Imperial College London – MSc Advanced Computing
Individual Study Option – 2004-2005, Term 2

AFFECTIVE COMPUTING, SOFTWARE AGENTS AND ONLINE COMMUNITIES

Guillaume Piolle
Supervisors: Pr Keith Clark, Dr Jeremy Pitt

Abstract: Transposition of human emotions and feelings in the context of software programs (multi-agent systems for instance) has been used in order to improve performances, to help human-computer interactive programs adapt themselves to the user, or to fight disruptive user behaviour caused by lack of conventional social cues. This ISO will examine the use of affective computing (in particular the representation of trust, reputation and forgiveness) in conjunction with software agents to assess the potential forgiveness for ensuring social order in online communities.

1. Introduction

In the field of Human-Computer Interactions, researchers try to improve their interface models by giving their entities more "human" appearance, but also reactive behaviours and internal mechanisms. Such intelligent and cognitive interfaces would be able to interpret the user's emotions and feelings, to develop their own ones, and to show them to their environment. The main advantage is that it improves the quality and the natural character of the communication, but trying to model and simulate human socio-psychological mechanisms can also lead to new emergent behaviours in software agent communities. For instance, there are nowadays many theoretical models and implementations for trust management layers in all kinds of multi-agent systems. It appears that even when no human is interfaced with the agents, properly simulating that totally human notion significantly improves the performances of the system. In the same state of mind, research in affective computing try to model human emotions and reactions, in order to build "emotive" cognitive software.

Considering the latter, our study will focus on affective computing in both fields of human-computer interactions and multi-agent systems. Both are often deeply linked and mixed.

In the second part of this study we will present an example of implemented trust management framework, on which we will work later, and present some of the current challenges of online communities. In the third part, we will review the current subfields of affective computing, and briefly present the areas being currently explored. In a fourth part, we will put a stress on the existing models for the forgiveness mechanism, and we will try to see whether it would be possible (and if yes, how) to introduce such a notion in a "classical" trust and reputation management framework. We will also identify the possible future directions of research in forgiveness simulations.

2. Background

a) A classical trust and reputation management framework

One of the more "classical" human behaviours translated in the world of multi-agent systems is the notion of trust and reputation. Basically the phenomenon is simple: we (humans) tend to rely more easily on people we know and we trust, rather than on unknown people, or people we don't trust. Trust management layers in software agent design try to make agents behave the same way, in order to improve the success rate of the actions they undertake. The objective here is to make this artificial trust influence the relationship scheme between agents, in the same way that real trust plays a significant role in the establishment and evolution of personal, social, professional, economical relationships in human societies. Indeed, one of the effects of this trust-based behaviour is to "discourage malicious behaviours and to isolate incompetent agents" [10]: If the agent needs to rely on the society to achieve its goals, and it is almost always the case in the society models we consider here, they need to be well-considered by other agents, and thus to have a good *reputation* (so that they would be well-recommended and often relied upon, and that they requests could be accepted more easily). This need for consideration is often a goal in itself. In many frameworks, the action of trusting is coupled with a risk analysis: for instance the level of trust needed to rely upon a specific agent for a given action is determined by the risk represented by the failure of that action.

This is only a very brief and naïve description of the phenomenon, there are many theoretical models of trust. As a basis for our study on affective computing, we have chosen a

trust and reputation platform build by Brendan Neville and Jeremy Pitt ([10], [11]), which could be considered as a partial implementation of Cristiano Castelfranchi's model for trust management [2].

We won't detail here the mechanisms involved in a trust management framework, but if we were to summarize this one, we could point out the following points:

Trust is a belief of the agent. It is considered with respect to a trustee and a context. It is quantified.

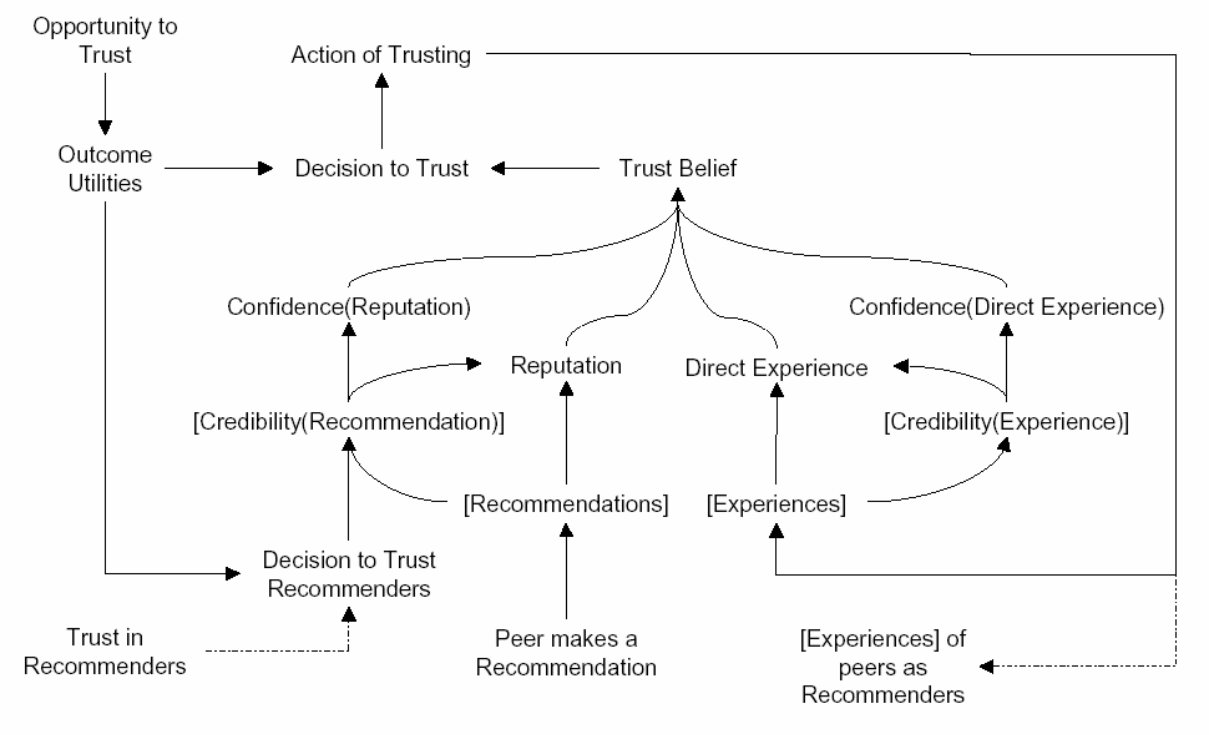
Trust is based on values computed from direct experience (evidence) and reputation (recommendations).

A level of Credibility is associated with each value of recommendation, and experience, and the final making of the trust belief considers the Confidence in direct experience and reputation.

