Autonomous Avatars? From Users to Agents and back

Matthias Scheutz[1] and Brigitte Römmer[2]

[1] University of Notre Dame, Notre Dame, IN 46556, USA
[2] Universität Wien, Vienna, A-1010, Austria

Abstract. We describe the architecture of an interactive, "believable" agent with personality, called *user agent*, which can act on behalf of a user in various multi-user game contexts, when she is not online. In a first step, information about the personality of the user is obtained from a questionnaire and then, in a second step, integrated in the reactive system of the user agent, part of which implements a primitive affective system. User agents can interact with their users through a simple affective natural language generation system (SARGS), which is integrated in the deliberative system of the user agent and can recount what happened to the user agent in the game while the user was not present.

1 Introduction

Recently, agents *with personality* have become a new area of interest in AI research (e.g., [17, 10, 8] and others). The rapid increase of multi-user dungeons (MUDs) and other kinds of online multi-user games provides novel territority for the application of agent technology: in virtual worlds autonomous agents can take on roles, which usually are of little to no interest to human players, either because the role is boring or short-lived (e.g., "the shoemaker down the corner" or "the soldier in the first row of the formation"). Agents may also function as "tour guides" and "sources of information" within the simulation (e.g., see [6]). To make games with agents exciting and entertaining, the "believability" of computer controlled characters [2] is crucial. Synthetic characters that always behave in a repetitive, clearly predictable way, are typically perceived as boring and users might not only loose interest in them, but *mutatis mutandis* in the whole game.

In MUDs, which are not solely text oriented, but based on 2D or 3D simulations of some virtual world (that may or may not have a graphical representation), users need to be present (or online) to control their *avatar*, i.e., their representation in the simulated world. However, since it is not possible for users to play the game all the time, the user's character will not be able to take part in all events. While this does not matter in some contexts, a continuation of the character's interactions can or may be possible and desirable in certain game contexts (e.g., when several players have to cooperate and coordinate their moves to attain a goal).

In this paper, we describe an approach to cope with this difficulty. Instead of classical avatars (which are completely controlled by users), we propose a special kind of *autonomous agents* to represent users within a game simulation. These agents, call them *user agents*, are really hermaphrodites between avatars and autonomous agents in that they live an autonomous life, if their users are offline, acting their behalf, but can be controlled by users, who are online. The availability of such agents will allow game designers to create a variety of different games, in which users can join in and participate at their leisure (without becoming marginalized within or excluded from the game if they cannot play very frequently).

We start out by placing our agent model in the context of a particular game and discussing the difficulties that arise out of integrating user personalities in agent architectures. We then briefly review different models of human personality and describe the hybrid reactive-deliberative architecture of the proposed user agent. In particular, we present the simple affective language generation system (SARGS), which is part of the deliberative system and allows user agents to report in an entertaining story-like manner what happened while their users were offline. Finally, we summarize the advantages and disadvantages of our agent architecture, briefly discussing immediate as well as possible applications and extensions.

2 From Users to Avatars

The user agent is intended to be used in massive multi-user online games with a strong community building character, where agents have to represent and act on behalf of their users. The game context, in which the user agent has been developed, is a *dating game*[1], where agents are "sent out" in a virtual world on behalf of their users to find a flirt partner (as represented by another agent). When users log on, a report of their agents' "experiences" (e.g., how many "dates" their agents had) is dynamically generated using text templates, while a (minimally animated) 2D graphical representation depicts their agent in a cartoon-like fashion in a particular part of the game environment (e.g., a bar).[2] A much simpler, predecessor version of the user agent has already been used successfully in this game environment.

2.1 How to Model Users in Game Contexts

Since user agents need to act on behalf of their users, certain user characteristics (in particular, aspects of the user's personality and character traits) need to be modeled insofar as they determine basic decision making and behavior of

[1] For commerical and legal reasons we are not permitted to reveal any details about the game beyond what is presented here.

[2] The main reason for such a reduced graphic realization is that user should be able to play the game over the internet with standard web browsers, i.e., without having to download and install simulation software.

the agent. However, modeling users is not trivial, as their limited patience, and willingness to reveal large amounts of personal data severely constrains the kinds and quality of applicable models. Consequently, one needs to be pragmatic in finding a compromise between the accuracy of the personality model and the information available, which is usually obtained from some sort of questionnaire. The questionnaire typically contains one or more questions for each relevant (psychological) dimension, from which a *user profile* is generated.

