Corroborating Emotion Theory with Role Theory and Agent Technology: a Framework for Designing Emotional Agents as Tutoring Entities

Bogdan Florin Marin
University of Duisburg-Essen, Institute for Multimedia and Software Engineering, Duisburg, Germany
bogdan.marin@uni-due.de

Axel Hunger and Stefan Werner
University of Duisburg-Essen, Institute for Multimedia and Software Engineering, Duisburg, Germany
{axel.hunger, stefan.werner}@uni-due.de

Abstract—**The use of animated agents in computer based learning environments as a tutoring paradigm can be benefic and increase the learners' motivation. This paper discusses the premises under which synthetic agents can be pedagogically effective as tutors in a collaborative learning environment. This research is a follow-up of our previous investigation on usage of agent technology in distance learning environments [5]. This work will highlight a framework for creating believable synthetic agents.**

Index Terms—**Agent Technology, Emotion Theory, Role Theory, Human Computer Interaction**

I. Introduction

Current distance education systems try to mitigate the difficulties encountered by learners when they try to follow a distance course. Then, it is necessary to take account of these difficulties when distance learning is set up, avoiding insulation and a lost of motivation by learners that are the cause of many giving up [18].

One of the most successful applications of life-like characters is computer based learning environments where synthetic agents can perform a variety of roles especially as tutors or trainers [1], [2], [3]. The use of animated agents in such environments as a tutoring paradigm can be benefic and increase the learners' motivation. Lester [4] investigates the impact of animated agents along the dimensions of motivation and helpfulness in an interactive learning environment.

Several major inconveniences can be noticed in distance education like laboratory experimentation/ practice. Usually, during these experimentations students have to be physically present in the university laboratories. A solution to avoid this disadvantage is virtual experimentation: the experiments are simulated and visualized by means of virtual reality [19].

In local laboratory experiments, students usually work together in groups of two or more. This learning paradigm is often called collaborative learning. One solution for this problem is usage of virtual collaborative environments which bring together users who are geographically distributed but connected via a network. Therefore the students can be trained using the virtual lab concept to work in spatially distributed teams.

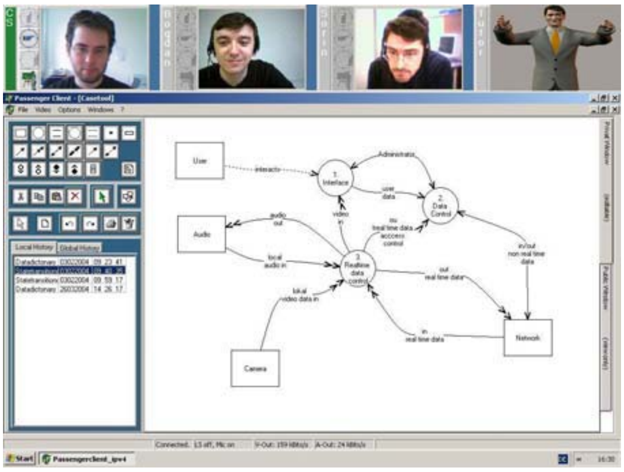


Figure 1 Passenger Learning Environment

Within the past years a synchronous groupware named PASSENGER was developed for this purpose at our university. This groupware (see Figure 1.) is composed from three modules: the communication component which contains video screens of each participant and a cooperation component which allows students to interact together on a common artefact. The participants can be three students and one tutor. A more detailed description will be given in a later section of this article.

This article highlights our efforts to link emotions theory with agent technology in order to show how to motivate distance learners. Our approach aims at highlighting the premises under which pedagogical animated agents can be an effective solution for these inconvenient. This work is a follow-up of our previous investigation on usage of agent technology in distance learning environments [5].

The research on human emotions has a long tradition, both on a cognitive as well as on a physiological basis. As Picard [7] puts it, researchers of this area do not even agree on a definition. For our aim, we find the so-called OCC theory by Ortony, Clore, and Collins [8] the most appropriate one. The third section of this article will provide a brief overview of this theory.

The remainder of this paper is structured as follows: Next section highlights the research objectives and provides an overview of the environment where our prototype performs its actions, 3rd section overviews the theories used in our research,, the following section highlights the proposed framework based on IF THEN rules. The final sections of this article highlight our experimental results and our conclusions of our work.

II. Context And Research Goals

Modern Software Engineering in any case signifies teamwork. The worldwide extension of the data networks and continuing globalization add another component to software engineering: the development of worldwide distributed teams. The use of this forward-looking form in university education could make a special contribution to the way in which students work, and are worked with, in future.

When comparing the professional field of Software Engineering with the Software Engineering education the following aspects have to be taken into consideration:

- working in a team, dividing up the given task into sub-tasks, discussing intermediate results.
- the usage of the new media and communication technologies also requires that students should work in a completely new scenario.

Within the framework of the lab for the lecture "Computer Supported Cooperative Work and Software-Engineering" at University of Duisburg-Essen (UDE), the students are trained using the virtual lab concept to work in spatially distributed teams. The virtual laboratory concept is quite general encompassing a range of technologies and human factors that are necessary for operation in any remote environment, whether remote in distance, time or scale.

The development and the setup of such a lab require resources as follows:

- Technologically mediated communication channel
- Shared workspace for a group
- Personal workspace
- Learning materials/ learning tools

For the special case of the software engineering lab regarded here, no complex problems are to be solved concerning the organization of the practical course or the invitation to a meeting. This could rather be solved by using standard Internet technologies such as HTML (Hyper Text Markup Language) and web-databases. The emphasis lies on the development of a synchronous groupware for the support of the students during a meeting in the context of a Software Engineering lab.

A synchronous groupware named PASSENGER was developed at UDE throughout the last years. This system is a client-/server-based application. The synchronous groupware Passenger is a distributed application for participants in partially not foreseeable places. The system was designed to support collaborative learning activities and especially for the usage in Software Engineering – Education.

Figure 1 shows the user interface of the client software. The system design and development are based on following concepts:

- A floor control [13] to handle the access to common resources, coordinate the course of communication and also to guarantee fairness and to prevent mutual exclusion or blocking.
- A user interface designed to support group awareness [14].
- A whiteboard concept materialized in tools to carry out Software Engineering tasks.