The decision of finally trusting an agent rather than another is based of a quantified computation of utilities values (like in Castelfranchi's model, [2]) associated with the potential success or failure of the undertaken action, for each agent considered. The utility is computed on the basis of the economic data of the market environment.

At each moment of the execution, the agent keeps in memory a few values: the current recommendations (for each trustee, from each peer) with their credibility values, the overall reputation of the trustee with its associated level of confidence, a history the previous experiences with the trustee with their credibility values, the overall experience value with its associated confidence.

Here is a synthetic schema of the trust and reputation mechanism in this framework (quoted from [10]):



This trust management framework has been used with *producer* and *consumer* agents, in order to simulate a simple economic market. In such a scenario, the trust and reputation management layer brings useful information to both kinds of actors about the market, telling them who are the efficient agents, and who are the incompetent or malicious ones. The producers can also compare their own reputation to their competitors', and thus are able to modify their selling strategy if they want to.

The benchmark tests performed with this framework [11], made by introducing one by one the different component of the framework, shows in an obvious way that trust and reputation management, in "helping" agents find the "good" and effective potential partner, makes them behave more efficiently and increases their overall performance. An efficient trust management layer also allows a more realistic evolution in the prices, allowing the market to reach a pseudo-equilibrium, profitable to a wider number of agents (the simulation involves elimination of non-profitable producer agents).

However, one could argue that the system is not perfect: isolated errors from usually efficient agents influence badly their reputation, and can lead to an undue decrease of the overall society performance. But such scenarios are not often simulated. From another point of view, the management of distrust (negative trust) as implemented here could lead (in a more complex environment) to some problems that we will discuss in the next sub-section.

b) Anonymity in online communities

On the Internet, anonymity is considered as a fundamental right for the users. It guarantees his privacy, his safety sometimes. This anonymity can be a major advantage for some applications, but misused, it can be a real curse. One can realise that on the IRC channels for instance, or on mismanaged web hosting services, where people abusively hide themselves behind pseudo-anonymity when performing illegal, immoral actions, or at least actions they would not undertake in real life (illegal document publication, aggressive behaviours…).

The problem is the same in every anonymous community: the lack of commitment from the user allows him to ignore social and moral rules, when it helps them reaching their goals. Since they cannot be identified, they do not care about being bad evaluated by the society. And of course, the undertaken actions often ruin the overall performance, preventing the other agents from accomplishing their planned actions properly, or preventing the whole community from reaching its "moral" goal.

From this point of view, clear identification of the agents, and maybe cryptographic authentication and non-repudiation mechanisms can easily be considered as an improvement in online societies. It would be interesting to have the possibility to compel users to identify themselves, to assume their undertaken actions.

This kind of capability is deeply linked to the notion of personal agents (human-controlled agents), however even in pure software agents identification can help integrating social, moral cues in the virtual society. Indeed, belief-based agents can easily be defined with a built-in goal which would induce agents to perform "good" actions that could help them being well-evaluated by the society. Trust and reputation management is a perfect tool for that! However, if it is not well implemented, it can have undesired side-effects: if an agent is "mistrusted" or "distrusted", if its trust level is lower than the default trust level, it can be induced to change its identity and present itself as a brand new agent in the community, with a new "trust virginity". This drawback of lousy implementations of the notion of distrust should be, according to me, considered jointly with the anonymity issue.

As we will see later, some of the aspects of affective computing, by reinforcing the identification and the links between agents, can help addressing such issues. Shame, embarrassment, blush, forgiveness could be used in that sense.

3. An overview of Affective Computing

a) Modelling emotions

In [6], in the "Literature review" part of his PhD thesis, Petar Goulev presents seven "emotional models", or theories of emotions, previously proposed by other researchers. The common aim is to describe, categorise and model the "human emotional state". Apparently no one is better than the other, they are different points of view of the same complex phenomenon. It is likely that one has to choose and adapt one of these models according to one's own application, research subfield, points of interest.

Jame's model (1890) consider the emotional state of a human being as a continuous variable, composed of four basic elements: *rage, fear, grief and love*. He links deeply the emotional state and the physiologic response. **Ekman's model** (1982) is based on facial expression. Ekman suggests that emotions should be considered as mixed and not "separated" in the emotional state. His "emotional components" are *anger, fear, sadness, enjoyment, disgust and surprise*. **Plutchik's model** (1962, 1980) considers emotions from an evolutionary point of view (psycho-evolutionary theory of emotions). For him, emotions are dependent of the specie evolution (there are animal emotions as well as human emotions, and they have some points in common – basic reactions), they even have a significant role in evolution and adaptation. He defines eight "primary emotions": *anger, fear, anticipation, sadness, joy, acceptance, disgust and surprise*. "Real" emotional states are a mix of these primary emotions. **Panskepp's model** (1982) has a model based on *rage, fear, panic and expectancy*. He focused on the neural (hardware) mechanisms involved in emotions. **Arnold's model** (1960) gives importance to personality and context: emotions are felt and expressed differently by each one. This model is based on *anger, aversion, courage, dejection, desire, despair, fear and hate*. **Izard's model** (1972) focuses on the communicational aspect of emotions, in particular in the case of infants. The model is based on *anger, fear, distress, joy, surprise, inerest, digust, contempt, guilt and shame.*. **Fridja's model** (1987) studies the expression of emotions in the subject's behaviour. In fact for him emotions are part of the subject's behaviour. This model is based on *anger, fear, distress, joy, surprise, aversion, contempt, pride, shame and desire*.

Many researchers in this field consider, like Rosalynd Picard in [14], that "emotions contribute to regulating and guiding attention, and to helping make decisions, generally biasing one's selection of next moves away from negative or harmful choices." They often

think that it is a negligible fact that emotions, by preventing rational behaviour, sometimes lead to problematic or non-optimal behaviours. Everybody knows that one must make a delicate balance between emotional and rational motivation in order to act efficiently, but such prerequisite is often taken for granted by researchers in affective computing.

b) The current challenges in Human-Computer Interaction

In Human-Computer interactions, researchers working with affective computing have basically the two following aims: to make the system express, show an emotion, to make the system understand the user's emotions and adapt itself to them.

Displaying a human emotion can be either a rather simple exercise, or a really complex challenge. Like Rosalynd Picard said in [14], the first Macintosh computers displaying a smile at startup was maybe the first step of affective computing… Things have of course evolved since that, and progress has been made in the field. One must keep in mind that expressing an emotion for an artificial system, is very different from the same system "feeling" that emotion, as we will see later: "affective demonstrations" do not always match with an inside emotional mechanism, just like in the case of the Macintosh smile.

