 129–143, 2002.

2 From Requirements, Existing Models to ParleE

To be consistent with the aim of building embodied agents with different personalities for various application areas, we introduce a set of requirements for our model. We then give a brief review of existing models to see how they can fit in our requirements.

2.1 Requirements

The requirements for the proposed system are:

Quantitative Model: The model should produce emotions with different intensities in an agent. The emotion intensities should also decay over time.

Flexible Appraisal Model: The model should be an event appraisal model. Moreover, the model should have a generic, domain-independent way of appraising events to be useful in a variety of applications.

Incorporation of Personality: Personality plays an important role in determining the intensity of emotions and should therefore be incorporated in a model of emotions.

Incorporation of Motivational States: Motivational states such as pain and hunger should be included because they also influence the emotion system.

Adaptability: The model should be able to learn to reflect the dynamic status of the environment.

2.2 Existing Computational Models of Emotions

Many models of emotions have been proposed for various purposes with differences in focus. For a survey, see [4]. We will now study how models, that inspire ParleE, suit our requirements.

2.3 Cathexis Model

Cathexis is proposed by Velásquez [11]. By considering both cognitive and non-cognitive elicitors of emotions such as sensorimotor and motivational states, Cathexis models six basic emotions: anger, fear, distress/sadness, enjoyment/happiness, disgust and surprise. Cathexis also differentiates emotions from other kinds of affective phenomena, such as mood. The intensity of emotion e at time t is calculated based on the intensity of emotion e at time $t-1$ and the values of emotion e's elicitors. The value of an elicitor of emotion e is derived from the value and weight of conditions that contribute to the activation of the elicitor. This model is limited because these values and weights have to be predefined. Therefore, it is not a flexible model. Moreover, the system is not adaptive and there is no integration of personality.

2.4 Elliott's Affective Reasoner

Elliott's Affective Reasoner [3] is a computational adaptation of the OCC model [9]. This model assesses the relationship between events and an agent's disposition (described by its goals, social standards, and preferences). The relationship is characterized in terms of a set of features called *emotion-eliciting conditions*. Elliott's Affective Reasoner is not a quantitative one as it does not consider the intensities of emotions. Besides, there is no adaptation, integration of personality and motivational states in this model. This model is also limited by the use of domain-specific rules to appraise events.

2.5 FLAME

El-Nasr et al's FLAME model [4] is a computational model of emotions based on event appraisal. It incorporates some learning components to increase the adaptation in modelling emotions. It also uses an emotion filtering component, which takes into account motivational states, to resolve conflicting emotions. FLAME uses fuzzy logic rules to map assessments of the impact of events on goals into emotional intensities. However, the agent in FLAME does not have clearly defined goals. Moreover, the model does not provide a way of calculating the impact of an event on an agent's goal. Instead, it uses a predefined reward value for the user's action's impact on an agent's goal. Hence, this is an inflexible model. Personality is also not mentioned in this model.

2.6 Gratch's Émile

Gratch's Émile [6] uses classical planning methods (detecting and resolving threats) to appraise the emotional significance of events as they relate to plans and goals, to model and predict the emotional state of other agents, and to alter behavior accordingly. In this model, Gratch has proposed a simple way of calculating the probability of a goal to be achieved, the probability of threats to a goal and its importance. Standards are defined as domain-specific constraints of behavior such as "thou shalt not kill". The limitation of this model is that it leaves out the value of an event's unexpectedness when calculating the intensity of event-based emotions like joy and distress. Moreover, Émile's threats detection approach would mistreat the event that is both establisher and threat to the agent's goal. For example, for an agent having a goal of watching TV, the preconditions (or subgoals in Émile) are the TV is in the living room and the TV is not broken. However, the TV currently is broken although it is in the living room. An available plan to achieve the top-level goal are "bring the TV to the shop to have it fixed", and then "bring the TV back". In Émile, the action of "bring the TV to the shop" is to satisfy the second condition/subgoal that the "the TV is not broken", but is considered as a threat to the first condition/subgoal that "the TV is in the living room". Émile would generate sadness for the event "the TV is in the shop", which is not logically sensible. Our approach, which uses a probabilistic planning algorithm with heuristic searching, does not encounter this problem. We do not appraise an event by considering it as an establisher or a threat to the agent's goal but by assessing how that event changes the probability of achieving the agent's goal.