Three essential differences of our tool compared to publicly available solutions can be specified:

a. *Passenger floor control*: the advantages of the developed Floor-Control protocol are to guarantee a defined fairness and to prevent the mutual exclusion and blocking. Thereby, the fairness definition is based on a theoretically equal distribution of the Floor-holding concerning the occurrence. Anyway this type of equal distribution is not forced. Since the defined roles in the group process model are opposed to any kind of equal distribution, the arrangements to guarantee the equal distribution of the Floor were renounced. In particular, the Passenger Floor Control does not limit Floor-holding duration.

b. *A user interface designed to support group awareness*: The design is based on common requirements and the special requirements from the analyzed group behavior. For all design decisions thought has been given to the requirement for measures to increase the group-awareness. Therefore a concept for the positioning and resizing of the communication windows was developed and implemented. Especially solutions for the Floor-Control and the group-awareness were developed during the design of the user interface. The developed Floor-Control is specialized for small, closed groups. It removes the lacks of existing systems concerning the fairness of the Floor assignment and the implemented measures to mutual blockings.

c. *The whiteboard concept materialized in tools to carry out software engineering tasks*. The implemented PASSENGER-CASE-Tool for the software design features its concept for realizing the private and public work area and its process specified support for software engineering. The separated realization between the work and the display area enables the Floor-Holder firstly to simultaneously access the last two design documents. Due to this fact he is simply able to compare the two schemes by switching between the work and display window. The rest of the participants have the same possibility under condition that they transfuse the content of the display field of their work area. The transfer of arbitrary documents to the work area of the Floor-Holder and there to the display area of the others, should not be commented, in order to set the process rights and the access to the speaking channel in a shareable Floor-Control.

The Software-Engineering lab at UDE is conducted as a project setup of student teams, each consisting of four participants: 3 students and one tutor, where the same tutor can be in several virtual teams. That can cause problems in terms of availability if the virtual teams meet at the same time but also if the teams meet at times outside the tutor consultation hours.

Another inconvenient may be tutor's inability to follow-up the collaborative learning process of those student teams. There are two perspectives in the follow-up of collaborative learning activities:

- First one is the follow-up of the collaborative activity: the tutor has to make sure that the learning process proceeds as envisaged (regulation of the learners' activities, according to a fixed schedule, etc).
- Second aspect refers to the individual work of each participant in the collaborative activity. That means that tutor should be able to provide individual help to those students who cannot keep up the progress with their groupmates.

The proposed solution to overcome these two major issues is design and development of a software agent which should play a kind of tutoring role within the Passenger environment. Next sections will present basic design decision and also an architectural overview of the agent.

III. Conceptual Model of the Tutor Agent

Studying agents that behave socially is an exciting research direction. The outcome of such research is relevant both for basic research into the nature of social minds as well as for design and development of systems in application areas that requires agent to show and use aspects of human intelligence like: ability to learn, recognize or express emotions.

Enhancing social roles to pedagogical agents can develop a new social-psychological model for animated tutoring agents similar with a human one. In particular humans can easily adjust their behavior based on their role in a socio-organizational setting, where their actions tend to be driven emotions, attitudes and personality. In this section the components of our model are described and it is explained how those components enable the specification of an artificial character at social levels. Our model includes the following concepts: personality, emotions, roles, and attitudes.

A. Emotion Theory

As previously said, we find the so-called OCC theory by Ortony, Clore, and Collins [8] the most appropriate one for our aim. First, the authors are very concerned with issues dear to the Artificial Intelligence community; for instance, they believe that cooperative problem-solving systems must be able to reason about emotions. Second, it is a very pragmatic theory, based on grouping emotions by their eliciting conditions | events and their consequences, agents1 and their actions, or objects | which best suits a computational implementation.

Further we highlight just a short overview of the structure of the OCC model based on types of emotions. It has three main branches, corresponding to the three ways people react to the world, which we have already mentioned. The first branch is very simple and relates to emotions which are arising from aspects of objects such as liking, disliking, etc. This constitutes the single class in this branch, namely that called attraction which includes the emotions love (if liking) or hate (if disliking). The other main branches are more complex, as they include further dimensions. We first present all items belonging to the second main branch, then those of the third one.

The second branch relates to emotions which are consequences of events. As a reaction to them, one can be pleased, displeased, etc. Further classes are described below.

fortunes-of-others

- person is pleased with event, is focusing on the consequences for the other and thinks the event is desirable for other: *happy-for*
- person is unpleased with event, is focusing on the consequences for the other and thinks the event is desirable for other: *resentment*
- person is pleased with event, is focusing on the consequences for the other and thinks the event is undesirable for other: *gloating*
- person is unpleased with event, is focusing on the consequences for the other and thinks the event is undesirable for other: *pity*

prospect-based

- person is pleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is relevant: *hope* - which can be either confirmed (satisfaction) or not (disappointment)
- person is unpleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is relevant: *fear* - which can be either confirmed (fears confirmed) or not (relief)

well-being

- *joy:* person is pleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is irrelevant
- *distress:* person is unpleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is irrelevant

The third branch, as the first one presented, also has only one class (although it is a richer one), namely the attribution class, comprising the following emotions:

- *pride:* person approves of self:
- *admiration:* person approves of other
- *shame:* person disapproves of self
- *reproach*: person disapproves of other

Also, it is worth mentioning that the prospect-based class was later augmented by Koda [12] to include the element of *surprise* which materializes when *hope* or *fear* are neither confirmed nor discredited. This is important because surprise is an emotion which is normally included among the basic ones.

B. Personality

Personality is also linked to goals: it can be viewed in terms of weights people attribute to terminal goals. For example, a sociable person gives more importance than others to knowing and staying with other people. A selfish person, when the goals of physical safety and others' care are conflicting, chooses to pursue the former, while an altruist pursues the latter. A proud person attributes a high value to self image and autonomy, while a dependent person cares others' image more than self-image.

Personality traits correspond to patterns of behavior and modes of thinking that determine a person's adjustment to the environment [16]. Trait theories assume that an individual's personality profile can be described in terms of psychological traits that influence that person's behavior. In other words, it is assumed that traits predispose people to behave consistently, indifferent to the situation. Thus the personality profile can be used to predict future behaviors.

These traits imply, in a sense, a lower threshold in emotion feeling [8]. For instance, a 'shy' person is keener to feel 'shame', especially in front of unknown people. A 'proud' person, who attributes a high weight to his goals of self-esteem and autonomy, will feel particular pride (will be proud of himself) every time one of these goals is achieved. And, conversely, every time they are threatened (if, say, he is obliged to ask for help), the person will feel the opposite emotion, shame. Thus, a personality trait (proud) is related to attaching a higher weight to a particular goal (self-esteem, autonomy); and, since that goal is particularly important to that kind of person, the person will feel the corresponding emotions (pride or shame) with a higher frequency or intensity.

