VirtualHuman—Dialogic and Affective Interaction with Virtual Characters

Norbert Reithinger, Patrick Gebhard, Markus Löckelt,
Alassane Ndiaye, Norbert Pfleger, Martin Klesen
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
firstname.lastname@dfki.de

ABSTRACT

Natural multimodal interaction with realistic virtual characters provides rich opportunities for entertainment and education. In this paper we present the current VirtualHuman demonstrator system. It provides a knowledge-based framework to create interactive applications in a multi-user, multi-agent setting. The behavior of the virtual humans and objects in the 3D environment is controlled by interacting affective conversational dialogue engines. An elaborate model of affective behavior adds natural emotional reactions and presence of the virtual humans. Actions are defined in a XML-based markup language that supports the incremental specification of synchronized multimodal output. The system was successfully demonstrated during CeBIT 2006.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Artificial, augmented, and virtual realities

General Terms

Algorithms & Design

Keywords

Speech and conversational interfaces, Multimodal input and output interfaces, AI techniques & adaptive multimodal interfaces Mobile, tangible & virtual/augmented multimodal interfaces

1. INTRODUCTION

Virtual humans are now used in a variety of applications, including education and training, therapy, marketing and entertainment. However, only a few systems have taken up the challenge to allow a direct face-to-face natural language interaction between a user and two or more virtual humans. This endeavor involves the integration of a whole range of technologies, most notably speech recognition and synthesis, natural language processing, action planning, and human figure animation [7]. Examples for systems that explicitly address these problems in realizing a multi-party scenario are the Mission Rehearsal Excercise project [21] and the one-act interactive drama Façade [12].

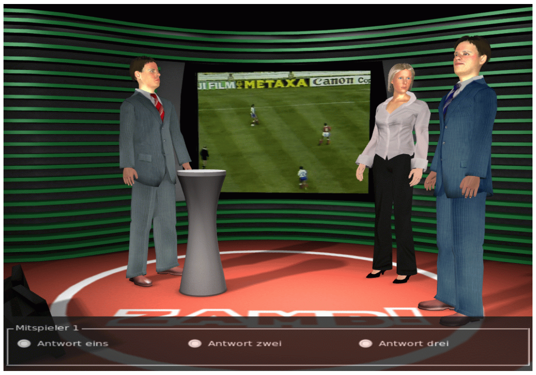


Figure 1: The studio of game phase 1 with three virtual characters

The interaction between the human user(s) and the virtual humans requires innovative solutions to realize a natural real-time communicative behavior. In the collaborative research project VirtualHuman[1] various institutions in Germany implemented a generic, component based framework for this type of interactions. In order to create a realistic interaction experience virtual characters must express themselves naturally through language, movements, gaze, emotions and turn-taking behavior. They also must react naturally to actions of the users, like spoken input or changes in the virtual environment through activities of the human or virtual dialog partners.

In the context of the football World Cup in Germany 2006, we realized a game-show like setup as demonstration environment for our technology. Figure 1 shows the studio of ZAMB with the moderator (to the left) and two football experts, Mrs. Herzog and Mr. Kaiser, who play with two humans participants. The game has two phases: in the first phase, the two human players prove their football knowledge

ICMI'06, November 2–4, 2006, Banff, Alberta, Canada.
Copyright 2006 ACM 1-59593-541-X/06/0011 ...$5.00.

[1] See `http://www.virtualhuman.de/`.

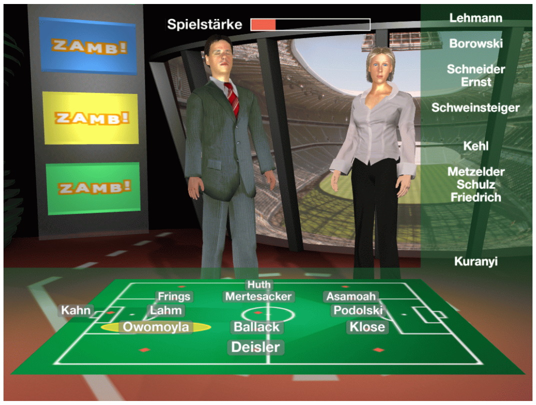


Figure 2: The studio of game phase 2 with two virtual characters

on video scenes from earlier championships. Users can ask the experts for advice and receive comments on their decisions. The winner of the first phase makes it to the second one (see fig. 2) where she can select the German team for a given opponent team. Finally, the experts evaluate the team and comment on the strengths and weaknesses.