Émile also does not pay attention to the way motivational states and personality influence emotion.

3 The ParleE Model

In this section, we will give an overview of ParleE. Its components will then be discussed in more detail.

3.1 Overview

When an event happens, an Emotion Impulse Vector (EIV) is generated by appraising the event using the rules proposed by the OCC appraisal theory [9] based on the agent's goals, plans and standards. An EIV contains the values of the event's impact on emotions. The EIV is then used to update the Emotion State Vector (ESV), that contains values representing intensities of emotion. This will be discussed in Sect. 4.4. An overview of the system can be seen in Fig. 1.

The Emotion Appraisal Component takes the event, personality, plan, and models of other agents as inputs to produce the EIV as output. This component will be explained in Sect. 3.2.

The Planner produces a plan to achieve the agent's goal. It also calculates the probability of achieving the goal. This probability is then used by the Emotion Appraisal Component to calculate the EIV. The planner is discussed in Sect. 3.3.

The Emotion Component takes the EIV and motivational states as inputs and produces the updated emotion vector as the output. This component also cooperates with the Emotion Decay component to produce decayed emotions. This component is described in Sect. 3.4.

The Emotion Decay Component calculates how emotions decay taking into account the personality. The decay function is discussed in Sect. 4.3.

Models of other agents are used to generate **Desire-other** emotions (emotions about the fortunes of others) and to predict other agents' behavior. They will be explained in Sect. 3.6.

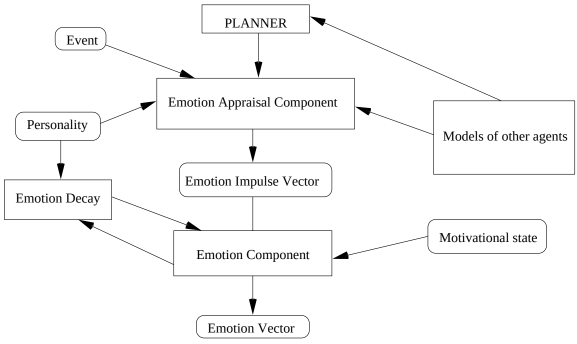

Fig. 1. Overview of the system

3.2 Emotion Appraisal Component

The Emotion Appraisal Component is in charge of generating emotions when an event occurs. This is done based on Ortony et al.'s appraisal theory [9]. Emotions are generated by assessing events relating to goals, expectations and standards.

ParleE appraises events based on learning and a probabilistic planning algorithm. For relating the events to standards, we also use a learning component. For assessing the significance of an event to the agent's goal, we make use of a probabilistic planning algorithm. In a probabilistic planning algorithm, the aim is to find a plan with the optimal probability of achieving the goal. We take the difference between the probability of achieving the agent's goal before and after the event happens as the effect of the event on the goal.

3.3 The Planner

We extend Blum and Langford's Probabilistic GraphPlan algorithm [1] to support planning in a multi-agent environment. Probabilistic GraphPlan uses graph structure to solve STRIPS-style planning problems. It allows the calculation of a plan's probability of success from a specific state of the world. We then take the change in a goal's probability of success when an event happens as the impact of the event on the goal.

In STRIPS domains [5], the state of the world is a set of true facts. A *goal* is a set of facts that we want to be true. An *operator* represents a legal action that may be performed. An operator has conjunctive preconditions, a list of facts to be added and a list of facts to be deleted. Blum and Langford extend STRIPS to have probabilistic actions. Each operator then has conjunctive preconditions and a set of possible outcomes. Each outcome is defined as a set of add and delete effects, each set having an associated probability. For example, a cook action might require that there is food, and with 80% probability of having a good meal, and 20% probability of having a bad meal.