Opposite to trait theories are social learning theories which assume that a personality is modified by each situation viewed as a learning experience. A person behavior may vary depending on the specific characteristics of the situation in interaction with the individual's appraisal of the situation and reinforcement history [17]. Our agent's personality conceptual model uses both of these theories.

As an example of trait theories in our model let's consider a trait which specifies level of a student activity as a numerical integer value from the interval [-5, 5]. -5 means that the student is very lazy or he or she is not interested in participating in the common learning, 0 that our student is neither lazy nor energetic, and 5 to define that the student is very active, energetic. The default value is 0 and it is assigned by the agent to each student at the beginning of a new Passenger learning session. This value can be incremented or decremented based on student's behavior during the learning process.

Mofat [15] highlights the close relationship between personality and emotion, although they seem very different: emotions are short-lived and focused while personality is stable and global. He also considers mood rather short-live like emotion and not focused like personality.

C. Attitudes

Another important concept included in our model represents the agent's *attitudes* which characterize a relationship between an agent and a student. The attitudes included in our model are based on degree of sympathy and trust. Those attitudes are: like and dislike, trust or don't trust. For example: if an agent likes a student it can offer him a second chance and hints in case that the student made a mistake or doesn't know the answer for a certain question. If the agent trusts a student it will give him the privilege to continue solving the problem in his/her manner even if the agent cannot foresee if the outcome will be the right solution or not.

We represented attitudes as numerical integers with values from the interval [-10,10], where above 0 means *trust* and bellow 0 means *don't trust*. 0 value corresponds to indifferent state: agent has no attitude toward the student. The default value for each parameter is assigned to 0 at the beginning of each semester. How these parameters are modified are shown in the following example: the following scenario can be assumed: during a session the students are asked whether they need help/hints or not after taking more time on a topic than the time allocated. When the students refuse the help and choose a different solution/action than the suggested one, the agent-tutor records this behavior as indirect feedback. If that solution proves to be wrong this is materialized as decrease by 1 of the parameter's value. In case that the solution is good the parameter will be increased by 1. After each session those values are stored in a students' profile database.

The attitude like or dislike is correlated with the trait which defines the activity of a student. The agent *likes* a student with a high level of activity and *dislikes* a lazy student. In a current running Passenger-session if the activity level of a student is already 5, and that student is the current floor holder or he/she has always the tendency to take action/initiative the agent tries to temperate the student by taking the floor from him/her and passes it explicitly to the inactive users. After each session the tutor-agent realizes an individual report for each student based on the student behavior during that session. The attitude of like is materialized as a bonus on the evaluation report: the agent recommends that the student should have a high mark at the final exam. In case of dislike the human tutor is informed of the negative behavior of a student during the learning sessions. The human tutor will investigate the reasons which conducted to such of a negative behavior and he will try to motivate the student to adjust his behavior to a proper one.

D. Role Theory

Equations should be centered in the column. The paragraph description of the line containing the equation should be set for 6 points before and 6 points after. Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). Italicize Roman symbols for quantities and variables, but not Greek symbols. Punctuate equations with commas or periods when they are part of a sentence, as in

We view agents in a virtual society of learners as entities that occupy a social position and perform several roles. Giddens[9] defines social position within a group as the social identity an individual has in a given group or society. Biddle[10] defines roles as those behaviours, characteristic of one or more persons in a context. In our case a role specifies a characteristics pattern of behaviour for the interactions of the agent so that the agent which plays that role behaves in a specific way under certain situations involving other learner(s) or agents.

```
<Role >::= "ROLE"
     <Role ID>
     <Skills>
     <Roleset>
  <Prerequisites>
<Responsibilities>
    "END ROLE"
```

Role ID is used to distinguish a role from other roles. A skill can be defined as the ability to carry out a task at a pre-defined level of competence. In our concept Skills of a role describe the properties (or the abilities) that the agent will need to possess in order to perform successfully the role. Skills should be linked together with roles: if an agent knows what role it has to play then it also knows the skill(s) required to successfully accomplish that role. As far as the procedures (the agent is developed in Borland Delphi, and a class in Delphi has procedures similar with the methods from a Java class) of the Role class there are a group of procedures which associate Role class with skill class:

- *TAddSkill()* - this procedure is used to bind a particular role with a particular skill
- *TRemoveSkill()* – destroys the link created by TAddSkill
- *TGetSkill()* – returns all skills relevant to the role under consideration.

Roleset refers to a set of roles that agent interacts with given this role. Prerequisites of the role refer to the credentials an agent needs in order to occupy the social position under that role. Responsibilities of a role refer to the duties of an agent undertaken within the context of the actual role.

To exemplify the concept in our environment the tutor-agent should have the following roles within a group of students:

- *Interrogator* – poses questions and the students of a collaborative group then provide answers. The questions should provide help for the students to reach a common learning goal.
- *Reviewer* – analyzes the students' answers, including whether it is correct or not.
- *Monitor* – records the answers from all the students and the communications among students during the collaborative learning process.
- *Instructor* – gives individualized instructions and helps those students who cannot keep up with the progress of their group-mates.
- *Group Manager* – has the ability to control the coherence of the group.

Let's take for example the *Reviewer* role: the skills required for this role are: agent should be able to understand the student's answer (natural language processing) then it should analyze whether it is correct or not. An important skill of this role can be considered the ability to display emotions and gestures (animations) to students' answers. Our prototype responds to the students' answers (also to questions) by synthetic speech, facial display and gestures. The facial display of our tutor –agent is limited to a predefined set of animations (e.g. happy, sad). In order to extend the animations for our model we implemented also gestures to express emotions like confused: agent is lifting shoulders or don't recognize the answer/ or question (this skill is also available for other roles – *Instructor*): put a hand to ear.

The *Roleset* for this role can be considered as the set composed from the roles ={*Interrogator, Instructor*}. Of course the *Prerequisites* for this role can be simply deducted: if the students answer to the question posed by agent (during the *Interrogator* role – here is also the link between these 2 roles: *Interrogator* and *Reviewer*).

In the case of the *Monitor* role the agent needs the skill to create a student profile [6] from the students' interaction. The student profile includes student's goals, plans, capabilities, attitudes and knowledge. This profile is based on student's activity during a learning session.