The display of emotions and feelings can be supported by almost any human-computer communication modalities. A stress has been made on 2D and 3D avatars and facial animation, but realism (and human appearance for such avatars) is not a fundamental need for believability. In the case of avatars, much work has been done related to smiles, laugher, facial animation in general. Actually, the more realist the avatar, the more complex the "expression engine" has to be, for a same level of believability. The expression of emotional states can also be done by simple textual communication, or even non-verbal "somatic signals": for instance in Star Wars, R2D2 make the public understand its overall feeling just with non-verbal sounds and coloured lights.

To understand a user's emotion is a much more difficult issue. In [14] Rosalynd Picard describes a system which is able to detect the variations of the emotional states of the user. The sensor system uses mainly skin-surface detectors: electromyogram, measure of skin conductance, blood volume pulse, respiration. In this model, eight discrete categories of user emotions have been defined to describe and categorise a complex emotional state, and the overall final accuracy is about 80%. However, this result has to be tampered: the biological symptoms of emotions vary significantly between two people, and from one day to the other. Such variations are even bigger than emotion-caused variations. Consequently, the learning

phase must be quite long and complex, and the system is more likely to detect variations, rather than "absolute" emotions. The building of a long-term relationship between the user and the system also helps the user to discover and appreciate the system (the principle in itself is a concept in affective computing), and the system to adapt itself to the physiological specificities of the user. But if the results are properly used, a system can adapt its behaviour to the user emotional response, the final objective being the design of adaptative, less frustrating human-computer interfaces, able to understand the affective state of the user rather than using standardised, idealised user models. Such an interface should also be able to infer a link between the user emotional variations, and the software context, and to suggest links between the system's actions and the user's reactions.

Facial recognition is sometime used by such emotional analytic systems in order to interpret the emotional state of the user, but the difficulties are multiple: technical (real-time edge detection technologies) and psycho-physiological (two persons don't express their feeling the same way, by the same facial expression, and variations are much greater than with skin sensors since users have a conscious control of their facial expression).

One must keep in mind that the performances of such systems are limited by our own analysis capabilities: even for a human being it is difficult to recognise the "emotional state" of somebody else, and even for oneself it is always difficult to describe it, to label it, because of the dual discrete-continuous character of emotions, well represented by the variety of emotion models described sooner. Not all emotional states can be analysed and described verbally, so it would be utopian to expect better results from affective computing systems. However even partial results, fuzzy ideas about the variations of the user's emotional state can help to significantly improve the system.

In [7] and [17] another kind of affective system is described: the SenToy. Here the context is the one of computer games, and the SenToy is a way to control a virtual character (a kind of avatar in the virtual universe, in fact). SenToy is a doll full of sensors, linked to the computer, and the user manipulates the character in the game by moving the doll. In the associated game FantasyA, characters are wizards fighting against each other. Actually the user does not communicate movements to his character, but directly an emotion (anger, fear, surprise, gloat, sadness, happiness) and depending on the resulting emotional state of the character, and also the perceived emotions of the opponent, the wizard will adopt a different strategy, will perform a different action (attack or defence movements). Depending on the emotion, the wizard can take risks if he is in an "optimistic" emotional state, for instance, or if

the previous actions lead to good results (unexpected bad results have another influence on the emotional state of the wizard, and thus on his further actions). The personality and the virtual "social background" (clan) of the character also plays a role in his reaction to emotions. Here emotions are viewed, in the role play, as a way to control the game.

The interest of SenToy and FantasyA mainly lies in the fact that it provides an overview on the users' reaction to affective, emotional applications, which may not react exactly as expected but for which emotions should be taken into account. It results that exaggerated emotions are perceived more easily: subtlety seems not to be expected from a computer application! Apparently the users do not like the emotions being a "fuzzy" way to control their application or character, or at least they do not always make the link between the "controlled" emotions and the actions of the character. Maybe it means that some work could be done in the models that lead from emotions to action planning. Another problem is that people don't express emotions and feelings the same way through their own body, and through a doll. Furthermore, expressing something through the doll implies a prior formalisation of the emotions and perhaps that step might lead to non-natural reactions.

In [7] we also discover Agneta and Frida, two characters on the desktop of the user, interacting with the environment and the actions of the user: making comments on the actions, the visited documents, talking to each other. It is considered in the paper as an "affective interface", but apparently the main objective of the application is to build a casual and humorous ambience in order to improve the user's "performance" at work. Actually the users seem to enjoy the company of Agneta and Frida, but they also may be disturbed by their presence.

c) The current challenges in Multi-Agent Systems

In the multi-agent field, researchers add emotions one by one, and sub-component by sub-component, in the cognitive model of the agent. Emotions are often considered as independent logical bricks, which can be activated or not, implemented or not. Envy, fear, shame and embarrassment are the most commonly explored and modelled emotions for multi-agent systems. One could also find in the literature the notions of humor and forgiveness, which are not emotions strictly speaking but which can be integrated in the "emotional layer" of the agent model.

In [3], Cristiano Castelfranchi proposes a detailed description of the "cognitive aspects in emotion", adapted to belief-based agents. For him, and that could be understood quite

easily, pleasant and unpleasant emotions are deeply linked to success or failure of the agent's goals. The emotions are "used" by the agent as a motivation tool, in order to encourage or avoid such or such behaviour, by a more immediate way than the rational one. From that point of view, Castelfranchi divides emotions into positive ones (joy, pride) and negative ones (fear, shame, guilt).

Castelfranchi describes the links between the internal beliefs of the agent, and its emotions: there are *activating beliefs* (often evaluation beliefs), which "trigger" an emotion. For instance, if A's goal is to be rich, and A believes that B is richer than it, then this belief may trigger *envy* in A (towards B). There are *causal attribution beliefs*, which could be considered as more introspective: the agents perceives the somatic expression of the emotion (arousal, sensations), and this causal attribution belief makes it link the expression to the emotion that caused it. An agent implementing such beliefs would probably be very oriented towards emotions, self-analysis, introspection, analysis of emotions. Castelfranchi also proposes *categorisation beliefs*, also in the "introspective" category, which help the agent "labelling" its emotion. It is useful only in the case when the agent implements a continuous model of emotion, in which its emotional state is for instance a point in a multi-dimensional space defined by a base of "primitive emotions".

He also links emotions with goal in the agent model. For him emotions can be *goals in themselves*: I may want, as a personal goal, to feel such or such emotion. Indeed it is actually a very human behaviour to perform actions in order to reach a given emotional state. Of course emotions also are a *monitoring tool* for the agent's goal: he feels positive emotions when it gets closer to its goal, and negative ones when it doesn't manage to, just like a human being can feel happy and motivated, or sad and frustrated. In some cases emotions can also activate or create some "impulsive" goals, which sometimes don't have a rational justification. I would be quite cautious about that, since it is often in such cases when human beings make mistakes because of their emotions…

Let's see more in detail how a few emotions, like envy and shame, can be defined for a belief-based agent. We will focus later on forgiveness.