To support planning in a multi-agent environment, we incorporate models of other agents (see Sect. 3.6) to predict other agents' behavior. We denote the probability for an agent a to achieve his/her goal at time t and the state of the world s as $P_{goal}(a, t|s)$. Provided it is agent b's turn to act, $P_{goal}(a, t|s)$ is calculated as:

$$P_{goal}(a, t|s) = \sum_{i=1,n} P(b, action_i, t|s) \times P_{goal}(a, t|s, action_i)$$

where $P(b, action_i, t|s)$ is the probability of an agent b to perform an action i at time t and the state of the world s, $P_{goal}(a, t|s, action_i)$ is the probability for an agent a to achieve his/her goal at time t and the state of the world s provided action i is performed. $P_{goal}(a, t|s, action_i)$ is calculated as:

$$P_{goal}(a, t|s, action_i) = \sum_{s' \in succ(s)} P(s'|action_i, s) \times P_{goal}(a, t+1|s')$$

where s' is a possible state of the world with probability of $P(s'|action_i, s)$ if action i is performed at the state of the world s.

We use probabilistic approach to select an agent's action to avoid local minima. For an agent a, an action i is selected with probability:

$$P(a, action_i, t|s) = \begin{cases} \frac{P_{goal}(a,t|s,action_i)}{\wp} & \text{if } \wp > 0 \\ \frac{1}{n} & \text{if } \wp = 0 \end{cases}$$
$$\wp = \sum_{j=1,n} P_{goal}(a, t|s, action_j)$$

We also extend this planning algorithm to support multiple goals. Each goal is then associated with a priority value (from 0.0 to 1.0). The goal with the highest priority will be planned first.

3.4 The Emotion Component

The Emotion Component contains a representation of emotions and is responsible for updating the emotional state. Emotions are represented as a vector of intensities for every emotion type. Each emotion is associated with two thresholds, which will be described below.

As in Cathexis [11], we distinguish between moods and emotions by the level of arousal. We adopt Cathexis's concept of two thresholds associated with each emotion type. The first threshold, denoted as α, controls the activation of the emotion type and differentiates between mood and emotion. Emotion types with intensity level lower than their α are considered as moods, and do not have as much influence as emotions on an agent's behavior. The second threshold, denoted as ω, controls the saturation of the emotion type (the upper limit for intensity of the emotion type).

3.5 Motivational State

As in [4], we consider hunger, fatigue, thirst, and pain as motivational states. In ParleE, motivational states influence emotions by changing the α threshold for each emotion. When the level of *fatigue* gets higher, the α thresholds for negative emotions decrease (when the agent is tired, his/her negative emotions seem to be easily activated) and the α thresholds for positive emotions increase. High levels of pain tend to decrease the α threshold for emotion fear, while high levels of hunger and thirst tend to increase the α thresholds for all emotions.

3.6 Models of Other Agents

In our system, the agent has also models of other agents' goals and plans (we consider the user as an agent as well) in order to predict other agents' behavior and in order to appraise **Desire-other** emotions (emotions about the fortunes of others). The goals of other agents can be known through communication. Based on that information, the agent can predict other agents' plans. The probability that another agent performs an action is derived from the predicted plan. To appraise emotions about another agent's fortunes, the desirability of an event to another agent's goals is derived from the predicted plan.

4 Intensity of Emotions

To compute the intensities of emotions, Ortony et al. [9] have proposed several intensity variables including *desirability*, *praiseworthiness*, *appealingness*, and *unexpectedness*. Based on that, we propose a quantitative model that uses six variables: *impact of an event on goal* and *goal importance* (to compute the *desirability*), *probability of achieving a goal* and *probability of an event occurring* (these two probabilities are equivalent to the *unexpectedness*), *praiseworthiness* (value of an action), and *liking* (between agents). These variables are used to calculate the EIV rather than the emotion vector. For each emotion type, we first calculate the value of the impulse with regard to each goal. The sum of all impulses for all goals is taken as the final impulse for that emotion type. The impulse vector is then used to update the emotion vector.

We will now discuss these emotional variables, followed by the EIV, the decay function and the integration of the EIV into the emotion vector.

4.1 Emotion Variables

In this section, we describe how to calculate the value of emotion variables. These variables are used by the Emotion Appraisal Component to calculate the EIV, which will be combined with the current emotion vector to generate a new emotion vector.

Goal importance is an important factor for determining the intensity of goal-related emotions. It is denoted as $Import(goal)$. In ParleE, we only consider top-level goals rather than any subgoals that arise in the plans developed to achieve top-level goals. Thus, if an event happens that affects the goal but does not make the goal succeed or fail, we consider that it partially affects the goal. Because of this approach, we are only concerned with the intrinsic importance of the goal (as defined in [6] as “the reward (utility) an agent receives from achieving the goal"). We do not consider a goal’s extrinsic worth (how a goal promotes other goals) as we do not have subgoals in ParleE. For the intrinsic importance of the goal, we assign predefined values as in [6].