Let's consider a trait called *Activity_Level* which specifies level of a student activity during a session, as a numerical integer value from the interval [-5, 5]. -5 means that the student is very lazy or he or she is not interested in participating in the common learning, 0 that our student is neither lazy nor energetic, and 5 to define that the student is very active, energetic. The default value is 0 and it is assigned by the agent to each student at the beginning of a new Passenger learning session. This value can be incremented or decremented based on student's behaviour during the learning process in the following manner: if the student has the initiative and takes the floor from the agent and performs successfully a sub task his activity level is increased by one, if he/she performs the sub task guided by agent without being able to make own decision the activity level is not modified. The activity level is decreased only when the student refuses to perform its task. Of course the most active student within a learning session can be considered as a leader for the group. In case that the final value of this trait is 0 this can characterize either a lazy or a cautious learner.

The tutor-agent has the responsibility to deliver after each session a report with the students' profile and its own beliefs and conclusions to a human tutor. This profile is based on several parameters like Activity_Level and the human teacher is capable to deduct student's behaviour during a session. Let's assume that after a successful session the levels of activity are 5, 0, -2 for Jan, Mary and Robert. The human teacher can deduct the following things:

Assessing collaboration: it is obvious that Jan dominated all the phases of the activity, Mary did everything under tutor supervision and Robert didn't want to participate.

Assessing contribution: Jan did almost everything while Robert did nothing. Based on the activity level agent can regulate its future teaching strategies and also create a report with its own mental beliefs and actions during a learning session:

Selecting tasks(subtasks) that need focus, a more detailed analysis or explanation: in case the students reached a deadlock the agent should be able to provide more hints, detailed theoretical demonstrations or other similar examples which can help students to overcome the current situation.

Assessing its own interventions: agent should be able to decide whether to intervene or not (students must learn to work in a team and as long as they make progresses do not intervene) and choose the right time when to intervene in the learning process.

Planning the group structure: decide based on the students profile whether the group structure is the optimal one or not. In our example Jan is too active for his group maybe moving him into another group and bringing someone else into his place can motivate also Mary and Roberts to involve more in the learning process.

For the *Group Manager* role the necessary skills are: to monitor the owners of the Passenger Floor Control PFC, to control the entries in the PFC-list[20] and to grant the Floor to the inactive users. Also a requirement of this role is to assure a fair distribution of PFC among participants in a learning session using Passenger. This role and the afferent skills try to solve one of the open problems in the collaborative virtual environments: communication issues among participants.

IV. Theory in Practice

The traditional "Computer Supported Cooperative Work and Software Engineering" lab at UDE is conducted as a project setup of student teams, each consisting of four persons: three students and one tutor, where the same tutor can be in several virtual teams. That can cause problems in terms of tutors' availability if the virtual teams meet at the same time but also if the teams meet at times outside the tutor consultation hours. To make sure that at least a virtual tutor is always available agent technology was used.

We chose a human character instead of an animal in order to impose learners a degree of realism due to the fact that the environment where our prototype acts is virtual environment for distance education system. Our previous work [5], [6], can provide a detailed description on the tutor-agent's architecture and the Passenger environment. To support the needs of task-oriented collaboration, Passenger-Agent Tutor (PAT) includes the following primitive actions:

Speak: PAT can produce a verbal utterance directed at a student or the whole team. PAT has a short range of utterances, all generated from text templates, ranging from a simple greetings like "Hello" or agreements "OK" or "no" to several basic descriptions of domain actions and goals.

Give tutorial feedback: To provide tutorial feedback on a student's action, PAT indicates a student error by shaking his head as he says "no" (see Figure 2) and he indicates a correct action by simply looking at the student and nodding. The motivation for shaking the head is to complement and reinforce the verbal evaluation, and the motivation for the head nod is to provide the least obtrusive possible feedback to the student.

Manipulate an object: To demonstrate domain task steps, PAT can manipulate objects in a variety of ways. Currently, this includes manipulations that can be done by grasping the object (e.g., moving, pulling, inserting, editing, deleting) or pointing at that object (using the Telepointer) to guide the student's attention.

Check the status of the whiteboard: PAT can also demonstrate domain task steps that simply require checking an object (e.g., checking the control unit of a system or a simple checking whether terminator is connected or not).

Point at an object: To draw a student's attention to an object, or connect a verbal referring expression to the object it denotes, PAT can point at the object using the Telepointer[20].

Offer turn: Since our goal is to make PAT's demonstrations interactive, we allow students to interrupt with questions ("What next?" and "Why?"). PAT explicitly offers the Floor Control to them after each demonstration act or when nobody is performing any action in order to achieve the learning goal. PAT's necessary skills to perform actions like this are: to monitor the owners of the Passenger Floor Control PFC, to control the entries in the PFC-list and to grant the Floor to the inactive users. Also PAT must assure a fair distribution of PFC among participants in a learning session using Passenger.

Acknowledge an utterance: When a student or teammate performs an action PAT can choose to explicitly acknowledge his understanding of their utterance by looking at them and nodding (see Figure2).

Attend to action: When someone other than PAT manipulates an object of the whiteboard, PAT automatically shifts his gaze to the object to indicate his awareness. Unlike all the above behaviors, which are chosen deliberately by the cognition module, this behavior is a sort of knee-jerk reaction invoked directly by the perception module.

A. Implementing the proposed framework

We have already motivated the use of the OCC theory in our framework. Additionally, this theory can be translated into a rule-based system which synthesises and generates cognitive-related emotions in an agent. Within this section, we will explain how rules look like in such a system.

Our approach uses the IF-THEN rules: the IF part tests either the desirability (of a consequence of an event), or the praiseworthiness (of an agent's action), or the appeal (of an object). The THEN part sets the potential for generating an emotional state (e.g., a joyful state).

Let $A(s; o; t)$ be the appeal that a student s assigns to the object o at time t, $P_h(s; o; t)$ the potential to generate the state of hate, $G = \langle gv1; : : : ; gvn \rangle$ a combination of global intensity variables, $I_h(s; o; t)$ the intensity of hate, $T_h(s; t)$ a threshold value, and $f_h()$ a function specific to hate. Then, a rule to generate a state of hate looks like:

IF $P_h(s; o; t) > T_h(s; t)$
THEN set $I_h(s; o; t) = P_h(s; o; t) - T_h(s; t)$
ELSE set $I_h(s; o; t) = 0$
This rule is triggered by another one:
IF $A(s; o; t) > 0$
THEN set $P_h(s; o; t) = f_h(A(s; o; t), G)$

Ortony et al. [8] omit many implementation details; a difficult issue, for example, may be to find appropriate functions $f()$ specific to each emotion. However it was not very demanding to come up with such functions in the Passenger learning scenario.

Before showing how to derive the IF-THEN-ELSE rule for different Passenger learning scenarios we need to highlight several notions concerning the learning material [5]. The learning material for one semester is represented by a set of topics, $T = \{T_i | 1 \le i < n\}$, where T_i represents a topic for a Passenger-session and n is the number of topics/lectures of Software Engineering course per semester.