From [3] we learn that **envy** is triggered (when agent A envies B for the fact of having i) whenever the following conditions are fulfilled:

- A believes that B has i,
- A believes that A has not i,

- A has the goal of having i,

This is the simple envy mechanism, when an agent A experiences resentment towards another agent B because B possesses something A would like to have. It can be classified with the "negative" emotions: this envy can prevent A from helping B, trading with B, trusting B, relying on B… and can persist even after the cause of the emotion disappeared. We could express the latter conditions with a prolog-like syntax, or with logical modalities, or another "more scientific" formalism, but it doesn't bring much in the context of that study.

The last example is based on physical possession, but the cause of envy can be a comparison in the capabilities of the agents (agent B is able to do some action that agent A cannot perform, though A needs to), a comparison of power, influence, social reputation… One should keep in mind that agent A absolutely must have a corresponding "frustrated" goal, and it must believe that he cannot fulfil this goal (maybe because of the envied agent, but not necessarily).

Envy is triggered when the three beliefs previously presented hold simultaneously. Envy generates resentment, frustration, and influences the emotional state of the agent in a "negative way". In the case of human behaviour, this influence can remain when the causes of envy have disappeared. Envy can trigger irrational, counter-productive goals (at the society scale), which would induce A to obtain what B has (and help A reach its original goals). It can be a source of motivation.

In the same paper, Cristiano Castelfranchi also presents a simplified model for shame. Here is a brief summary of this simplification. Shame is a more complex phenomenon than envy, more beliefs are involved in it. In order to feel ashamed, you must:

- believe that your action is "bad",
- believe that some other people know you have done it,
- believe that these people believe that the action is bad,
- want to be well-evaluated by these people.

In [3] Castelfranchi expresses that by the mean of the following expressions (quoted and adapted from his paper), in the case where the agent x is ashamed in front of agent y for having done the action i.

```
(Negative-eval x i)
(Bel x (Bel y (Did x i)))
```

```
(Bel x (Negative-eval y i))
(Goal x (Positive-eval y x))
```

He also adds a consequence of the previous expressions, a belief which is a threat for x's goal to be well-evaluated:

```
(Bel x (Negative-eval y x))
```

One can find more or less the same formalisms in [15], more developed. Jeremy Pitt builds in it a more complex model for shame, embarrassment and "digital blush".

After the emotional state of the agent evolved towards shame, this emotion can lead to blush, which is, ideally, a sign visible from other agents, signifying the ashamed emotional state of the subject, and which cannot be simulated by the agent.

Shame, embarrassment and artificial blush ([3], [15]) have not been designed in order to simulate human behaviour by all means, but to fight the problems caused by anonymity, evoked sooner. With shame, embarrassment and digital blush the aim is to make the agents aware of their responsibilities, by making their actions have public social consequences. Elimination of anonymity is an absolute need in this context.

d) Embodied Conversational Agents and Affective Computing

Embodied Conversational Agents (ECAs) are a specific kind of agents. In [18], Ruttkay, Dormann and Hoot define them as "synthetic characters which can converse with the user (or with other ECAs) by some natural modalities of human-human communication. Classically, they have a multimodal user interface, (often) including a moving face, vocal or written speech facilities and/or non-verbal way of expression, and of course user entry interface (skin sensors and video-assisted speech recognition for the most advanced ones, classical keyboard entry for the other ones). In fact they are the limit and the interface between Human-Computer Interactions and Multi-Agent Systems, for they focus on the user interface of cognitive (and often affective) agents. ECAs are often privileged interlocutors for the users in business-to-customer or educational applications, or they can be a "personal agent", an avatar of the user in a virtual world (graphical chat rooms for instance).

The papers [18] and [9] are oriented towards evaluation and comparisons of these Embodied Conversational Agents, and from them we can extract the most particular characteristics of ECAs:

Embodiment: Attention is often given to the appearance of the ECA, it is often a graphical avatar, human or animal. The whole range of realism and details is possible, from schematised, non-human characters to the most advanced facial expression 3D engines.

Believability: Believability of an Embodied Conversational Agent lies not only on the look and the realism of the face (and not essentially: non-realist faces can be more easily "believable"), but even more in the communicational behaviour, the accuracy of the reactions, the quality of speech recognition and generation, the personalisability, the possibility to establish long-term relationships with the user, the richness and coherence of the emotional model, the expressiveness of the face, the lip synchronisation…

Control and Interactivity: ECA designers always pay much attention to the Human-Computer interaction issues. In particular, the quality of speech recognition (or natural language processing of keyboard entries, depending on the user input modalities) is critical, when present. The ECA can also be able to detect and analyse the emotional state of the user, or at least his/her emotional variations. ECAs can be animated online or off-line, controlled by the user or by an autonomous software agent.

Researchers working with Embodied Conversational Agents put a real stress on facial communication / body language, and thus have great interest in affective computing, which is the only way for ECAs to gain a true believability. Thus ECAs are likely to "interface" software agents with a complex affective engine. In fact this embodiment is a great feature for an emotional agent, as we will see in the next sub-point. To summarise, the interface principle of Embodied Conversational Agents are a very powerful way to express the inner emotional state of an affective agent model.

In [12], Anton Nijholt explores the possibilities of integrating the notion of humour in Embodied Conversational Agents: such agents would be able to tell jokes, to smile, to laugh, to detect humour and laugher from the user. He justifies his interest for the sense of humour by explaining how humour is used in negotiation, goal-oriented and casual communication by humans: humour helps building confidence, intimacy, trust, it helps filling the gaps in the conversation. Different kinds of smiles and laughers are used by humans to express different feelings: sincere or feigned joy, happiness, amusement, mockery, boredom, scepticism…

Obviously, including such capabilities in an Embodied Conversational Agent would significantly increase its believability, inasmuch as it would be considered acting in a more "human" way. However, even though smiles and laughers could be expressed quite well

(many kinds of them have been studied and analysed, categorised with respect to the emotions or feelings they express) by an ECA, detecting humour in a user's behaviour or speech, or deciding whether a situation is humoristic or not, and which reaction would be adapted, is a very difficult challenge, out of reach for the moment. An ECA laughing at the wrong time, or after a flat or out of place joke would totally ruin its own believability.

e) Feeling and simulating

When one have a theoretical model for an emotion (or another human socio-psychological notion), it is (usually) quite easy to implement it and to make an agent or a robot simulate the associated behaviour. However, in some cases one can find out that simulating an emotion, exteriorising the symptoms, sharing "computed" somatic signs is not enough! And one could want his agent to actually "feel" the emotion, not only to simulate it. But what is the difference, and how could anyone make an agent "feel" something? It's a very relevant issue, and the question has been posed often ([3], [13], [14]) but the answers given are not always clear or adapted to computer science.