The probability of achieving a goal, denoted as $P(goal)$, is adopted from Blum and Langford’s Probabilistic GraphPlan algorithm. This algorithm computes the optimal probability of a goal success from a state of the world. We take that value as the probability of achieving the goal. The advantage of a plan based emotion model over other approaches is that the planning algorithm allows a generic (domain-independent) way of calculating this probability.

The probability of an event occurring, which is denoted as $P(event)$, is defined in ParleE as follows:

- If the agent is waiting for the result of his/her own action then $P(event)$ is the probability of an outcome of an action (c.f. Sect. 3.3). This probability is first assigned some predefined value. It is then updated by learning from what actually happens, which will be discussed in Sect. 6.

- If the agent is waiting for another agent's action then $P(event)$ is the probability of another agent's action at the current world state s times the probability of an outcome of that action. The probability of another agent's action is calculated from this agent's model of another agent (c.f. Sect. 3.3 and 3.6).

The impact of an event on a goal in this model is the difference between the probability of achieving the goal before and after the event happens (denoted as *Impact(event,goal)*). Thus, if the value of impact is negative, that means the probability decreases after the event happens, the event is undesirable for this goal; if the value of impact is positive, that means the probability decrease after the event happens, the event is desirable for this goal.

$$\textit{Impact(event,goal)} = P_{\textit{after}}(\textit{goal}) - P_{\textit{before}}(\textit{goal})$$

Praiseworthiness is used to evaluate how the agent's action meets standards of behaviors. We use the idea of FLAME [4] about learning values of actions to form the standards in appraising emotions. The variable is denoted as *Praiseworthiness(action)*. Another agent can provide feedback on the agent's action. In ParleE, the feedback is a point which ranges from -1.0 to 1.0. The average feedback point for an action over the time is taken as the value of *Praiseworthiness(action)*.

Liking, denoted as $\textit{LikingLevel}(\textit{another agent})$, is a variable that contributes to the intensity of **Desire-other** emotions. In ParleE, the agent maintains a dynamic liking level towards each another agent. Each liking level ranges from -1.0 to 1.0. Starting with a neutral attitude toward another agent (the value of liking level is 0.0), the agent's liking level changes when another agent has performed an action that affected the agent's goal. The agent's liking level increases if it is a desirable event and decreases if it is undesirable. The agent's liking level towards another agent is updated as follow if the preceding event is caused by another agent:

$$\textit{Desirability(event)} = \sum_{\textit{all goal}} \textit{Import(goal)} \times \textit{Impact(event,goal)}$$
$$\textit{LikingLevel}_{t+1} = max(-1.0, min(1.0, \textit{LikingLevel}_t + 0.1 * \textit{Desirability(event)}))$$

4.2 The Emotion Impulse Vector

We now present formulas to calculate the intensity of some emotion impulses. For other emotion impulses, it works the same way based on the value of all variables described above.

Hope arises from the belief that a goal is going to succeed and it is important that the goal does not fail. So **Hope** arises only when the probability of the goal to succeed is higher than 0.5. On the contrary, **Fear** arises from the belief that a goal is going to fail (probability of achieving the goal is lower than 0.5). Therefore, the probabilities of the goals to succeed directly relate to the intensity of **Hope** and **Fear**. In addition, the probability of achieving a more important goal seems to have more influence on the intensity of **Hope** and **Fear**. To capture those characteristics of **Hope** and **Fear**, we propose the following formulas:

$$\textit{Hope} = \sum_{\textit{all goal}} \textit{Import(goal)} \times (P(\textit{goal}) - 0.5)$$

$Fear = \sum_{all\ goal} Import(goal) \times (0.5 - P(goal))$

Happiness arises when a goal succeeds or becomes more likely to succeed (the value of the impact from the event on the goal is positive). **Sadness** arises when the goal is more likely to fail (the value of the impact from the event on the goal is negative). The value of the impact of an event and the importance of the goal are the two factors contributing to the intensity of **Happiness** and **Sadness**. Moreover, the unexpectedness of the event also plays a role in determining the intensity of the two emotions, and it is likely to have more effect on **Happiness** than **Sadness**. The intensities of **Happiness** and **Sadness** impulses are modelled as follows:

$Happiness = \sum_{all\ goal} Impact(event,goal) \times Import(goal) \times \sqrt{1 - P(event)}$

$Sadness = \sum_{all\ goal} Impact(event,goal) \times Import(goal) * (1 - P(event))^2$

Anger arises when another agent performs some action that is undesirable for the agent. The intensity of **Anger** is also calculated using the value of the impact of the event , the importance of the goal and the probability of the event to happen:

$Anger = \sum_{all\ goal} Import(goal) \times \sqrt{1 - P(event)} \times Impact(event,goal)$

Surprise arises when some unexpected event happens. In ParleE, **Surprise** arises only when $P(event) < 0.5$. The intensity of the **Surprise** impulse is considered as the unexpectedness of the event:

$Surprise = 0.5 - P(event)$

The **Happy_for** emotion arises when an agent that is liked is happy (when the value of liking level is positive). We consider this emotion as an example of **Desire-other** emotions. The intensity of **Happy_for** is determined by the agent's liking level towards another agent and the intensity of another agent's **Happiness** (derived from the models of other agents):

$Happy\_for = LikingLevel(another\ agent) \times (another\ agent's\ Happiness)$

Pride and **Shame** are two emotions related to standard of behaviors. The value of these depends on the praiseworthiness of the performed action (if the praiseworthiness is positive then **Pride** arises, if it is negative then **Shame** arises):

$Pride = Praiseworthiness(action)$

$Shame = -Praiseworthiness(action)$

4.3 Decay Function

It is important to derive a reasonable decay function of emotions over time. It seems that emotions decay slower when their intensity levels are lower. With the definition of mood above, it is reasonable that moods persist much longer than high intensity emotions. Moreover, negative emotions tend to decay slower than positive emotions. Personality also influences how emotions decay. Taking into account these considerations, we propose a decay function as follows:

$\Psi(E_{i,t+1}) = \Phi(E_i, personality) \times E_i(t)$

where $E_i(t)$ is the intensity of an emotion E at time t, $\Phi(E, personality)$ is a function of emotions that generates different decay rates for each emotion with regard to the agent's personality. In ParleE, the *Inclination Level* of the agent's *feeling emotions* (see table 1 in Sect. 5) will affect the function $\Phi(E, personality)$. The higher the *Inclination Level* of the agent's *feeling emotions* is, the slower the emotions decay. The decay rate for an emotion E_i is modified as follows:

$$\Phi(E_i, personality) = decayFactor(E_i) * feelingLevel$$

4.4 Updating Emotions

To update emotions over time, one has to consider the previous emotion states, the decay function, and the values of emotion impulses.

The intensity of emotion i is the decayed value of the intensity at previous time plus the reduced value of emotion impulse i. As the value of emotion impulse i is affected by the intensities of emotions at previous time, it is proportionally reduced by the sum of effect values from all emotions to emotion i. Finally, the intensity of emotion i is limited to the upper threshold ω_i by the function min:

$$E_{i,t+1} = min(\omega_i, \Psi(E_{i,t}) + EI_i \times (1 - \sum_j \frac{S_{j,t}}{\omega_j} \times M_{j,i}))$$

$$S_{j,t} = \begin{cases} 0 & \text{if } E_{j,t} < \alpha_j \\ E_{j,t} & \text{if } E_{j,t} \geq \alpha_j \end{cases}$$

where $E_{i,t}$ is the intensity of emotion i at time t, ω_i is the upper threshold for the intensity of emotion i , $\Psi(E_{i,t})$ is the decay function as described above, EI_i is the intensity of emotion impulse for emotion i, $M_{j,i}$ is the effect factor from emotion j to emotion i. To assure that the effect of an impulse on an emotions has as minimum 0 and as maximum the intensity of the impulse itself, we introduce the following constraint on $M_{j,i}$:

$$0 \leq \sum_j M_{j,i} \leq 1 \quad \forall i$$

Since $0 \leq E_{j,t} \leq \omega_j \quad \forall j$

$$\Longrightarrow \quad 0 \leq S_{j,t} \leq \omega_j \quad \forall j$$

$$\Longrightarrow \quad 0 \leq \frac{S_{j,t}}{\omega_j} \leq 1 \quad \forall j$$

$$\Longrightarrow \quad 0 \leq \frac{S_{j,t}}{\omega_j} \times M_{j,i} \leq M_{j,i} \quad \forall j$$

$$\Longrightarrow \quad 0 \leq \sum_j \frac{S_{j,t}}{\omega_j} \times M_{j,i} \leq 1$$