The playing field is depicted as a lying rectangle for space reasons. This introduces a first ambiguity with regard to the reference system that is used during the game. It is customary to call player positions as seen from the position of the goal. That means that a "left defender" for the team in the left half of the playing field will actually be positioned in the *upper* half of the graphics on the screen. Likewise, to move a defender "right" could mean either to put him in a midfield position, or downwards on the screen.

Moves can be under-specified in that when the contestant assigns a player to the midfield without specifying the side. The game logic will look for an unoccupied position in the midfield (if there is one) and put the player there. The contestant also has the option to ask the expert or the moderator for advice. The expert and the moderator also frequently give comments on their own. Below we give an example of a dialog translated from German:

(1) Moderator: *Ok, let's get started.*
(2) User: *Put Oliver Kahn into the goal.*
(3) Expert Herzog: [nods] *That's an excellent move!*
(4) Moderator: [nods] *Great, Kahn in the goal position.*
(5) User: *Miss Herzog, give me a hint!*
(6) Expert Herzog: [smiles] *I would definitely put Ballack into the central midfield.*
(7) User: *Ok, let's do that.*
(8) Expert Herzog: [smiles] (nods) *You won't regret this move.*
(9) Moderator: (nods) *Great, Ballack as central midfielder.*
(10) User: ... [hesitates]
(11) Moderator: [encouraging gesture] *Don't be shy!*
(12) User: *Hhm, put Metzelder to the left of Ballack.* [. . .]

The interaction mainly uses spoken language, but the human players can also use a pointing device to interact. The high flexibility and the personalized interaction requires very detailed models on various levels:

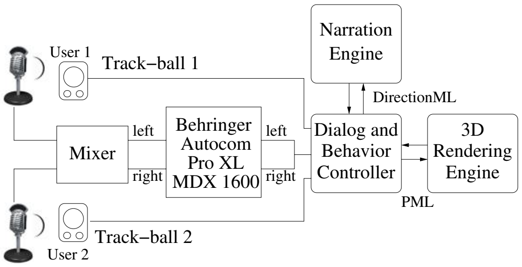


Figure 3: Basic Architecture of the VIRTUALHUMAN

2. SYSTEM ARCHITECTURE

The dialog between n-partners cannot be scripted in advance. Therefore we need a common knowledge base, describing the world, i.e., the studio and the football related facts. Flexible control and reaction mechanisms enable true real-time interactivity: for each human and virtual interaction partner we have a Conversational Dialogue Engine (CDE) that interprets and controls the interaction of the dialog participant, and generates reaction in real-time in the case of the virtual characters

Realistic believable behavior in virtual characters requires also affective control of verbal and nonverbal behavior, e.g., facial expressions. We employ a layered model of affect that, for each participant, analyzes the interactions and generates active emotions.

Natural body gestures, including turn-taking signals, are essential to express the inner state of (real or virtual) humans and to control the dialog flow, e.g., through head or eye movements. Realistic virtual characters must be empowered to use this interaction register to establish their role in communications.

Fig. 3 illustrates the architecture of VIRTUALHUMAN with an emphasis on the components described in this article. The narration engine which has control of the overall story, the 3D player, the characters and their gestures are contributed by project partners. The descriptions of the gestures with information about shape and duration are stored in a gesticon. The ontology, realized using the Protege 3.1 tool, contains all declarative knowledge, including the dialog knowledge, the soccer related data, and the game structure. For speech recognition we use the open-source ISIP system, for speech synthesis the most recent versions of NUANCE's high quality voices.

The components communicate over shared blackboards they can subscribe to for publishing and reading data. The whole interaction between the components is based on the exchange of structured data through messages. The format and content of these messages is defined by three XML-based markup languages.

- DirectionML: The Narration Engine controls the global session development and the sequence of dialogs. Its output for the Conversational Dialog Engines are directions encoded in the direction modeling language.
- PML: The Player Markup Language allows to specify the properties and the behavior of characters and objects in a 3D virtual environment. It supports the incremental specification of synchronized multimodal output using both qualitative and quantitative temporal constraints.

- AffectML: The affect modeling language is used for representing the computed affective characteristics of the virtual agents for processing and for conveying this information to the subsequent modules that control the virtual characters behavior.