For instance, in order to understand clearly the problem, let's consider the example of blushing. When you're in front of somebody, and the other person is blushing (provided it is not an excessively common reaction for that person), you have no reason to think that she (yes, let's face if: it's often "she") simulates the blush. Why? Simply because it's impossible for a human to simulate that! It would be as difficult as changing one's own heart beat. On the other hand, when a software agent shows to another agent the signs of a "digital blush", it is possible that the causes for this blush (however they re defined in the model) are not present in the agent's execution context, and that the blush has been simulated, being part of a strategy (for instance, if the agent wants indulgence from its interlocutor, whereas the "bad" action perform was premeditated, and not regretted at all). In that case the other agent should be very cautious about its interpretation of the digital blush it perceives… Why is there such a difference? It is a paradigm problem. In the inter-human communication paradigm, we know that the biological, vascular mechanisms which lead to blush are complex, and not controllable at will. In the software agent paradigm (not really defined, though, and not unique), a software program, like an agent, has possibly the expression power of a Turing machine, so why couldn't it simulate any pair name/value it wants? Since an agent doesn't know the source code of another agent, it should consider any "artificial" simulation possible.

Since the software agent paradigm is open, how is it possible to solve that problem? I see three families of solutions here.

The first one is a purely software solution. One could impose in one's "affective agent" environment that the agents identify themselves in a way which would guarantee a part of their behaviour. Such a solution could make use of cryptographic certificates, or proof-carrying code. The need here is to guarantee the execution model of the "affective" layer of the agent: every agent should be able to check that any other agent's affective layer is a black box, whose execution cannot be controlled by the agent (out of specified cases), which has exclusive access to the "somatic signals" defined (like digital blush), and thus which cannot allow digital emotions (and all parts of that layer) to be used as a part of the agent's strategy. This kind of solution could be very technical, demanding, and could involve a non negligible execution overhead, and thus performance losses.

The second possibility would not be a constraint on the environment but directly on the agents' code (the "interpreting", perceiving agents). If you manage to build totally strategy-proof affective mechanisms for your agents, then you don't care whether the other one is simulating or not: the principle is, it is not advantageous for it to simulate (in the bad sense). For instance in the case of blush, we could ignore the blushing reaction of agents which spend their time changing their digital colour… and adapt our own reaction to blush, considering that blush is a rare, occasional reaction… In that case blushing too often (with the hope of being forgiven often for instance) doesn't bring anything but a credibility loss. It is a more elegant solution than the latter, and moreover it is a translation of the human reaction: if someone "average" is blushing, you may think that you've embarrassed him/her, and maybe you'll do some action because of that conclusion. But if you know that this person blushes very easily, you pay less attention. This idea of strategy-proof affective mechanisms for agents is still a bit fuzzy here, and needs to be paid more time and attention. I think it has to be discussed separately for each part of the "affective layer" one would like to build.

The last possibility is also very close to the human paradigm, but impossible to adapt to purely software agents: it lies in an idea discussed by Cristiano Castelfranchi and Rosalynd Picard in [3] and [14]: *an emotion can be felt only with a physical body*. In [3], Castelfranchi justify this assertion by self-consciousness: I can feel emotions only if I am conscious of my body, if I have self-perceptions and internal reactions. It is important, but not enough I think. In [14], Picard criticises this assertion, but at the human interface level, saying that a machine doesn't need a humanoid appearance in order to show emotions. What I want to say here is

closer to Castelfranchi's point of view: if I want my agent to "feel" an emotion, or anything, I need self-consciousness. If I want other agents to be able to trust this feeling (i.e. I believe that this agents is actually blushing and not simulating), the agent (robot) must have a body *that he cannot totally control*. That way, it is possible for the agent to show somatic signs of an emotion (blush) without being able to simulate it. It is a redefinition of the agent paradigm, in fact. In another part of [14], Picard evokes a robot used at the MIT, in which they have implemented "fear" reaction, in order to preserve itself from damages in case of misuse. This could have been implemented from that point of view: such a robot could have fear reaction (physical, hardware, reflex reactions) that its central/conscious/cognitive/rational intelligence/program cannot control. It is another way to implement the "closed affective layer" of the first solution, by defining a new software/hardware paradigm, in which one can trust some of another agent's affective reactions.

4. Forgiveness in multi-agent systems

a) Models for forgiveness

In [19], Asimina Vasalou and Jeremy Pitt present a model for forgiveness in the context of multi-agent systems. The model they propose is to be used with Digital Blush, their shame and embarrassment management framework. The principle is simple: under certain well defined condition, after an action performed by the agent A, the action being damageable for an agent B in collaboration with A, B can decide to forgive A (and for instance not to decrease the trust value B has towards A). It is typically the kind of development undertaken to fight against the side-effects of anonymity and the lack of social cues in virtual communities, in a more "human" way, including possibility of apologise, reparation, regret…

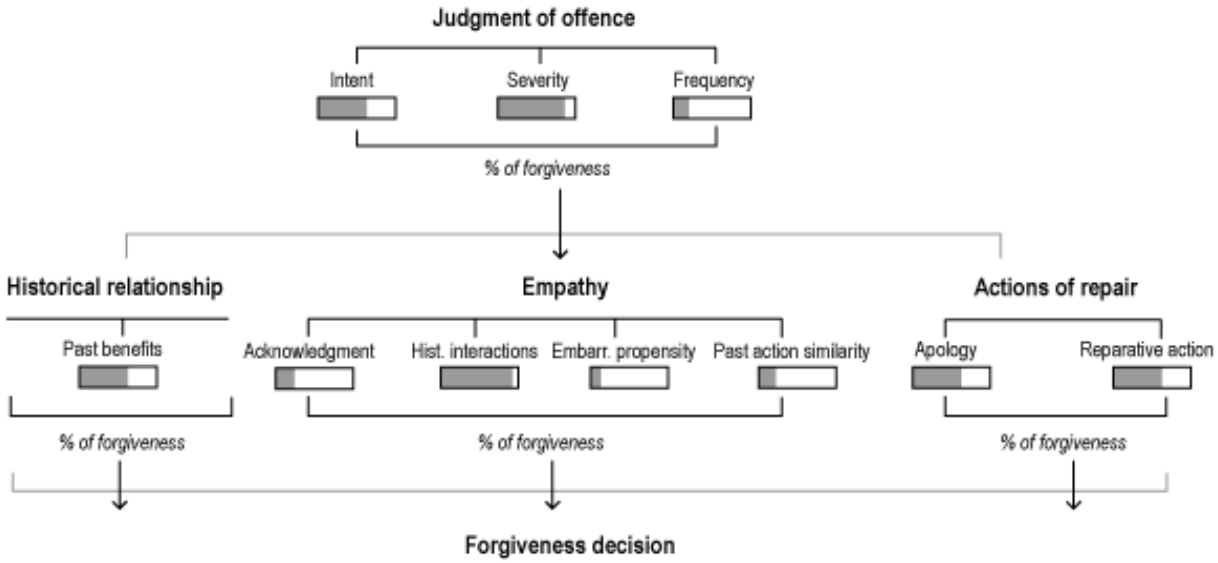
The interest of a forgiveness model is multiple. It can be a purely "ideological" motivation, in "high care" agent organisations for instance: forgiveness is the kind of behaviour that helps agents building social links with others, and maintain positive attitude towards other agents and society in general. In terms of pure performance, it could be an improvement in the case of a normally efficient agent failing one action, for some temporary reason. Without forgiveness, the reputation of this agent would be affected and the other agents could be induced to rely on other individuals, possibly having a lower efficiency. With an effective, well designed and implemented mechanism of forgiveness, we could imagine that the social link would not be damaged by an isolated misbehaviour. The model for forgiveness should of course be robust and strategy-proof: one could easily imagine "evil