5 Model of Personality

5.1 Rousseau's Model

We use Rousseau's model of personality [10] as it presents a clear view on the relevant dimensions of a personality. It also provides "a sufficiently rich structure based on convention architecture of an intelligent agent" [10]. Personalities are classified according to different processes that an agent can perform: perceiving, reasoning, learning, deciding, acting, interacting, revealing, and feeling emotions. Each process is considered at two levels: the natural inclination that the agent has to perform the process, and the main aspect that the agent focuses on while performing the process. A summary of the dimensions of a personality can be seen in table 1. We found this model convenient and easy to implement and to assess the influence of personality on other processes (eg. emotion). We also extended this qualitative model to a quantitative model. A personality is now represented as a vector in 16-dimensional space:

Personality $= (p_1, p_2, ...p_{16})$ where $0.0 \leq p_i \leq 1.0$

We now illustrate how this vector works by explaining some of its components. The first component of the personality vector represents the *Inclination Level* of *Perceiving* (we symbolize it as *perceivingLevel* for convenience). The lower the value of this component is, the more absentminded the agent. The higher the value of this component is, the more alert the agent. The second component represents the *Focus Aspect* of *Perceiving* (we symbolize it as *perceivingFocus*). When the value of this component is low, the agent focuses more on expectation (the agent is more imaginative). The agent focuses more on reality (the agent is more realistic) when the value of this component is high. Other components of the personality vector are interpreted and symbolized in the same way.

The values of some components of the personality vector are used to modify emotion intensities by influencing the values of the emotion variables. Several components of the personality vector are used to decide the learning rate and how the agent displays emotions.

Table 1. Dimensions of a personality (from [10])

Process	Inclination Level	Illustrated word(s)	Focus Aspect	Illustrated word(s)
Perceiving	Low High	Absentminded Alert	Expectations Reality	Imaginative Realistic
Reasoning	Low High	Silly Rational	Undesirable effects Desirable effects	Pessimistic Optimistic
Learning	Low High	Incurious Curious	What is learned only What is known only	Gullible Intolerant
Deciding	Low High	Insecure Self-confident	First reaction Good decision	Impulsive Thoughtful
Acting	Low High	Passive Zealous	Anything besides the task Result of the task	Indifferent Perfectionist
Interacting	Low High	Introverted Extroverted	Addressee as a threat Addressee as a help	Hostile Friendly
Revealing	Low High	Secretive Open	Lie Truth	Dishonest Honest
Feeling emotions	Low High	Emotionless Sensitive	Self Others	Selfish Unselfish

5.2 How Personality Affects Emotion in This Model

We now present formulas that show how personality influences the way an agent experiences emotion.

If the agent focuses more on expectations when perceiving (the value of *perceivingFocus* is close to 0.0), then the value of the expectation variables ($P(goal)$ and $P(event)$) have higher influences on emotion intensities:

$$newExpectation = Expectation^{(perceivingFocus+0.5)}$$

If the agent focuses more on undesirable effects (the value of *reasoningFocus* is close to 0.0), the impact variable when negative will have more influence on the intensities of negative emotions. If the agent focuses more on desirable effects, then the impact variable when positive will have more influence on the intensities of positive emotions:

$$newImpact = \begin{cases} -\mid impact \mid^{(reasoningFocus+0.5)} & \text{if} \quad impact < 0 \\ impact^{(1.5-reasoningFocus)} & \text{if} \quad impact \geq 0 \end{cases}$$

The agent with a lower inclination level of feeling emotions (less sensitive) will have lower intensity of emotions for the same event compared to the agent with a higher inclination level of feeling emotions (more sensitive):

$$newImpulse = impulse^{(1.5-feelingLevel)}$$

The agent that focuses more on others when feelings emotions (the value of *feelingFocus* is closed to 0.0) will have higher intensities for **Desire-other** emotions than the agent who focuses more on self when feelings emotions:

$$new\ Desire\text{-}other\ impulse = Desire\text{-}other\ impulse \times feelingFocus$$

The value of *revealingLevel* is used in the emotion displaying component and the value of *learningFocus* influences the learning rate.