There are two ways of incorporating the user profile in the game, which differ with respect to the degree to which the user profile is integrated in the agent: the first option keeps the user profile separate from the agent architecture and only derives general action tendencies for the user agent from it, which serve the role of "meta-knowledge" about how to act in certain situations. It is by following behavioral guidelines determined by such meta-knowledge about the user's action tendencies that the agent reflects the user's personality to some degree.

The alternative, which we pursue in this paper, is to integrate the user profile directly into the agent architecture, i.e., to "recreate" to some (in our case very limited) extent the user's personality in the agent. Obviously, this is a much harder problem, but will in the end contribute significantly to the believability of the agent. While some projects have integrated human personality models in agents [14], they have not attempted to model human players in agents participating in the game.

2.2 A Brief Overview of Models of Human Personality

A commercially popular model of human personality is based on the works of C.G. Jung.[3] Keirsey and Bates [9] suggest to analyze human personality along four dimensions, which they experess in terms of binary categories yielding 16 types of personality. The model does not only make detailed assumptions as to what the interests and behaviors of each individual type are (in public and in private life), but also suggests whether and how well each type gets along with any other type. The definition and implementation of such a model, however, would be rather complex if not practically infeasible, since for each possible situation all possible actions of all 16 personality types would have to be determined in advance and explicitly represented within the game simulation (e.g., in the form of condition-action rules). How such a model could be embedded in an agent architecture is unclear.

A different analysis of human personality, the Eysenk PEN-model [1] proposes three dimensions of personality, namely *psychoticism* (with the extremes "troublesome" - "socialized"), *extraversion* ("sociable" - "quiet"), and *neuroticism* ("anxious" - "calm"), the extremes of the dimensions being pathological. While Eysenk's suggestion that all three dimensions are ultimately linked to physiological states seems to make the model applicable (at least for *embodied* agents), the details of this relation are not clear (e.g., see [3]). And, furthermore,

[3] Such models are typically used in consulting (e.g., for the management of teams).

given the model's origin in clinical research, some of the questions used in the *Eysenck Personality Questionnaire* [5] are not suitable for a game context.

In psychological research, the dominant model of personality is the 5-factor-model, or "Big 5" [7]. It is based on a factor analysis yielding *extroversion*, *neuroticism*, *openness*, *agreeableness*, and *conscientiousness* as the basic dimensions of human personality, and has also been used as venture point for attempting to equip agents with human personality [14]. The problem with "Big 5" (as with the other above-mentioned models), however, is that it is silent about the relation between personality and affective states. There is one model, which claims to provide some of this information–Mehrabian's three dimensional PAD model [12] (whith dimensions "pleasure" - "displeasure", "arousal" - "nonarousal" and "dominance" - "submissiveness"). The PAD model is intended as a model of human temperament as well as human emotions [13], where the influence of personality on emotional states is given by the very fact that the basic dimensions of personality and emotions coincide (different emotions and personality types are viewed as variations along these three dimensions).

2.3 Integrating Aspects of User Personalities in Agent Architectures

Given the difficulties of obtaining an accurate user profile in game context,[4] we decided to use a hybrid model of personality and emotion, which is inspired by the PAD Model: the dimensions which represent emotions (or, more generally, affective states), are also taken to be basic to personality. To the three dimensions *pleasure*, *arousal*, and *dominance* we have added a fourth dimension, which determines the general attitude towards anything new, as well as the agent's expressivity.[5]

This affective system can be thought of a four-dimensional dynamic system, whose state space corresponds to the personality of the agent. Different regions in this four- dimensional space correspond to different emotions (in analogy to the PAD model), and we say that if the system is in a state, which is part of one of those regions, that the system is in the *affective state* (or, put metaphorically, "has the emotion") associated with this part of the space. In analogy to the PAD Model, low pleasure, high arousal, and high dominance, for example, may define "anger", whereas low pleasure, high arousal and low dominance may define "fear". Through external input the system will be able to change states and traverse the space on trajectories that are determined by the original setup (i.e., personality) and current input.