Each topic T can be defined as $T = \{Q_k, P_j \mid 1 \le k \le n, 1 \le k \le m\}$, where Q_k represents a question and P_j represents possible answers by participants to this question, and also agent's plan/reply to each of these answers, n is the number of questions per topic, m is the number of possible plans per question. In other words we can define $P_j = \{M_j, A_j, | 1 \le i \le n\}$, where n is the number of possible answers/solutions for the question/task Q_i, M_j represents students reply to the question, while A_i is agents' response to these replies according to its goals.

There is a unary relationship $Time(t),\ t \in T$, which represents the time allocated for each topic. The value of $Time(t)$ is a number in time units and it differs from a topic to other.

Also, there is a unary relationship $C(t),\ t \in T$, which represents the credits allocated for each topic. The value of $C(t)$ is a number in time units and it differs from a topic to other.

Figure 2 Animations for attending to an action and deny

We defined $C_{student}(t)$ as the relationship which defines the credits won by a *student X* during a t–topic learning session. It is obvious $C(t) = \Sigma\ C_{student}(t)$. It is worth mentioning that for each topic there is a predefined constant value $Y(t)$ (called also minimum value) for the number of accumulated credits. In order to successfully pass a learning session, this relationship must be satisfied $Y(t) \le C_{student}(t)$.

Besides these parameters we use below: $D(s; e; t)$ for the desirability that a student s assigns to event e at time t, $W(s; a; t)$ for the praiseworthiness that a student s assigns to ask for help to agent a at time t, $L = \langle lv1; : : : ; lvn \rangle$ a combination of local intensity variables, x is a pre-defined constant on the number of students (in our case x=3 which represents the number of students/session), y is a pre-defined constant on a student's number of accumulated credits, T_0 a predefined constant on allocated time for a student to perform a task, and ε is a fixed increment to the values returned by the emotion-specific functions.

PAT will be subject to five independent emotions: joy, distress, pity, boredom and fear. These are relatively easy to portray using the IF-THEN-ELSE rules and are sufficiently distinct from each other.

- Rules for joy:
  IF $D(s; e; t) > 0$
  THEN set $P_j(s; e; t) = f_j\ (D(s; e; t)\ G; L)$
  f_j returns ($T_j(s; t)$+ε) IF student s has collected at least y credits AND $Time(t) < T_0$
  IF $P_j(s; e; t) > T_j(s; t)$
  THEN set $I_j(s; e; t) = P_j(s; e; t) - T_j(s; t)$
  ELSE set $I_j(s; e; t) = 0$

- Rules for distress:
  IF $D(s; e; t) < 0$
  THEN set $P_d(s; e; t) = f_d(D(s; e; t); G; L)$
  f_d returns ($T_d(s; t)$+ε) IF student s has not collected at least y credits AND for at least x-2 team-mates (other students) agent feels distress AND $Time(t) < T_0$
  IF $P_d(s; e; t) > T_d(s; t)$
  THEN set $I_d(s; e; t) = P_d(s; e; t) - T_d(s; t)$
  ELSE set $I_d(s; e; t) = 0$

- Rules for pity:
  IF $D(s; e; t) < 0$
  THEN set $P_i(s; e; t) = f_i(D(s; e; t);G)$
  f_i returns $(T_i(s;\ t)+\varepsilon)$ IF student s has not collected at least y credits AND $Time(t) > T_0$
  IF $P_i(s; e; t) > T_i(s; t)$
  THEN set $I_i(s; e; t) = P_i(s; e; t) - T_i(s; t)$
  ELSE set $I_i (s; e; t) = 0$

- Rules for boredom:
  IF $W(s; a; t) > 0$
  THEN set $P_b(s; a; t) = f_b(W(s; a; t);L)$
  f_b returns $(T_b(s; t) + \varepsilon)$ IF for at least x-1 students agent has identical type of emotion AND $Time(t) < T_0$
  IF $P_b(s; a; t) > T_b(s; t)$
  THEN set $I_b(s; a; t) = P_b(s; a; t) - T_b(s; t)$
  ELSE set $I_b(s; a; t) = 0$

- Rules for fear:
  IF $D(s; e; t) < 0$
  THEN set $P_f(s; e; t) = f_f(D(s; e; t); L)$
  f_f returns $(T_f(s;\ t)+\ \varepsilon)$ IF student s has not collected at least y credits AND $Time(t) < T_0$
  IF $P_f(s; e; t) > T_f(s; t)$
  THEN set $I_f(s; e; t) = P_f(s; e; t) - T_f(s; t)$
  ELSE set $I_f(s; e; t) = 0$

Similarly, PAT feels pity for the student who couldn't reach the number of credits. Boredom is felt when for at least a certain number of students PAT has the same feelings for example fear and the students do not perform in the PAT's expected way (e.g. this turns PAT bored and it tries to provoke its students by offering turns, providing more hints or helpful questions, making small theory demonstrations). Finally, fear is experienced when PAT observes a student whose credits are so few and the allocated time may finish soon.

One can easily notice that these rules are domain-oriented. However Picard [7] argues that many emotions do have complicated rules, therefore this work can be used as a starting point by anyone who wants to generalize the rules, at least for similar scenarios. Within the next section we will exemplify how these rules work.

Figure 3 Emotions' display: distress and joy

B. System Architecture

Agent's goals might be locally achievable which means that tutoring material can be retrieved from a local knowledge database, or require interaction with another tutor-agent from a different session or a human tutor in case of a word pattern for which it cannot find it in its knowledge database.

When students pose a question, the tutor-agent can provide immediate proper answers or hints by matching a regular pattern in its knowledge database. A word pattern refers to a unique collection of words. A regular pattern is a word pattern that appears with high frequency and reflects the relationships within or among topics. Regular patterns are the skeleton of topics and the agent uses these regular patterns to provide students proper answers or future steps in modeling.

Here is an example to explain this approach. The following scenario can occur: one of the students doesn't know how to draw a control memory thus he/she asks the tutor-agent the following question: "*How (1) can I draw (2) a control memory (3)?*" where *(1)(2)* and *(3) ~How… draw… control memory…~* represent a pattern example. After recognizing a pattern agent will search its knowledge database for a proper answer and will provide this answer to the student. However words like "how" or "where" cannot be regarded as regular patterns because they lack of the essential structure of a question.