6 Learning Components

There are two learning components in the system to make the agent more adaptive. The agent gradually learns the probability of an action's outcome and the values of actions (standard of behavior).

The first component is used to learn the probability of outcomes of an action. Initially, we assign some pre-determined values for the probability of each outcome. We also assign a presumed value for the number of previous observations. This number of previous observations is determined by the learning rate. Thus, the higher the value of this number of counts, the longer it takes to learn new probabilities. We then update these probabilities each time the action is performed through new observation of the actual outcome. Suppose probabilities of outcomes of an action are P_1,P_2,...,P_m. The current number of observations is n. The actual outcome of the action this time is i. Then the probabilities are updated as follows:

$$new\ P_j = \begin{cases} \frac{P_j \times n+1}{n+1} & \text{if} \quad j = i \\ \frac{P_j \times n}{n+1} & \text{if} \quad j \neq i \end{cases}$$

$$new\ n = n + 1$$

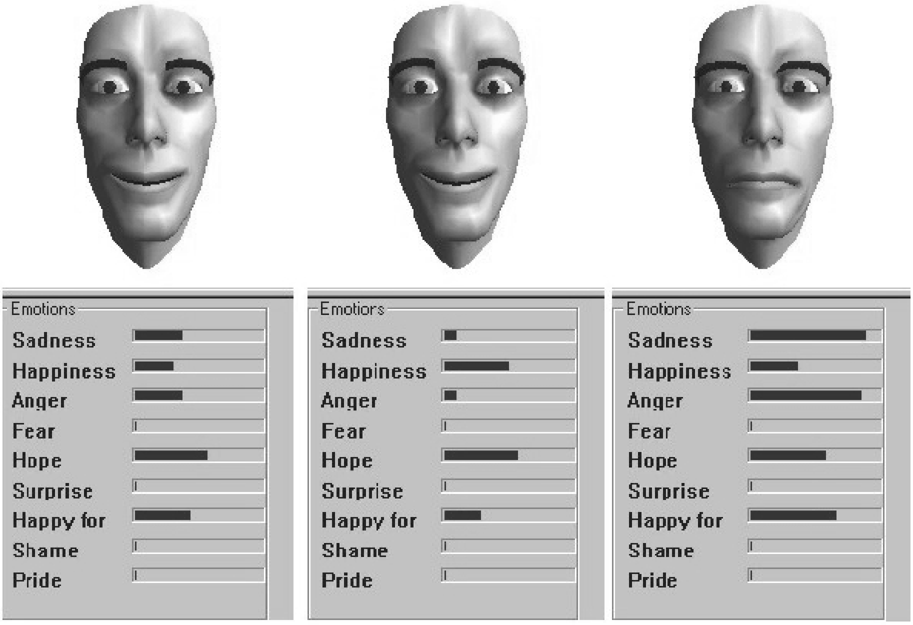

Fig. 2. Neutral Obie (left) expresses facial expressions compared to optimistic Obie (middle) and sensitive Obie (right) after the user eats his bread

The second component is to learn values of actions. For each action that the agent performed, the user can choose to feedback with a point from -1.0 (very bad feedback) to 1.0 (very good feedback). The average point over time is used as the value of the action.

7 Illustration

We have implemented ParleE into Obie, a conversational agent to illustrate the model (see Fig. 2). Obie has been developed with a mixture of text-based and graphical environment to provide interactions with humans. The graphical environment provides a representation of Obie’s 3D face that displays emotional facial expressions. We have used OpenGL to implement this 3D face [2]. Obie communicates with the user through text-based interactions.