---

[4] The psychological models ofq the previous section all rely on questionnaires of around 50 questions to yield reliable personality measures. However, the type of game we refer to is aimed at the casual player, who tends to loose interest quickly. From experience with previous questionnaires (used to "personalize" the predecessor version of the user agent) we learned that the upper limit of an acceptable number of questions is about 12.

[5] Note that we do not claim that this fourth dimensions is basic to a model of human personality, but rather we introduce it because of its usefulness for generating behavior which is comprehensible for the user.

Since the "personalization procedure" needs to be as brief and efficient as possible, we choose questions directly corresponding to the four dimensions.

Example You think you did a fine job writing a summary report of last week's meeting, but your boss rips it to pieces, because he does not like the format. How do you react?

(a) I apologize and offer to reformat the report immediately.
(b) I tell him not to be ridiculous.

The example corresponds to the dimension of dominance, where answering (a) leads to a more submissive, (b) to a more dominant personality of the user agent.

Another type question, employed to make the actions of the agent transparent to the user, asks for preferences regarding actions which are of importance in the context of the game. Whether an agent in the dating game will tend to ignore hunger if excited, for example, may depend on whether the user provides a particular answer to a question like the following:

Example Suppose you have been craving a juicy steak all day long. As you finally sit down to order, this gorgeous, foreign-looking babe walks up to you and asks you whether you could recommend any sights for her to see. What would you do?

(a) Tell her that you are busy and call the waiter.
(b) Smile at her and ask her if she wants to join you.
(c) Immediately get up and show her around the city.

The user's answers are then used to adapt the affective system of the agent. Depending on the representation of this dynamic space, different adaptations methods can be used. E.g., for affective systems realized in trainable neural networks, where affective states are represented by the degree of activations of various units, question-answer pairs can become part of the training set.

This affective model is a pragmatic solution, which allows for great flexibility, since the agent can be easily adjusted to various game contexts and even modified during a game. For example, it is possible to start a game with a user agent (based on inital questionnaires with only a few questions) and then retrain the agent as more information about the user becomes available (e.g., because users are asked more questions in the course of the game).

3 The Architecture of User Agents

We use a hybrid architecture for user agents, which consists of a reactive and a deliberative system. The former controls inputs to, outputs from, and bodily functions of the agent, and implements a primitive affective system. The latter is concerned with planning, reasoning, and in particular the communication with the user. Since user agents should also be applicable in mere text-based games (e.g., games played using wireless devices such as cellular phones with SMS or WAP, etc.) and not limited to games with graphical interaction, it is crucial to provide a natural language generation (NLG) system, which updates users on past and current events, and, furthermore, allows them to learn about the experiences of their agent (its motivations, feelings, etc.).

3.1 The Reactive System

The reactive system integrates external and internal sensory information to produce the basic behaviors needed for the "survival" of the agent. In particular, it is in charge of satisfying two basic bodily needs: "hunger" and "thirst" (which can be satisfied in various ways depending on the game contexts), and also implements the rudimentary affective personality model described above.

In our current implementation, the reactive system is realized as a three-layer neural network, where the input layer is divided into two classes: inputs from exteroceptive sensors or perceptual systems *Extinput* (e.g., vision) and inputs from proprioceptive sensors *Intinput* (e.g., the level of energy). Similarly, the output layer is divided into units representing "inner states", called *Context* (such as curiosity or arousal), and units representing actions *Action* (such as drinking or moving in various directions). A probability-based action selection mechanism is employed: the activation of the *Action* is taken as the probability that the action associated with the unit will be *attempted* by the agent–whether it can actually be executed will depend on various additional factors (i.e., energy level, constraints of the current environment, etc.).

The mapping from sensory input and internal states to action and internal states is learned using a slightly modified, faster version of the backpropagation algorithm. The training set for supervised learning contains some general rules, vital to the agent's survival (e.g., the rule "if you see food and hunger is high, attempt to eat it"), but also rules derived from the results of the user questionnaire. Note that even for identical answers on the questionnaire the trained neural nets will vary slightly because their weights are initalized at random (such variations are quite welcome as experience suggests that monotonicity and similarity of agent behavior is a major cause of "boring games").