Therefore in order to realize this pattern algorithm three functions were implemented:

- *Pattern recognizer:* recognize a pattern
- *Pattern miner*: search the recognized pattern in the knowledge database
- *Pattern adder:* adds a new pattern to the knowledge database

The system consists of a number of tutor-agents, representing workflow participants, and a coordinator agent which provides directory service to tutoring-agents. All agents are logically dispersed and arranged in an interconnected network of Passenger platforms.

Tutoring-agents register their services with the coordinator agent and when a non-local activity like one tutor agent needs help for an unknown pattern/situation, needs to be performed the coordinator can be looked up to find the appropriate provider in our case can be another tutor agent or a human one.

To make tutor-agent's design process easy and accessible, we proposed a framework where the agent is composed of a set of categories of components (see figure 4).A message handler composes the communication heart of the tutor-agent. Messages may be transmitted either to the session or through the interconnection network directly in case help is required. The message handler processes incoming messages and processes appropriate logic to handle them to the core engine.

The core engine is responsible for executing one or more learning tasks. Also this component composes the agent decision system, the brain of the agent – agent can choose and plan its future actions according to its goal. Here are implemented pattern recognition and learning algorithms. This component realizes also the evaluation of student – student profile database, based on his behavior during a learning session: it is the one who decided to increase or decrease the parameters (e.g. activity level) which characterize the students' behavior.

The global knowledge database contains information about agents' beliefs or plans while the coaching knowledge contains information concerning the learning material which is represented by topics.

Agent interface (see Figure 2, Figure3) is an animated cartoon with human like gestures. Our agent responds to the students' questions/actions by synthetic speech, facial display and gestures. We choose to design and implement our own animated agent instead of using Microsoft Agent to ensure Passenger platform independence and extensibility. We choose Borland Delphi environment to realize the Passenger environment and also for our prototype implementation.

Navigational component: that is responsible for tutor-agent's travel itinerary when it needs to communicate with other tutors (humans or agents) in order to receive help for accomplishing its task. In order to realize the agent's mobility we implemented the following algorithm:

```
get_hosts_list(CAG);
search_hosts(n, P_j, A_k)
  begin
for i= 0 to n-1, i≠k do
 if KL.A_k > KL.A_i then jump to  next
host;
   else if ask_help_pattern (A_i,
P_j)=NULL then jump to next host;
       else return(ask_help_pattern
(A_i, P_j)); break;
return(NULL);
end  search_hosts;
if search_hosts(n, P_j, A_k)=NULL then
ask_human(P_j);
```

where CAG represents the coordinator agent, n is the number of active hosts (we define as an active host a current running Passenger learning session), A_k the tutor agent which needs help, P_j the unknown pattern, and $KL.A_k$ the knowledge level of A_k. In case that agent A_k needs help for the unknown pattern P_j, it first connects to coordinator agent and gets the list of the current running Passenger sessions.

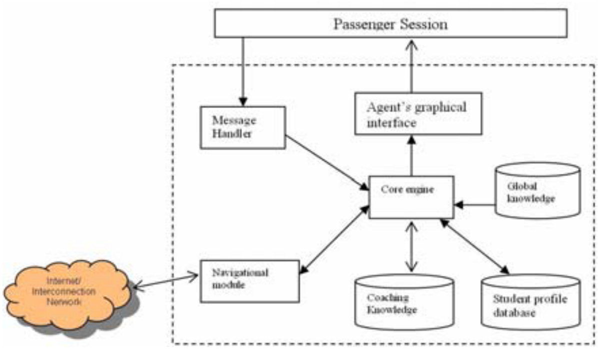


Figure 4 Tutor-agent's architecture

For each agent we defined as a knowledge level the binary relationship between very two different topics T_i and T_j called *Precedence(T_i,T_j)* which also highlights the fact that T_i is a prerequisite for T_j, where $i<j$. Thus if between agent A_k and agent A_j the following statement $KL.A_k > KL.A_i$ is true that means that agent A_i is teaching a topic T_i, which is a prerequisite for T_k, topic of A_k. In other words, A_i cannot provide answers to A_k, therefore A_k has to move to the next host. In case that the agent A_k cannot find adequate help from the other tutor-agents it has to communicate with a human-tutor and get from him/her proper answer.

C. Experimental Results

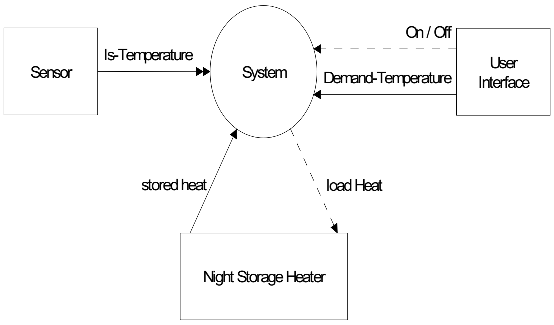
The Software Engineering lab at UDE is conducted as a project setup of student teams, each consisting of four students. During the semester these teams will experience the entire life-cycle of software engineering. The students start with a requirement analysis following the Ward & Mellor [24] approach during the modeling phase. The given problem for the practical training is chosen in such a way, that it cannot be solved by one student on its own. Therefore, the project teams have to divide up the problem amongst each other. The teams meet once a week at a certain time for two hours in a computer lab at the university. During this time, tutors are available. We first conducted a series of tests on simplified scenarios in order to prove our theory.

Here is a sample of a practical exercise: "Given is a night storage heating system. This is a heating system, which uses electrical power to store heat at night, because charge is cheaper. The system works like this: The user defines the status of the heating system: besides the decision, if the system is on or off, the user sets a temperature to the system. This temperature is compared with the current temperature stored in the night heating system and the room temperature. To measure the room temperature a separate sensor is given. Design the system according to the Ward & Mellor".

In this exercise a night storage heating system is used as an example. Ward and Mellor introduced added some notational nuances to handle interrupts and control flows to the Data Flow Diagram; they also introduced the notion of state-transition diagrams for modeling the time-dependent behavior of such systems.

We provide part of the solution in order to be able to explain our theory in practice. There are two different procedures for the structured system design: Top Down design and Bottom Up design.

Topic of this exercise is the design method by Ward & Mellor, a "Top Down design" which is divided into two subjects: Essential Model and Implementation Model. The Essential Model describes the demanded behavior of the system and it is divided into two subjects: the Environment Model, and the Behavioral Model. The Environment Model has three sub-topics: Context Diagram, Information Model and Data Dictionary. The Context Diagram (see Figure5) requires two additional refinement diagrams: one for the heat control and the other for temperature control.