The blackboard approach with declarative interface languages facilitates the integration of the various components. Even though they are realized in different programming languages (mainly Java and C++) on different platforms (Linux and Windows XP) their integration was straightforward. Also, since, e.g., speech synthesis and the 3D player are computationally expensive, this architecture allows us to distribute the workload on different computers, thus enabling real-time processing. This is an important prerequisite for the life-like experience of human users.

3. PLAYER MARKUP LANGUAGE

The Player Markup Language (PML) serves as an interface language between the CDEs, the affect module, the action encoder, and the 3D player (see Fig. 3). It allows to specify the properties and the behavior of characters and objects in a 3D virtual environment in an incremental way and on different levels of abstraction. PML is implemented in XML and based on the Rich Representation Language (RRL) developed in the NECA project [19]. Both languages focus on the specification of verbal and non-verbal behaviors of characters in multi-party dialogs. PML distinguishes three types of documents:

- **PML definitions** are used to specify the properties of objects and characters in the 3D environment, e. g., their initial position and orientation, the acoustic parameters of the synthetic voices (pitch baseline, volume, speed), the available animations, their default durations, and the phoneme-viseme mapping to be used. New behaviors (e.g. different idle behaviors) for each character can be defined as combinations of available animations. Definitions can also be used to specify graphical elements (e.g. on-screen menus) and to create references to multimedia objects (audio files, images, videos) that will be used in the scenario.

- **PML actions** are used to specify the behavior of all characters and objects in a 3D environment. The PML supports the incremental specification of synchronized multimodal output (e.g. postures, gestures, facial animations, speech) using both qualitative and quantitative temporal constraints. In a first step actions are synchronized by specifying temporal relations (e.g. before, overlaps, during). In a second step these qualitative constraints are resolved by the action encoder (see Sect. 5) which computes the start time and duration for each action. The exact timing information is represented in a SMIL (Synchronized Multimedia Integration Language) compliant syntax (see `http://www.w3.org/AudioVideo/`).

- **PML messages** are used by the 3D character player to inform other modules about the execution state (e.g. started, failed, finished) of actions. This information is used to synchronize the behavior of characters and objects *across* different sets of actions and to return error statements in case something goes wrong. Messages are also used to inform the CDEs about user actions (e.g. the user has selected a menu entry).

The PML provides both high level abstract concepts (e.g. gestures, complexions, emotions) and detailed, technical information required for the character- and player-related realization of those concepts, for example, animation parameters and exact timing information. The mapping between these two sets of elements is done at runtime by the action encoder as described in Section 5.

4. PARTICIPATING IN CONVERSATIONS: CONVERSATIONAL DIALOG ENGINES

The contributions of human and virtual dialog participants are realized in the system by independent, autonomous entities called *Conversational Dialog Engines* (CDEs). A CDE representing a human user has to process her multimodal input and convert it to the ontological representation. The task of the CDE for a virtual character comprises processing contributions the character can perceive, deliberating about it with respect to the goals of the character, deciding on a course of action, and finally executing its intended actions. Accordingly, there are two different types of CDEs (see Fig. 4): *User CDEs* and *character CDEs* that differ in how they operate and what sub-components they contain. A character CDE contains a fusion and discourse engine (FADE), an affect engine, an action manager, and a multimodal generation component. A user CDE contains multi-modal recognition and interpretation modules (speech recognizer, NL understanding, gesture recognizer, and gesture analyzer) and a FADE.

4.1 Representing Meaning

All internal knowledge used by the CDEs is represented by means of an ontology, which provides a formal taxonomy of all objects and events that exist in the restricted world of the VirtualHuman system. Each concept comprises a set of slots that define its sub-components or attributes.

The top-level of the ontology is based on that introduced in [20]. Key to this hierarchy is the type *Events* that serves as a top-level type for all things that take place at particular times and places. The most important sub-concepts of this type are *PhysicalObjects* and *Processes. PhysicalObjects* serve as super-concepts for domain specific concepts like, *FootballPlayer* (which is grouped under *Person*, *Living* and, eventually, *PhysicalObject*). *Processes*, in contrast, denote events that either simply occur in the world or are conducted by a character (see section 4.3). A *FootballQuiz*, for example, is a sub-concept of an *CommunicativeProcess.*