After successful training, which takes place every time a new agent is created (i.e., when a new user joins the game), the network is partly rewired: the connections from the *Intinput* units to the hidden units are connected to the *Context* units, which now constitute a "context layer"–it still receives input from the proprioceptive sensors, yet the previous activations of its units are also taken into account (which, strictly speaking, augments the reactive system by "inner states"). Furthermore, the update rule for those "context" units is altered to implement the personality model: while all other units use a sigmoid activation function to compute their activation and output values, these units use an IAC update rule [11]. For all four dimensions of the affective system, rest levels and decay rates are set in accordance to the basic personality type as determined by the user questionnaire (e.g., an agent for a user with "high dominance" will have a high positive rest level of the "dominance" context unit).

3.2 The Deliberative System

The deliberative system provides the basic natural language interface for user interaction and, furthermore, allows for additional planning and reasoning components, which permit game designers to extend the agents' capacities and adapt them to specific game environments. The natural language interface uses an additional "text sensor" to read in commands from the user and responds through an additional "text effector" using the *simple affective report generation system* (SARGS), to be described below. SARGS is a rudimentary NLG system, which is fully integrated in the deliberative layer (thus amenable to changes in the deliberative rule system, although these possibilites are not taken advantage of at present). It can be embedded in a lager hybrid NLG system of the game simulation, which in the dating game uses text templates [15] to provide background information about the current state of the game.

The deliberative system also provides short-term and long-term memories, which can be accessed by components extending the deliberative capabilities. The former is standardly used for the storage of perceptions and recent experiences, while the latter typcially contains knowledge about the simulation environment, the agent's past experiences together their "affective evaluation" (see the next section), and rules about what to do (and how to do it). Depending on the particular game, different rules will be part of the agent's memory (e.g., in the dating game there is a calendar for the agent to enter dates and schedule appointments, and a global directory of the simulated world, which the agent can use to plan its moves, meetings, etc.).

Currently, the deliberative system is implemented in POPLOG (which implements a PROLOG virtual machine among others). The databases for short and long term memory is mapped onto the built-in database (available in PROLOG). Any kind of *factual knowledge*, past experience, etc. is stored similar to a PROLOG *fact* (with a time stamp to avoid inconsistencies of the database). For example, if the agent meets another agent at time 145, the following facts about this meeting might be stored: $meet(a12, 145)$, $woman(a12)$, $hair(a12, brown)$, etc. Furthermore, the agent also stores its affective status, e.g., $arousal(0.8, 145)$, $hunger(-0.7, 145)$, etc.[6] *Condition-action rules* can be applied to those facts (similar to PROLOG *clauses*), where antecedents may contain a default conjunct to allow for non-monotonic extensions [16]. Planning can be achieved using forward chaining (or, if the PROLOG virtual machine is used, unification on a particular goal clause with a free variable [18]). While no particular planning (or reasoning) mechanism is part of the agent architecture, an interface to the action selection mechanism is provided for a planner to pass a sequence of actions with corresponding probablilities that the action will be attempted (a probability of 1.0 means that the action will be attempted with certainty). It is also possible

[6] Note that a new constant is automatically generated for each perceived entity, and if the encountered entity can identified (e.g., as a previously known agent by virtue of a name), then the constant will be automatically replaced by that identification (e.g., if $a12$ turns out to be the same agent as $a3$, never mind how, then all occurrences of "$a12$" will be replaced by "$a3$").

to "override" the reactive system completely by substituting the plan actions for whatever action the reactive system may have selected in a subsumption style manner.

3.3 The Simple Affective Report Generation System (SARGS)

SARGS was conceived as a tool to allow users to learn about the "inner lifes" of their agents and to be able to understand better, why their agent chose a particular action and not another. The idea is that agents describe what they "perceived" and how they "felt" at a given time, what action(s) they chose in response to their perceptions and inner states and what actions were actually executed. Call such a tuple $\langle P_t, F_t, I_t, A_t, \rangle$ (consisting of a set of perceptions $P_t = \{p_{t,1}, p_{t,2}, ..., p_{t,n}\}$, a set of "feelings" or "inner states" $F_t = \{f_{t,1}, f_{t,2}, ..., f_{t,k}\}$, a set of action intentions $I_t = \{i_{t,1}, i_{t,2}, ..., i_{t,m}\}$, and a set of performed actions $A_t = \{a_{t,1}, a_{t,2}, ..., a_{t,j}\}$) an *event* E_t at time t. A *story* of length l starting at time t, then, is a sequence $S(t, l) = \langle E_t, E_{t+1}, ..., E_{t+l}\rangle$ of events.