agents" misusing the capabilities of a badly designed forgiveness mechanism to perform antisocial actions without any consequences (or with a lower one) on their reputation. Therefore it is important to note that forgiving is not forgetting. From a purely psychological point of view, it is known as it is said in [19] that "the act of issuing forgiveness alone is known to stimulate the offender into positive actions of repair", whereas "punishing the offender for a low intent action [..] will often result in anger and low-compliancy behaviours". The power of the forgiving behaviour in human communities can be an interesting justification of an attempt to test a similar mechanism in an affective software context.

This model of forgiveness is based, as we said on a shame and embarrassment model [3], [15]). The idea is to use shame, embarrassment and blushing in order to show the attitude of the offender towards its own passed action. Then the offended agent can evaluate this attitude and (partly) base its decision on its measures. This is a very important part of the mechanism: the offender has to acknowledge the offence, and to display its related emotion, in order to generate sympathy and potentially provoke forgiveness.

The work present in [19] focuses on the importance of the acknowledgement of the action by the offender. As it is said, "the positive consequence may occur only when the transgressor takes responsibility and cares about his/her action. […] In the absence of punishment, the one in violation is spared of responsibility and is therefore encouraged to maintain a harmful position". That is why embarrassment and shame management is a powerful tool when building a forgiveness decision mechanism.

The decision of forgiveness is based on four quantified measures: the judgement of offence, the Historical Relationship, the offended agent's empathy, and the "actions of repair". Each of these results is a partial forgiveness percentage, all of these being considered together when taking the final decision. The following figure (quoted from [19] synthesises this).



In all the following we consider an offender x and an offended agent y, the offence being the action a. All the formulas and the formalisms given here do not come from the original paper; they constitute a kind of interpretation of the existing work. Except when told differently, all the measures are considered in the interval [0,1], or as percentages. We will arbitrarily define the overall forgiveness value as:

$$Forg = \alpha.Forg_{offence} + \beta.Forg_{historic} + \gamma.Forg_{empathy} + \delta.Forg_{repair}$$

With α, β, γ, δ summing to 1. One should keep in mind that it is just a first proposition, and relationships other than linear could be more efficient (for instance, it could be interesting to set sigmoidal activation functions and thresholds…).

The Judgment of offence is a mix between a measure of the offender's intent about the action, the severity of the action and the frequency of similar actions, performed by the same offender. The frequency can be measured easily if a history of transactions is kept by each agent (which seems to be necessary), but severity and intent seem more fuzzy and subjective. How could an agent measure another agent's intent in a given action? It could be the result of an internal evaluation (with an algorithm chosen by the offended agent y, for instance). About the severity of the action, it is obviously context- and application-dependent, and should be precociously defined with the users and the designers of the system. A simplistic interpretation of this would lead to a first component of the forgiveness value:

$$Forg_{offence} = \alpha_1.(1 - Intent(x, y, a)) + \alpha_2.(1 - Severity(a,y)) + \alpha_3.(1 - Frequency(x, y, a))$$

With α_1, α_2 and α_3 summing to 1. Note that we considered here that the severity of an action can be dependent on the offended agent. Other components could be taken into account in supplementary context parameters.

The "historical relationship" factor is a unique value, also based (like frequency) on the history of transactions of the agents: the past benefits, which could be for instance a ratio between the evaluated utility of the performed actions (normalised between 0 and 1, values below 0.5 corresponding to damages rather than benefits) and an evaluation of the total cost of these actions (also normalised). That would lead us to something like that:

$Forg_{historic} = PastUtility(x, y)/PastCost(x, y)$

Once again, we could introduce a more complex activation function if we want the negative results to influence more severely the forgiveness component.

The empathy component is based on the acknowledgement of the action by the offender, a summary of the historical interactions, an "embarrassment propensity" and a measure of y's past actions similarity. This component is a measure of how much the offended agent is linked with the offender, and how much it is ready to understand it. The acknowledgement measure can be for instance a measure of the offender's blush, or a value based on a specifically defined acknowledgement message. The historical interactions measure is in fact an evaluation of the "density" of the history of transactions, a measure of frequency or just a cardinality value. The embarrassment property is an internal value of the offended agent, which could be more or less subject to empathy and embarrassment. One sould note that the past actions similarity measure refers to y's actions, not x'ones! The idea being that if y has performed in the past several actions similar to the present offence, it will identify itself to x, and be more subject to empathy towards it. Once again a simplistic interpretation would lead us to:

$Forg_{empathy} = \gamma_1.Ack(x, y, a) + \gamma_2.HisDensity(x, y, a) + \gamma_3.Emb(y) + \gamma_4.PastActions(y, a)$