Beside the model of physical objects and processes, the ontology also incorporates a number of abstract concepts that represent meta-information. A branch of the taxonomy under *AbstractObjects*, for example, specifies the sub-classes of the concept *DialogActs.* These concepts have in common that they are used to describe the content of contributions on different levels of complexity (i. e., the semantic content, accompanying non-verbal behavior, or the spoken utterance). However, each individual sub-concept has its own specific semantics (e. g., *Greeting*, *Request* and *Statement*). Another example for sub-concepts of *AbstractObjects* are *Gestures* defining all relevant aspects (e. .g, physical form, meaning,

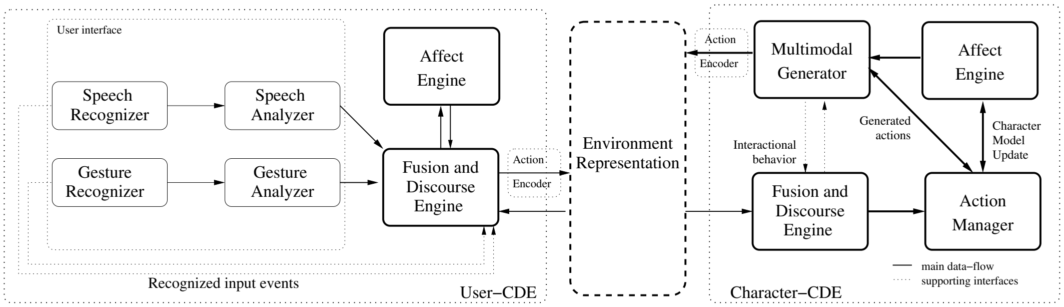


Figure 4: Architecture of the User- and Character-CDEs

and communicative function) and sub-concepts of communicative gestures.

4.2 Fusion and Discourse Engine—FADE

The discourse modeling component of a CDE is responsible for interpreting the contributions of the agents within their context of use and for maintaining a coherent representation of the ongoing discourse. It comprises two submodules, a short-term local turn context based on a production rule system and a long-term, three-tiered discourse context representation, and models the flow of the interaction from the perspective of an individual agent. The interpretation of perceived events is based on the agent's current conversational role (e. g., speaker, addressee, overhearer).

The local turn context provides a comprehensive model of the current situation within the conversation. It models the physical environment plus all attending co-participants with respect to their current role within the conversation and their perceived internal state. This enables a CDE to interpret the perceived interactional contributions with respect to the current state of the conversation. If, for example, a virtual character raises its index finger into the visual field of another agent, this means either that the agent wants to take the turn (if its current role is that of an addressee, or overhearer) or that it wants to prevent another agent from taking the turn (if its current role is that of a speaker). The second sub-module, the discourse history, keeps track of the ongoing discourse and provides a comprehensive history of the discourse contributions. This enables the generation component to produce referring or elliptical expressions.

Moreover, FADE is also responsible for eliciting reactive, semi-conscious behavior of the characters. Semi-conscious behavior comprises actions that are hard to control for humans, e. g. displaying the individual understanding of the current state of the turn-taking process or displaying backchannel feedback. This behavioral class demands for some reasoning and inference processes in order to display appropriate behavior. If, for example, another participant starts to speak, FADE directly triggers the generator to produce appropriate gaze behavior. For a detailed description of FADE see [18].

4.3 Action Management

Each character CDE is endowed with an action management unit that plans, executes, and monitors the actions of the character. The actions are influenced by the goals set by the narrative engine, communicative input by other characters, updates about affective state from the affect engine, and the internal state of the character.

The dialogue is modeled using a *dialogue games* approach (see e. g. [11]), which defines an interaction in terms of rule-based moves that are exchanges of dialogue acts (the games also occasionally include physical acts, such as placing a player on the football field in phase 2. Physical acts are treated the same as dialogue acts, with the exception that they do not have an addressee). As in a board game, there are constraints on which type of move can follow another. The constraints are seen as social conventions that ensure that the dialogue is coherent, and that the participants in an exchange can synchronize their actions to achieve *joint goals* [8]. For example, the joint goal—or *purpose*—of a question-answer dialogue is that the questioner comes to know the answer to her question. As in human-human dialogue, the participants in an exchange can expect the others to adhere to the social conventions, i. e. the questioner can expect that in response to a question, she either will get an answer, a statement of ignorance, or at least a refusal to answer. On the other hand, a *greeting* dialogue act following a question violates the rules.