To create a report, SARGS first retrieves all events from short term memory that are part of the story $S(t, l)$, where t was the last time the user inquired about the agent's *status quo* and $t + l$ is the current point in time. From this story SARGS produces a sequences of natural language sentences, which describe the story to the user in an *entertaining* way. To be entertaining, the story should at least

1. have an appropriate beginning and ending (i.e., set up the context for users to get into the right mind frame to see things from their agents' perspective– e.g., "Hi. Haven't seen you in a while. Last time you checked in I was not doing all that well..." or "So, that's where we are right now.")
2. have an element of non-determinism (i.e., if the same story is recounted twice, word choice and sentence structures should be somewhat different and vary from time to time)
3. use affective language to evoke emotions (such as sympathy, empathy, etc.) and express the degree of activation of the reported affective states (e.g., "...quite excited...")
4. either omit states that have not changed over a pre-determined interval of time or report them as such (e.g., "Still hungry, ...")

SARGS is an affective NLG [4], which uses the agent's affective states for content planning, i.e., for deciding, which aspects of the description of perceptions, inner states, and actions (intended and performed) of an event to describe to the user based on its current affective states: the higher the activation of a state, the more likely that the corresponding aspect will be part of a description. Furthermore, depending on the length l of the time frame to be described, events will be ignored and erased from short-term memory if their affective evaluation is below a certain threshold to avoid long-winded and boring stories (the remaining events will be transferred to long-term memory).

For sentence planning and realization, it uses a *grammar database* of schemas of different sentence types (declartive sentences, questions, etc.). For each event, a schema is selected at random (also taking into account the schema used to describe the previous event). For example, a schema for perceptions could look like this: $\mathsf{S} \rightarrow \mathsf{TAdv}\ \mathsf{N}\ \mathsf{V}\ \mathsf{NP}$ (where S is a sentence, TAdv a temporal adverb, N a noun, V a verb, and NP a noun phrase, which itself can be decomposed into a definite or indefinite article, possible adjectives, and a noun).

To describe a particular event, the grammatical categories in the sentence schemas are filled in with words from a *word database*, which contains a variety of different applicaple words (or expressions) for every sensory modality and possible percept, every inner state, and every possible action. The perception of a bar in the dating scenario, for example, could then be mapped onto the above schema: "Then I saw a cool bar". Furthermore, the database contains various adjectives, adverbs and conjunctions, which can be used to connect sentences and to describe properties of perceived entities (e.g., "cool") as well as degrees of "feelings".

Special rules are used to deal with "aggregation" [15] within and across events. By keeping track of what changes and what stays the same between two events, SARGS determines which inner states to talk about: only those are chosen that have actually changed within some predetermined time interval (as users are not interested in learning every time that their agent is "hungry", for example). It then uses adverbs such as "still", "again", "not yet", etc. to describe the developments of parts of events (perceptions, inner states, intentions, and performed actions) over sequences of events. Furthermore, it uses other adverbs such as "very", "not much", etc. to describe the intensity of a "feeling" (i.e., the degree of activation of an inner state) [4]. For example, if an agent has an arousal level of 0.8 (where 1.0 means "maximally aroused"), which was at only 0.3 at the previous event, SARGS might add the following phrase to its description of the current event: "...even more excited...".

4 Discussion and Future Work

We have proposed an architecture for agents, which can represent users to a certain degree in game environments. A simple characterization of the user's personality is obtained from a user questionnaire and used to adjust parameters in the agent's control system, which model (in a simplistic way) these personality traits. The main advantages of our design are:

- The agent's sensors, effectors, and inner states can be easily extended by adding units to the neural network and new rules to the training set.
- User characteristics beyond the simple personality model are modeled (e.g., concrete decisions in particular circumstances).
- SARGS is a first attempt to recount a story according to the user's own preferences (insofar as they are determined by personality and affective states), while at the same being quite adaptive (i.e., it can be used in different game contexts with only minor modifications of names of objects, locations, actions, etc.).

– By virtue of using neural networks, somewhat inconsistent user profiles–users sometimes seem to contradict themselves on questionnaires–can be used to define agents (although the compromise found by the learning algorithm may not be always be appropriate).