F

Figure 5. Exercise partial solution: Context Diagram

The Behavioural Model is the second part of the Essential Model. It describes the response behaviour from the environment of the system and it consists of State Transition Diagram, the Pre-Post-Condition and the extended Data Dictionary.

In order to successfully pass this practical training the students have to realize the Essential Model for the required system. The given credits for this task and its subtasks are as follows: choosing the right type of design method (and justifying their choice) 3 credits, 1 credit for each definition of the Essential Model, Environment Model and Behavioural Model, 15 credits for the design of the Context Diagram with the 2 additional refinements, 10 credits for the State Transition Diagram and 5 credits for each realization of the Information Model, Data Dictionary, extended Data Dictionary like also for establishing the Pre-Post-Condition. The total amount of credits is 51 and the minimum required number of credits of each student is 14 (if all the students get only their minimum credits and not more they can fulfill more than 80% of the required task which is enough to pass to the next topic). Each subtask must be accomplished in a pre-defined period of time in order to get the credits.

For instance, to check if PAT feels pity for student_1, it tests whether student_1 has not collected at least y = 14 credits AND the given period of time is not over. If student_2 has collected 18 credits and the time allocated to the task is not over PAT feels joy for him.

These emotions depend of course also on the values of the potentials and thresholds defined in the rules associated with them, which are also task dependent. Within this section we tried to provide a simplified example of our theory applied in the synchronous learning environment called: Passenger.

V. RELATED WORK

The application of agents in the educational sector comes about mainly in the form of personal assistants, user guides, alternative help systems, dynamic distributed system architectures, human-system mediators and others. As a result of all of the changes that have taken place in the educational system, one now sees the increasing emergence of complex and dynamic educational infrastructure that needs to be efficiently managed.

According to Aroyo and Kommers [21], agents can influence different aspects in educational systems. They supply new educational paradigms, support theories and can be very helpful both for learners and for teachers in the task of computer-aided learning.

Lees and Ye [22] believe that the application of the agent paradigm to CSCW potentially can exchange information more fluid among the participants of groupware systems (as decision-making systems), help in control of the process flows and also supply groupware interfaces. These ideas also are applicable to other domains, such as is the case of interactive learning.

For McCalla & all [26], learner models may have a variety of purposes depending upon the type of knowledge that needs to be stored and processed. For them, the computation of all of the learner (sub-)models of an environment can be computationally expensive and not always necessary. In the work cited four purposes are presented for a model: reflection, validation, matchmakers and negotiation.

Guizzardi et al. [23] investigate the nonhierarchical relationship between teachers and students in an environment where everyone can teach and learn. They gather two perspectives: one from an implementation point of view and the other one from a software engineering perspective and propose an agent-based system to support extra-class discussions between students and teachers.

For Kay [25], there are several problems from the learners' point of view. One is the increase in the power of choice and control over the model. This could increase the learners' workloads or even turn into a distraction. In this case, the learners should take advantage of the moments such as the end of a course or a topic to evaluate and reflect upon their participation and the learning process. Another potential problem is incorrect data being supplied by the learners. The solution adopted in this work for that problem was to store the type of information learners are providing and the type the environment extracts.

Mustapha [27] considers the roles of an agent in an educational environment to be the following: to monitor, control and catalyze the social knowledge building among the community of learning. Social knowledge is considered to be derivable from socializing oneself with the peers, communities through formal or informal discussion, chitchat or social gatherings.

Role theory and agent technology are also investigated by Depke [11] in his work, where he discusses a concept of roles and how it can be used to improve agent modelling processes. According to Depke roles are "a means to ease systematic transition between different stages of the modelling process: e.g. *analysis* to *design*".

Prendinger and Ishizuka [28] present the use of animated conversational agents in a pedagogical environment where Japanese native speakers practice English. Their approach identifies social role awareness as an important concept for the agents. User-agent interactions are materialized as role-playing interactions.

In the literature, there are a few authors who have written on the use of agents for distance learning. For example, Santos and Rodriguez[29] discussed an agent architecture that provides knowledge-based facilities for distance education. Their approach is to take advantage of recent standardization activities to integrate information from different sources (in standardized formats) in order to improve the learning process, both detecting learner problems and recommending new contents that can be more suitable for the learner's skills and abilities. They accomplish this by using a suite of different agents, such as a "learning content agent," a "catalog agent," a "competency agent," a "certification agent," a "profile agent," and a "learner agent."

VI. EVALUATION

The intended evaluation study for this prototype concerns two levels:

- Usefulness level: the usefulness of the agent facilities within Passenger groupware needs to be evaluated by human teachers.
- User friendliness level: this level highlights how the agent was accepted by students.

Several experiments took place in the local area network of our institute. Only the second part of the evaluation study was conducted among 25 first year Master-students. Each session consisted of three students and one tutor (human or agent). The student experienced the traditional lab with the human tutor and also with the agent feature of the Passenger system. After these experiments, students had to answer to questionnaires files. A sample of questions concerning the second level that were asked to the students is the following:

1. *Do you consider the application attractive? If yes, what did you like about it?*
2. *Do you think that the "agent" features prevented you from understanding the educational process better?*
3. *Do you prefer the agent tutor instead of the human tutor? Please justify your answer.*

Based on these questionnaires several statistics could be made. Some results concerning the agent integration and acceptance are shown in the Figure 4.

Although the number of participants in the evaluation test was rather small for a quantitative evaluation, the trends seem to be unambiguous. We plan to realize the full-evaluation test including an evaluation result for the first level and also to increase the number of student participants

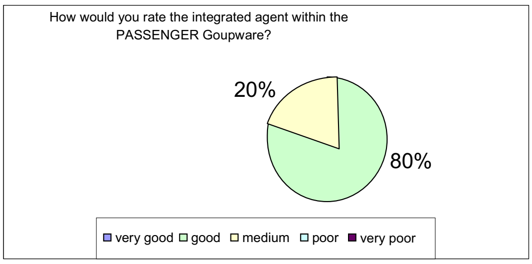


Figure 6. Acceptance of Agent Tutor

VII. CONCLUSION

Enhancing emotional behaviour to animated pedagogical agents can develop a new social-psychological model for life-like characters similar with a human one. In particular humans can easily adjust their behaviour based on their role in a socio-organizational setting, where their actions tend to be driven emotions, attitudes and personality.

This paper goal was to show how to corroborate emotions theory with agent technology in order to support collaborative learning in distributed environments. The aim of this research is to provide the first steps to define a method for creating a believable tutor agent which can partially replace human-teachers and assist and motivate students in their process of learning.

The outcome of such research is relevant both for basic research into the nature of social minds as well as for design and development of systems in application areas that requires agent to show and use aspects of human intelligence like: ability to learn, recognize or express emotions.