The assumptions underlying the games are formulated as preconditions and postconditions. A character may assume that the rules of the game are shared by all other participants. Beyond the expectations for future moves, it may also assume that upon a *confirmation* of a *command*, the confirming character will be committed to perform the respective action. Asking an "insincere" question (i. e. one the questioner already knows the answer of) also is against the rules in the general case. However, depending on the situation, e. g. the moderator asking a quiz question, the social rules are different. This can be accommodated by using a game with different conditions.

The narration engine is the primary source of goals for the virtual characters; in addition, goals can also be triggered by changes in the internal state of a character (e. g., a strong affect can cause a sub-goal to complain). Each goal type is associated with an activity that it triggers, described in ontological terms, and it can be furnished with additional constraints, e. g. a timeout, or failure conditions.

For example, a goal for the moderator to pose a quiz question is parameterized by an instance of *FootballQuizQuestion* with roles that specify the addressee of the question, an instance of *FootballSituation*, and several alternative answers. The virtual character that executes an activity will start a sequence of dialogue games to bring about the associated goal. In this case, it will be the moderator that wants the addressee to answer the question. To this end, the moderator will first start a dialogue that presents the football situation to the addressee: Hhe shows a video and enumerates the alternative outcomes of the situation. Subsequently, he will play a question-response game with the addressee to find out her opinion. If something goes wrong, e. g. the addressee do refuses to answer, the character sends feedback to the narration engine about the failed goal. Also, in case of a goal succeeding, the feedback to the narration engine includes information about the task state, e. g., whether the addressee answered correctly. The narration engine can react in both cases by adapting the story.

Participants not actively engaged in a dialogue game can nevertheless perceive the utterances as *overhearers.* This is used, e. g., to achieve bickering among the rival football experts. If one experts overhears a human user endorsing the opinion of the other expert, it will send a negative affect to the affect engine. This is likely to eventually trigger a derogatory comment, as the mood of the character goes bad.

When a character decides to make a dialogue move, it generates an ontological instance containing the information, e. g. an *OfferSelection* move containing a list of answering options for a football question, like in Fig. 5. Next, the character asks FADE whether it can grab the turn, which may take a while if another utterance is currently being realized. While waiting for the turn, the characters generate turn-grabbing gestures (hand waving etc.). As soon as the character gets the turn, the utterance instance is forwarded to the multimodal generation component to be enriched with appropriate gestures, e. g. counting gestures if the moderator enumerates the items of a possible selection, and timing information (see Fig. 6). Finally, the generated utterance is dispatched to the multimodal player. Only after each part (speech utterance, gestures, etc.) of the contribution is realized, its semantic content is forwarded to the CDEs of the overhearers, which perceive it this way and can react.

Dialog systems usually use one of several established approaches for dialog management, with specific advantages and disadvantages. Common variants are based on planning and/or logical inference, finite-state machines, and forms, in order of decreasing representational power, flexibility, but also computational complexity. Whether an approach is suitable depends on the application domain and task structure.

Our domain shows mixed characteristics. The story contains elements (like introductions) that have little variation and can be scripted, but the user interaction and autonomous behavior by the virtual characters also allow for flexible deviations interweaved into the story controlled by the director. Both types of tasks share a common task model, the activity, but the necessary dialog games can be initiated using either a finite-state model, or a plan-based approach using the dialogue games with their preconditions and postconditions as plan operators. It is only assumed and not guaranteed that other characters, and especially the human users, fully cooperate to fulfill the expectations about future moves. For example, the answer to a question may be postponed while the user asks an expert for his opinion. Therefore, it is necessary that a plan of action is constantly monitored and accommodated to reflect the actual situation in the dialogue. The action management mechanism is described in more detail in [10].

4.4 Generation of Multimodal Contributions

```
<OfferSelection>
  <has_initiator>
    <Character><has_name>Moderator</has_name></Character>
  </has_initiator>
  <has_addressee>
    <Character><has_name>User1</has_name></Character>
  </has_addressee>
  <has_content>
    <ListElement>
      <has_listPosition> ... </has_listPosition>
      <has_content>
        <Response>
          <has_content>
            <Parade>
              <has_agent>
                <GoalKeeper> ... </GoalKeeper>
              </has_agent>
              <has_style> <FingerTips/> </has_style>
            </Goal>
          </has_content>
        </Response>
      </has_content>
    </ListElement>
    ...
  </has_content>
</OfferSelection>
```

Figure 5: Example ontological instance representing an *OfferSelection* dialogue act