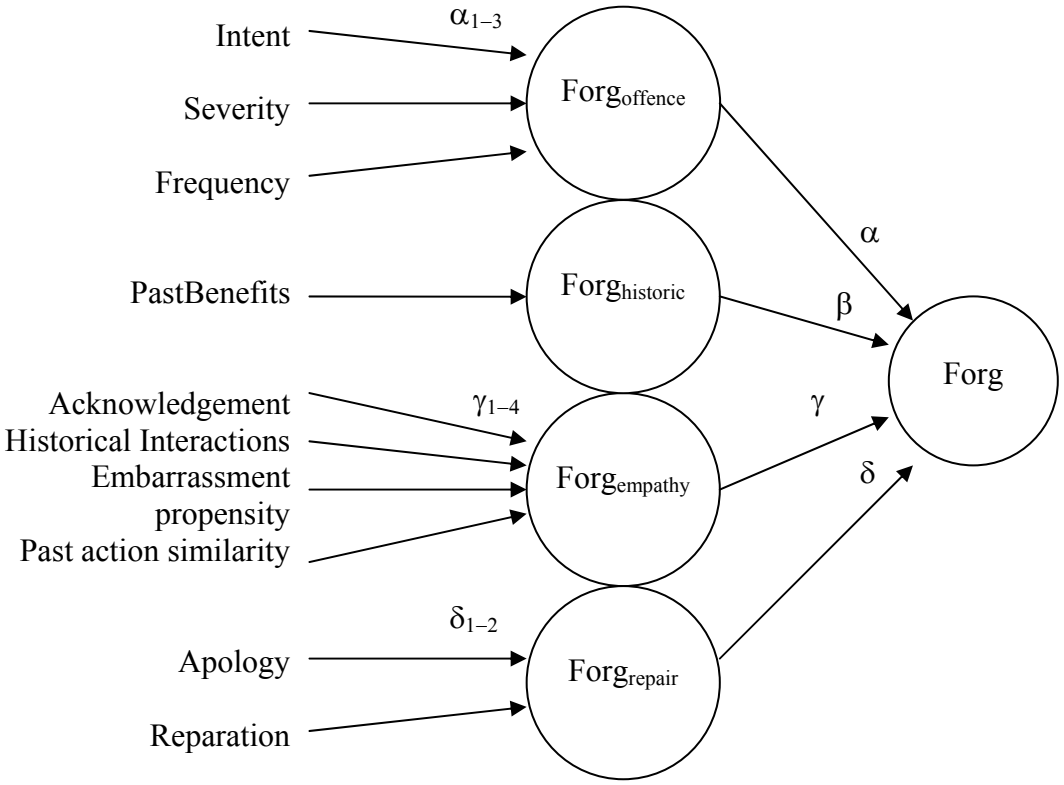
With γ_1, γ_2, γ_3 and γ_4 summing to 1, of course. Once again, more complex activation functions could be used.

The last component is the Repair component, based on both apology from the offender, and the reparative actions undertaken by it. Both values have to be defined with respect to the design of the agent architecture, for such notions can be expressed in different ways. Once again:

$Forg_{repair} = \delta_1.Apology(x, y, a) + \delta_2.Repair(x, y, a)$

With δ_1 and δ_2 summing to 1.

As one can guess, with such a model it is very difficult to set all the parameters, and the learning phase can be long and fastidious. In the paper a survey system is proposed with subjective entries in evaluation tables. Given the equations we have previously given, one could remark that the computation is exactly the one of an artificial neural network:



This is the network derived from the simplistic interpretation of the model in [19], but other architectures could be more efficient, for example the introduction of a real input layer, allowing different activation functions for each parameter. With such a structure, combined with the kind of tables one can find in the appendix A of [19], it would be easier to set the parameters of the system (for instance by fixing some inputs in the learning phase of the neural network, when necessary) but it would still need a good "intuitive" initialisation and lots of training data.

There is also the problem that in a real application, offences would be of different "classes", so the computational model (and the survey/learning mechanism) should take that into account. Different kinds of offences, in a given context, with the same values, do not lead

to the same forgiveness decision. Maybe some parts of the system should be implemented in a different way for each kind of violation defined in the application.

One could remark that in that model, information coming from the history of past transactions has a multiple influence, in several components of the forgiveness decision. This inelegance could be "erased" by designing a more standard multi-layer perceptron, but we would loose much information about the different components, and thus about the initialisation values.

In this model the output is a quantified measure of the forgiveness decision. This output could lead to a quantified forgiveness action (quantified influence of the offence on the trust value, for instance), or to a quantified advice to the user, telling him/her how interesting or recommendable it would be to forgive in that case. In some complex implementation, it could be coupled to a natural language fuzzy inference engine in order to transmit the advice to the user in a human-friendly way.

As suggested in [19], the forgiveness action (automatically performed by the agent) or the forgiveness recommendation to the human user could be coupled with a proposition (or enforcement) of a reparative action, or maybe a punishment.

b) Mixing trust and reputation with forgiveness

One interesting experiment would be the integration of an implementation of the forgiveness model described in [19] in a trust and reputation management framework like the one presented in [10] and [11]. It is a multi-agent platform based on an economic society (trading agents), which is intuitively a domain where feelings and emotions do not have a significant part in the human counterpart as well as in the specialised software systems. As we all know, the economic market never forgives. For that reason (this intuition of preconceived idea), it may be the ideal environment for testing a forgiveness mechanism. One could measure with precision the impact of forgiveness (and of its parameter settings) on the performance of economico-cognitive agents, and infer the kind of influence it could have in virtual societies more focused on human users and collaboration.

If we want to integrate a forgiveness mechanism in the previously described trust and reputation framework, we will need to slightly modify the memory model of the agent concerning the history of transactions. In order to implement the trust and reputation mechanisms, the least information we need concerning history is two numeric values: an

evaluation of the history, and a degree of trust of this evaluation (for each known agent). For instance the mean utility of the undertaken collaborative actions, and the cardinality of these actions (but the measure could be much more complex). Anyway, it is not enough if we want to compute all the inputs of the forgiveness calculus.

We could use for the forgiveness computation, according to what we said sooner, this mean utility and cardinality of transactions. However, when an offence is forgiven, it has to be reminded by the forgiveness part (in order not to forgive indefinitely the same offence… it would be a huge security breach), but not necessarily by the trust management part (this is the point in forgiving)! Thus we have to have one set of values for each part; we cannot share the values between trust and forgiveness. We could keep an exhaustive history of past transaction, but for most applications it is not acceptable, so we have to derive partial information variables for each input component of the forgiveness computation. Thus we will keep a mean utility of past transactions with the agents and a cardinality of these transactions, for the "past benefits" input. We also need a history of the offences of each kind from each known agent (it could be a fixed dimension array for instance, the dimension being the number of different kinds of offences) for the "frequency" input in the Judgement of Offence. The cardinality of past transactions could be used to measure the history of interactions in the empathy component. For the past action similarity, we need that each agent keep an array of the offences he has done (or performed actions that could be considered as offences in a given context).

We also have to define an "embarrassment propensity" for the agent. This sole point could justify a whole study. The agent should also be able to measure, by some mean, the severity of the offence, the intent, the acknowledgement, apology and reparative actions from the offender. These notions are much application-dependent.