REFERENCES

[1] C. Conati "Probabilistic assessment of user's emotions in educational games", *Applied Artificial Intelligence*, vol.16, 2002, pp. 555-575.

[2] F. de Rosis, B. de Carolis, and S. Pizzulito. "Software documentation with animated agents", *Proceedings 5th ERCIM Workshop on User Interfaces For All*, 1999.

[3] Kitamura, Y., Tsujimoto, H., Yamada, T. and Yamamoto T., "Multiple character – agents interface: an information integration platform where multiple agents and human users collaborate", *Proceedings of AAMAS-02*, New York, 2002.

[4] J.C. Lester, S.A. Converse, S.E. Kahler, S.T. Barlow, B.A. Stone, and R.S. Bhogal, "The Persona effect: Affective impact of animated pedagogical agents", In *Proceedings of CHI-97*, 1997, pp. 359-366.

[5] B. Marin, A. Hunger, S. Werner, "A Framework for Designing Emotional Agents as Tutoring Entities",*Proc. of IEEE 2006 International Symposium on Applications and the Internet*, Phoenix, USA 2006.

[6] B. Marin, A. Hunger, S. Werner, S. Meila and C. Schuetz, "Roles of an Intelligent Tutor Agent in a Virtual Society". *Proc. of IEEE 2005 International Symposium on Applications and the Internet*, Trento, Italy, 2005.

[7] R. W. Picard. *Affective Computing*. The MIT Press, Cambridge, MA, 1997.

[8] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, UK, 1988.

[9] Giddens, A. *Sociology*, Polity Press, 3rd edition, ISBN: 0745618030, London, UK, 1997, pp.585.

[10] Biddle, B.J. and Thomas, E.J. *Role Theory: Concepts and Research,* K.E. Kriger Publishing Company, New York, 1979.

[11] R. Depke, R. Heckel and J.M. Kuster "Improving the agent-oriented modelling by roles", *Proceedings of the 5th Int. Conf. on Autonomous Agents,* Montreal, Canada, 2001, pp.640-647.

[12] T. Koda *Agents with faces: a study on the effect of the personification of software agents,* Master thesis, MIT Media Lab, Cambridge, MA, 1996.

[13] H.-P. Dommel and J.J. Garcia-Luna Aceves, "Group coordination support for synchronous Internet collaboration", *IEEE Internet Computing*, March-April 1999, pp.74-80.

[14] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces", In J. Turnier and R. Kraut, (Eds.) *Proc. Of CSCW'92- Sharing Perspectives*, ACM Press Toronto Canada, pp. 107-114.

[15] D. Mofat, "Personality parameters and programs", In R. Trappl and P. Petta, editors, *Creating Personalities for Synthetic Actors*, Springer, 1997, pp. 120-165.

[16] R.L. Atkinson, R.C. Atkinson and E.R. Hilgard, "*Introduction to Psychology*", Harcourt Brace Jovanovich Inc. 1983.

[17] A. Bandura, "*Social Learning Theory*", Prentice-Hall, Englewood Cliffs, 1977.

[18] L. Rene-Boullier, "Pedagogical coordination: a global vision of the personalized accompaniment of DESS DICIT", *Revue Sticef*, vol.10, 2003

[19] C. Schmidt, "A remote laboratory using virtual reality on the web", *Simulation*, vol.73, no.1, pp.13-21, 1999.

[20] B. Marin, A. Hunger, S. Werner, S. Meila and C. Schuetz, "A synchronous groupware tool to conduct a spatially distributed collaborative learning process", *Proc. of the 5th Int. Conf. on Information Technology Based Higher Education and Training*, Istanbul, 2004, pp.269-274.

[21] L. Aroyo, and P. Kommers, "Preface - Intelligent Agents for Educational Computer-Aided Systems", *Journal of Interactive Learning Research*, Vol.10 (3/4), 1999, pp. 235-242.

[22] Lees, B., Ye, Y.: "Preface of the Proceedings of ASCW01 – Workshop of Agent-Supported Cooperative Work", *The 5th International Conference on Autonomous Agents*, Montreal, Canada, 2001.

[23] R.S.S. Guizzardi, G. Wagner and L. Aroyo, "Agent-oriented modeling for collaborative learning environments: a peer-to-peer helpdesk case study", XIII-Brazilian Symposium on Computers in Education, November 2002.

[24] Ward, P.T. and Mellor, S.J., *"Structured development for real time systems"*, Prentice-Hall International, 1985.

[25] Kay, J, *Learner Control. User modeling and User Adapted Interaction*, 11, Kluwer Academic Publishers, Netherlands 2001, pp. 111-127.

[26] McCalla, G., Vassileva, J., Greer, J., Bull, S.: "Active Learner Modeling", In: *Gautier, Frasson & Vanlehn (Ed.). Proceedings of ITS'2000*, Springer LNCS 1839, 2000, pp. 53- 62.

[27] S.M.F.D. Syed-Mustapha, "Agent mediated for intelligent conversational channel for social knowledge-building in educational environment", *Proc. of the 5th Int. Conf. on Information Technology Based Higher Education and Training*, Istanbul, 31 May-2 June 2004, pp.533-538.

[28] H. Prendinger and M. Ishizuka, "Let's Talk! Socially Intelligent Agents for Language Conversation Training", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol.31, no. 5, September 2001, pp. 465-471.

[29] Santos, J. M., & Rodriguez, J. S. Towards an agent architecture to provide knowledge-based facilities for distance education. *Proceedings, 24th Int. Conf. Information Technology Interfaces (ITI2002)*, Cavtat, Croatia, June, 2002.

Bogdan Florin Marin received his Dipl. Eng. degree in computer science and communication engineering from University of Craiova (Romania) in 2002.

In 2003 he joined University of Duisburg Essen as a research assistant. His research interests include various aspects of agent technology, distance education, virtual collaborative learning environments.

Axel Hunger received his PhD from RWTH Aachen Elektrotehnik, in 1982. From 1979 till 1987 he was the leader of a research group "Testing and Simulation of Digital Circuits". In 1988 he was appointed as Professor for Technical Informatics at University of Duisburg.

His spheres of activity cover the development of Multimedia tools for teaching and learning platforms, curriculum development for international courses of studies and engineering education in the international context.

Stefan Werner received his PhD from University of Duisburg in 2002.

His research interests focuses in the area of the Computer-Supported Collaborative Work (CSCW), Groupware in the Internet and he is the designer and developer of the synchronous Groupware of PASSENGER.