The generator takes an ontological instance of a dialogue act (see Fig. 5) and turns it into a multimodal contribution represented in PML. The output of the generator contains the spoken utterance, as well as synchronized nonverbal actions (gazes, adaptors, emblematic, iconic, deictic and beat gestures). In the dialogue act, some of the content may be marked as optional. This information is realized depending on the speaker's emotional state and the current discourse context. Moreoever, the realizations of referring expressions will be different, depending on whether an uttered element is a newly introduced concept (*"a car"*), has already been introduced (*"the car"*), etc.

```
<actions id="ac0">
  <character refId="Moderator">
    <speak id="s0" dur="2655" ...>
      <text>b) the goalkeeper saves the
        ball with his fingertips </text>
    </speak>
    <animate id="ag0" dur="1400"
        alignTo="s0" alignType="starts">
      <gesture refId="gazeAtVC1"/>
    </animate>
    <animate id="aa0" dur="2000" ...>
      <gesture refId="countTwo"/>
    </animate>
    ...
  </character>
</actions>
```

Figure 6: PML output for the act in Fig. 5

Since the generation takes place in real-time, the generator tries to cope with time-critical aspects. First of all, generation is bound to be fast and efficient. Secondly, the generator estimates the amount of time necessary to generate the utterance. If the estimate exceeds a certain threshold, we use additional predefined expressions (like *"hhm"*, *"well"*), suitable in the given situation and mood, to signal the other participants that the turn is not available. Our multiparty scenario demands for a turn-taking approach incorporating general gazing behavior as well as actions to take and yield turns. When the generator detects that its virtual character is not the one holding the floor, it might attempt (depending on characteristics like urgency and mood) to claim the next turn by making interrupting statements and gestures.

5. ACTION ENCODER

The action encoder decouples the action planning on an abstract symbolic level from the execution and visualization of those actions in a 3D character player. The CDEs specify the behavior of characters and objects in the scenario by generating PML actions for verbal utterances, accompanying gestures, multimedia objects and graphical user interface elements. Available gestures are defined in a so-called *gesticon*. We use this term analogous to lexicon for a repository of gesture specifications. Gesticon entries describe gestures in terms of their physical form, meaning, and communicative function. In addition, information about character- and player-specific animations associated with this gesture is provided.

The PML actions generated by the CDEs comprise the symbolic name of the gesture (e.g. finger ring) and possibly additional parameters (e.g. speed and hand(s) to be used) that are required by the action encoder to select a corresponding animation. At this particular point, there is no information available about the exact duration of the specified actions since the related animations and audio files have not yet been selected or generated. Therefore, only qualitative constraints can be used to synchronize these actions. In our system we use a set of temporal constraints (e.g. before, overlaps, during) for this purpose. However, the 3D character player needs to know exactly when each action begins and ends. This information is provided by the action encoder which processes the PML actions like follows:

- A text-to-speech (TTS) system is used to generate the audio files and to obtain information about the phoneme types and their duration. This information is later required by the player to select character-specific animations (visemes) for the lip-synchronous mouth movements.
- For each gesture specification an appropriate animation is selected based on information provided in the gesticon and the PML character and object definitions.
- After processing the utterances and gesture specifications, the exact duration of all actions has been determined and the temporal constraints can be resolved. For this purpose a constraint solver is used that computes the exact start and end time for each action.

The action encoder is also responsible for the nonverbal behavior associated with a character's affective state. It receives the affect output produced by the affect module (see Sect. 6) and produces PML actions that control a character's facial expression, complexion, and idle behavior. A character's dominant emotion and its intensity are used to select an appropriate facial animation and to instruct the player to change a character's complexion by smoothly interpolating between different textures. The current mood is expressed through the character's idle behavior. The idle behavior for each mood is a set of animations that are performed in between and sometimes in addition to the gestures specified by the CDEs. A character in a hostile mood, for example, might adopt a tensed posture with the arms folded across the chest. The information about the dominant emotion and the current mood can also be used to modify the animation parameters, e.g., to increase or decrease the speed of conversational gestures and the frequency of the eye blinking.