The present version, however, has also some disadvantages:

– If the update interval is high, running many agents on one system becomes infeasible. Furthermore, if users rarely check on their agents, huge amounts of data will accumulate in each agent's short-term memory.
– Depending on coding, number and nature of the rules, the backpropagation algorithm might not be able to learn a sufficiently adequate mapping within a reasonable timespan (or, in the worst case, not at all)–general constraining factors are needed to guarantee that backpropagation will not fail (e.g., estimates on the number of required hidden units, etc.).
– SARGS does not easily extend to a genuine NLP/NLG system, where users can "chat" freely with their agent.
– The affective system only implements a so-called "shallow model of emotions" and higher affective states (such as "frustration", "disappointment", etc.), which essentially depend on a deliberative layer, are not supported [19].

Ideally, at some point, we would like to have agents with personalities *very similar* to those of the users, especially if they are to be used in "social games", where personality and social interaction are the crucial features. Unfortunately, we are still far from achieving this goal and would only deceive ourselves if we took labels such as "surprise", "disappointment", "anxiety", etc. (which too often are prematurely assigned to rather simple control states these days) at face value.

In the current project, we have restricted ourselved to a shallow model of affective states and personality, given the constraint of producing a functional system for multi-user games rather than a tenable computational model of affect in a reasonable amount of time. As of now, our model is implemented in POP11 and currently being tested in the context of the dating game. Once the tests are concluded, the code will be transcribed in JAVA and the model will become part of a new commercial multi-user game, which is played over the internet.

While user agents may find rather immediate application in certain game environments, we see this project, however, as a first step in a long term investigation, which attempts to capture and implement in much more detail the properties and processes involved in human personality and emotions.

References

1. Amelang, M., Bartusek, D.: Differentielle Psychologie und Persoenlichkeitsforschung, Stuttgart, Kohlhammer (1997) 324–335
2. Bates, J.: The role of emotion in believable agents, Communication of the ACM, Special Issue on Agents (1994)

3. Brody, N.: Personality: In Search of Individuality, San Diego: Academic Press (1988)
4. de Rosis, F., Grasso, F.: Affective Natural Language Generation, Proceedings of the IWAI (1999) 204–218
5. Eysenck, H.J., Eysenck, S.B.G.: Manual of the Eysenck Personality Questionnaire London, Hodder and Stoughton (1975)
6. Foner, L.N.: Entertaining Agents: A Sociological Case Study, The First International Conference of Autonomous Agents (1997)
7. Goldberg, L.R.: Language and inividual differences: the search for universals in personality lexicons, Review of Personality and Social Psychology **2** (1981) 141–165
8. Hayes-Roth, B., van Gent, R., Huber, D.: Acting in Character, Proceedings of AAAI Workshop on AI and Entertainment (1996)
9. Keirsey, D., Bates, M.: Please Understand Me. Character and Temperament Types, Prometheus Nemesis Books Company (1984)
10. Loyall, B., Bates, J.: Personality-Rich Believable Agents That Use Language, Proceedings of the 1st International Conference on Autonomous Agents (Agents97), ACM Press (1997)
11. McClelland, J.L., Rumelhart, D.E.: Parallel Distributed Processing, Vol. 1 and 2, MIT Press, Cambridge (1988)
12. Mehrabian, A.: Framework for a comprehensive description and measurement of emotional states, Genetic, Social, and General Psychology Monographs **121** (1995) 339–361
13. Mehrabian, A.: Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament, Current Psychology **14** (1996) 261-292
14. Nicholson, A., Dutta, A.: Intelligent agents for an interactive multi-media game, Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (1997) 76–80
15. Reiter., E.: NLG vs. Templates, Proceedings of the 5th European Workshop on Natural Language Generation, Leiden, Netherlands (1995)
16. Reiter., R.: A logic for default reasoning, Artificial Intelligence **13** (1980) 81–132
17. Rizzo, P., Veloso, M., Miceli, M., Cesta, A.: Personality-driven social behaviors in believable agents, Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents (1997)
18. Russell, S., Norvig, N.: Artificial Intelligence: A Modern Approach Prentice Hall (1995)
19. Scheutz, M., Sloman, A., Logan, B.: Emotional States and Realistic Agent Behaviour, Proceedings of GameOn2000 (2000)