If we look at the trust management mechanism figure in part 2.a of that document, we can try to identify how we could integrate the forgiveness computation in the loop. Apparently it would be a derivation of the rightmost arrow (leading to the modification of experience values in the trust management framework). The information for the action resulting from the trusting decision (measure of the potential offence) is taken as an input for the forgiveness decision computation. The agent "switches" to the forgiveness part, takes measures, computes the level of forgiveness. After that, depending of the interaction model we have chosen, the agent can:

- Emit a recommendation to a human user and wait for his/her decision (no forgiveness, total or quantified forgiveness – option obviously not desirable in the case of this application!)
- Decide itself between no forgiveness, total or quantified forgiveness

Depending on the previous decision, the agent will modify or not (or at a given level, corresponding to the forgiveness quantification) the experience values in the trust and reputation management part. Of course the values in the forgiveness part are updated whatever decision is taken (the offence is there anyway!)

This basic architecture idea would allow us to mix efficiently forgiveness with trust and reputation, in order to measure the impact in terms of performance.

c) Future work on forgiveness

We still have to study more thoroughly the actual implementation of the trust and reputation framework in which we are interested, before building a more precise architecture integrating the forgiveness mechanisms. Before that, the theoretical formal model for forgiveness should be affined and defined more precisely, and maybe the idea of the neural network should be developed. If this solution is chosen, much attention should be given to its architecture. The problems inherent to initialisation and parameter settings (weight learning of the network) are highly dependent of the technical solution, so they should be addressed at this stage.

All the application-dependent notions still have to be defined: categories of offences, severity, intent, acknowledgement, embarrassment propensity, apology, reparative actions. The experimentations should be performed in order to evaluate the impact of various definitions or settings for these.

A complete implementation, with correct parameter settings, would allow us to run performance test of the same kind as the ones described in [11], in order to compare the pure trust and management framework, and the "forgiving" one. More detailed test could be designed to study the influence of the settings and the inputs. More "forgiveness-specific" tests are likely to appear as desirable when we begin to implement the system.

This study should allow us to gain a precise idea of the influence of (this model of) forgiveness in cognitive agent societies.

5. Conclusion

In this study we have seen what the current challenges are in affective computing: how to express emotions, how to recognise emotions, how to model and use emotions in software agents in order to improve performance and believability. We have had a brief overview on the different affective interface issues, and the different emotional models that could be integrated in cognitive software agents.

However, it appears that sometimes people working in affective computing want to integrate emotion-like human behaviours in their systems by all means, and it might not be always desirable, be it in multi-agents or human-computer interactions issues (or a mix of both of course). In [7], Kristina Höök says that "the field of affective computing often make simplistic statements where it is claimed that users will more easily bond with an affective system, become more efficient if not stress or disturbed at the right moment, etc." This is an example of what we said: sometimes it is better not to add a component, even "affective", because it will not lead to a real improvement. Affective computing seems to be subject to fashions, like many technologies and sub-fields in computing…

We went further into a specific kind of affective computing development, dedicated to online communities and multi-agent systems: the propositions of Jeremy Pitt and Asimina Vasalou for a forgiveness mechanism. We have begun to see how this could be implemented and tested, but much work remains to do before we can measure the impact and the efficiency of integrating this notion in the particular kind of multi-agent system we have described.

6. References

[1] Burleson W, Picard R, Perlin K, Lippincott J, *A Platform for Affective Agent Research*, Workshop on Empathetic Agents, International Conference on Autonomous Agents and Multiagent Systems, Columbia University, New York, NY, July 2004.

[2] Castelfranchi C: *Trust mediation in knowledge management and sharing*, Proc. Second International Conference on Trust Management (iTrust 2004), pp. 304-318, 2004

[3] Castelfranchi C, *Affective Appraisal vs Cognitive Evaluation in Social Emotions and Interactions*, In Paiva A., Affective Interactions, Towards a New Generation of computer Interfaces (pp. 76-106), Springer, 2000.

[4] Castelfranchi C, Tummolini L, Positive and Negative Expectations and the Deontic Nature of Social Conventions, ICAIL 2003 (pp. 119-125), 2003.

[5] Falcone R, Pezzulo G, Castelfranchi C, *A Fuzzy Approach to a Belief-Based Trust Computation,* Trust, Reputation and Security 2002 (pp. 73-86), 2002.

[6] Goulev P, *An Investigation into the Use of AffectiveWare in Interactive Computer Applications*, PhD thesis (supervisors Pr Abe Mamdani, Dr Jeremy Pitt), Imperial College London, 2004, pp. 13-16.

[7] Höök K, *Evaluation of Affective Interfaces*¸ In Evaluating ECAs, edited by Catherine Pelachaud and Zsófia Ruttkay, Kluwer (2004?).

[8] Krenn B, Gstrein E, Neumayr B, Grice M, *What can we learn from users of avatars in net environments?*, In Catherine Pelachaud, Zsofia Ruttkay (eds), Evaluating ECAs, Kluwer (2004?).

[9] Isbister K, Doyle P, *Design and Evaluation of Embodied Conversational Agents: A Proposed Taxonomy*, in Embodied Conversational Agents: Let's Specify and Evaluate Them, AAMAS 2002, Bologna, Italy, 2002.

[10] Neville B, Pitt J, A Computational Framework for Social Agents in Agent Mediated E-Commerce, ESAW 2003 (pp. 376-391), 2003.

[11] Neville B, Pitt J, *A Simulation Study of Social Agents in Agent Mediated E-Commerce*, Seventh International Workshop on Trust in Agent Societies, AAMAS 2004, 2004.

[12] Nijholt A, *Humor and embodied conversational agents*, CTIT Technical Report, 03-03, University of Twente, Enschede, ISBN 1381-3625, University of Twente, Pages: 15, 2003.

[13] Picard R, *What does it mean for a computer to "have" emotions?*, Chapter in "Emotions in Humans and Artifacts," ed. by R. Trappl, P. Petta and S. Payr, 2001.

[14] Picard R, *Affective Computing: Challenges*, Int. Journal of Human-Computer Studies, Vol. 59, Issues 1-2, July 2003, pp. 55-64.

[15] Pitt J, Digital blush: towards shame and embarrassment in multi-agent information trading applications, Cognition, Technology and Work, 2004.

[16] Pitt J, Artikis A, *Socio-Cognitive Grids: A Partial ALFEBIITE Perspective*, Intelligent and Interactive Systems Group, Dept. of Electrical & Electronic Eng. Imperial College London, 2003.

[17] Prada R, Vala M, Paiva A, Hook K, Bullock A, *FantasyA – The Duel of Emotions*, in Proceedings of the 4th International Working Conference on Intelligent Virtual Agents - IVA 2003, 2003.

[18] Ruttkay Z, Dormann C, Noot H, *Evaluating ECAs – What and How?*, in Embodied Conversational Agents: Let's Specify and Evaluate Them, AAMAS 2002, Bologna, Italy, 2002.

[19] Vasalou A, Pitt J, Reinventing forgiveness: a formal investigation of moral facilitation, iTrust 2005, 2005.