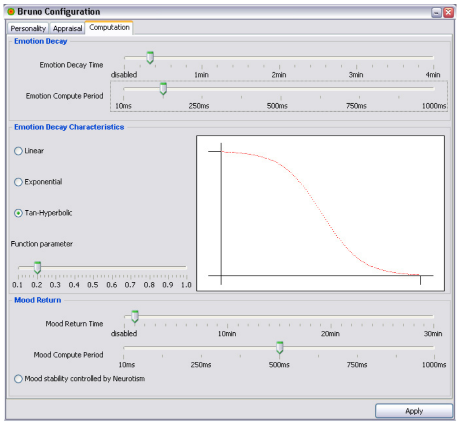
6. AFFECT MODELING

As known from other projects employing virtual characters, like COSMO [9], Émile [6], Peedy [1] and the Greta agent [2] affect successfully help controlling behavior aspects. When analyzing them according to their temporal characteristics, there are short-term behavior aspects, like facial expressions, gestures, or the wording of verbal expression. Also, there are medium-term and long-term aspects, like the process of making decisions, or the motivation of characters. The latter are traditionally represented by CDE like processes. And, there are behavior aspects that consist of mixed-term aspects, like a character's idle behavior that includes for example eye blink (short-term) and medium-term posture changes. Our approach to control such behavior aspects relies on a computational model of affect [3] that provides different affect types. It simulates three interacting kinds of affect as they occur in human:

1. **Emotions** reflect short-term affect that decays after a short period of time. Emotions influence facial expressions, facial complexions (e.g. blush), and conversational gestures.
2. **Moods** reflect medium-term affect, which is generally not related with a concrete event, action or object. Moods are longer lasting affective states, which have a great influence on humans' cognitive functions [16].
3. **Personality** reflects long-term affect and individual differences in mental characteristics [13].

Our work is based on the computational model of emotions described in [5]. It implements the OCC model of emotions [17] combined with the five factor model of personality [13] to bias the emotions' intensities. All five personality traits (openness, conscientiousness, extraversion, agreeableness, andneuroticism) are used to influence the intensities of the different emotion types. The OCC cognitive model of emotions is based on the concepts of appraisal and intensity. The individual is said to make a cognitive appraisal of the current state of the world. Emotions are defined as valenced reactions to events of concern to an individual, actions of those s/he considers responsible for such actions, and objects/persons. The OCC theory defines 22 emotion types. The intensity of emotions underlies a natural decay, which can be configured by several decay functions (linear, hyperbolic, exponential), and which is influenced by personality values (see figure 7).

The employed computational model of moods is based on the psychological model of mood (or temperament) proposed by Mehrabian [15]. Mehrabian describes mood with the three traits pleasure (P), arousal (A), and dominance (D). The three traits are nearly independent, and form a three dimensional mood space. A PAD mood can be located in one of eight mood octants. A mood octant stands for a



Bruno Configuration

Personality | Appraisal | Computation

Emotion Decay

Emotion Decay Time

Emotion Compute Period

Emotion Decay Characteristics

○ Linear

○ Exponential

● Tan-Hyperbolic

Function parameter

Mood Return

Mood Return Time

Mood Compute Period

○ Mood stability controlled by Neurotism

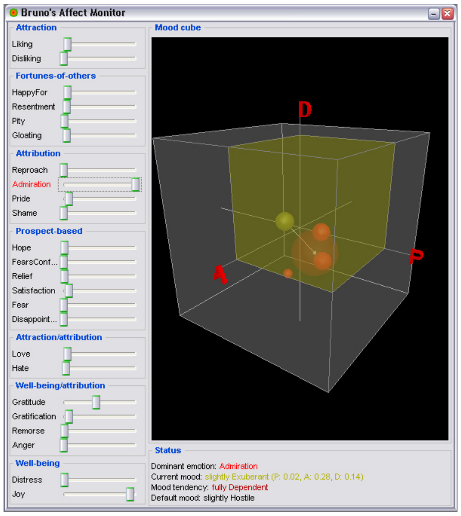
Apply

Figure 7: Affect configuration parameters

discrete description for a mood: +P+A+D is exuberant, -P-A-D is bored, +P+A-D is dependent, -P-A+D is disdainful, +P-A+D is relaxed, -P+A-D is anxious, +P-A-D is docile, and -P+A+D is hostile. Generally, a mood is represented by a point in the PDA space. For mood computation, it is essential to define an virtual human's default mood. The mapping presented in [14] defines a relationship between the big five personality traits and the PAD space. Using this mapping, the above mentioned model of emotions, which uses the big five personality model to define a character's personality, is thereby able to compute a default mood. The computation of mood changes is based on active emotions generated by the computational model of emotions. Each appraisal of an action, event or object elicits an active emotion that once generated, decays over a short amount of time (i.e. one minute). All active emotions are input for the mood function. The function has two scopes. Based on all currently active emotions the function defines whether the current mood is intensified or changed. It will be intensified if all active emotions are mapped into the mood octant of the current mood. A mood will be changed progressively if all active emotions are mapped into a different mood octant than the current mood. The mood function is visualized within an affect monitor that is shown in figure 8.