We have picked a simple domain for this illustration. Obie and the user live in a house and share a car. Whenever Obie is hungry, a goal of feeding himself with the food is initiated. Several actions are available for Obie to achieve the goal: “go shopping", “buy food" and “unload the food from the car". The user can help Obie or can perform some other actions that may prevent Obie from achieving his goal, such as “drive the car to work". Obie knows the user’s goals by reading them from a data file (this process should be done via the dialogue). A piece of dialogue between Obie and the user looks like this:

```
FACT: (in-car bread car1) (at-home car1)
    Choose an action: (-1 to exit)
        0: No Action
        1: (unload-food bread car1)
        2: (go-shopping car1)
        3: (go-working car1)
    You choose: 0
Agent did: (unload-food bread car1)
    case 1 (with probability of 100 %) happened; effects: (in-fridge bread)
FACT: (at-home car1) (in-fridge bread)
```

During the interactions, Obie’s emotions are displayed in the 3D face (see Fig. 2). This shows how Obie is capable of expressing his emotions in response to an event.

Figure 2 shows three Obies with different personalities: a neutral, an optimistic and a sensitive one. The scenario was as follows: “Obie goes to the shop and buys bread. He brings the bread home. The user eats his bread." The neutral Obie gets angry after the user eats his bread. As the optimistic Obie tends to ignore this negative event, he is still happy with what have happened before. The sensitive Obie gets very angry with the user. This shows that Obies with different personalities respond differently to an event.

Although we focus on using ParleE for an agent’s emotional expressions, ParleE can also be used to alter the agent’s behavior. Moreover, with a generic way of appraising events, ParleE can be used in other domains for various application areas.

8 Conclusion

In this paper we have described ParleE, an emotion model for a conversational agent. ParleE enables the conversational agent to respond to the environment and the user with emotional expressions. It is a quantitative, flexible and adaptive model of emotions in which appraising events is based on learning and a probabilistic planning algorithm. ParleE also models personality and motivational states and their roles in determining the way the agent experiences emotion. In the future we will extend the model and test it out for different situations.

References

1. Blum,A. L. and J. C. Langford (1998). Probabilistic planning in the graphplan framework. In *AIPS98 Workshop on Planning as Combinatorial Search*, pages 8–12, June 1998.
2. Bui,T.D., Heylen,D., Poel, M. and A., Nijholt(2001). Generation of facial expressions from emotion using a fuzzy rule based system. In: Proceedings 14th *Australian Joint Conference on Artificial Intelligence (AI 2001)*, December 2001, Adelaide, Australia, Lecture Notes in Artificial Intelligence 2256, M. Stumptner, D. Corbett & M. Brooks (eds.), Springer, Berlin, ISBN 3-540-42960-3, 83–94.

3. Elliott, C. (1992). *The Affective Reasoner: A Process model of emotions in a multi-agent system*. Institute for the Learning Sciences, Evanston, IL: Northwestern University, Ph.D. Thesis.
4. El-Nasr, M. S., Ioerger, T. and J. Yen (1999). FLAME: Fuzzy Logic Adaptive Model of Emotions, *Autonmous Agents and Multi-Agent Systems*, Kluwer Academic Publishers.
5. Fikes, R.E. & N.J. Nilsson (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2(3–4).
6. Gratch, J. (2000). Émile: Marshalling Passions in Training and Education. In *Proceedings of the 4th International Conference on Autonomous Agents*, Barcelona, Spain, June 2000.
7. Nijholt, A., M. van den Berk and A. van Hessen (1998), A natural language web-based dialogue system with a talking face, *Proceedings Text, Speech and Dialogue*. Sojka et al (eds.), Brno, Czech republic, pp. 415–420.
8. Nijholt, A., D. Heylen and R. Vertegaal (2000), Inhabited interfaces: Attentive conversational agents that help. In: Proceedings 3rd international Conference on Disability, Virtual Reality and Associated Technologies – CDVRAT2000, Alghero, Sardinia.
9. Ortony, A., G.L. Clore, and A. Collins (1988), *The cognitive structure of emotions*, Cambridge, Cambridge University Press.
10. Rousseau, D. (1996). Personality in Computer Characters.In: *Proceedings of the 1996 AAAI Workshop on Entertainment and AI / A-Life*, AAAI Press, Portland, Oregon, August 1996, pp. 38–43.
11. Velásquez, J. (1997). Modeling emotions and other motivations in synthetic agents. In: *Proc. AAAI Conf. 1997*, Providence, RI, pp. 10–15.