The current version uses also the current mood to compute the intensity of active emotions in order to adapt the emotions' intensity on the characters' current situation. This increases, for example, the intensity of joy and decreases the intensity of distress, when a character is in an exuberant mood. A detailed description of the mood function can be found in [3].

In affect computation, the first step is to appraise relevant input by using a character's own subjective appraisal rules, introduced in [5]. Three types of affect input are distinguished: 1) basic appraisal tags, 2) act appraisal tags, and 3) affect display appraisal tags. Basic appraisal tags express how a speaking character appraises the event, action or object about which it talks. Act appraisal tags describe the underlying communicative intent of an utterance, e.g. tease, or congratulate. Affect display appraisal tags are visual cues of an experienced emotion or mood, e.g. a blush of shame or a character that looks nervous for a specific amount of time. The output of the appraisal process is a set of emotion eliciting conditions. Based on them active emotions are generated that in turn influence a character's mood. The relevant input is provided by a character's CDE and consists of appraisal tag input, dialog act input, emotion and mood input, about speaker, addressee and listener. The affective profile is passed to the character's CDE and also to the player component which is responsible for rendering the character's visual appearance and its speech output.



Bruno's Affect Monitor

Status

Dominant emotion: Admiration
Current mood: slightly Exuberant (P: 0.02, A: 0.28, D: 0.14)
Mood tendency: fully Dependent
Default mood: slightly Hostile

Figure 8: Affect monitor

The evaluation of this computational model of affect shows that nearly all generated affect types (emotions, moods) are plausible to humans [4]. Based on these results we are confident that the affect visualization through facial expressions and complexions, gestures, posture changes, and through different dialog behavior realized by each character's CDE is plausible too.

7. CONCLUSIONS

In this paper, we presented an overview to the VirtualHuman system. It provides a knowledge-based framework to create interactive applications in a multi-user, multi-agent setting. The behavior of the virtual humans and objects in the 3D environment is controlled by interacting affective conversational dialogue engines. An elaborate model of affective behavior adds natural emotional reactions and presence of the virtual humans.

During CeBIT 2006, VirtualHuman was displayed for one week. Figure 9 shows the setup. The studio was displayed on a back projection screen. The two interaction posts were equipped with a microphone and a track-ball for the user's interaction. The microphones were open all the time. The sound compressor limited the signal level so that users could participate without using a push-to-talk button. Even though the fair hall was very noisy, we had almost no false starts of the recognizers and the sound quality of the speech signal was always sufficient. People could approach the system without control and interacted naturally with the virtual characters in the game show.

The results of an evaluation with 21 naive subjects demonstrated the general acceptance of the system. The subjects evaluated the systems on various dimension, using a 5-point Likert scale. Especially the appearance and the behavior

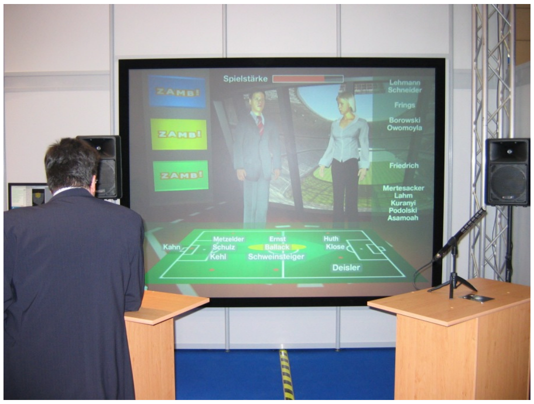


Figure 9: The CeBIT installation with two interaction posts

of the virtual humans in the studio scenario where ranked positive. The affective expressiveness was noticed and provided an added-value. The speech related capabilities of the system need to be expanded, especially the quality of the speech synthesis was valued more negatively.

8. ACKNOWLEDGMENTS

The research presented here is sponsored by the German Ministry of Research and Technology (BMBF) under grant 01 IMB 01A (VirtualHuman). We thank our student assistants (in particular Benjamin Kempe who developed the multimodal generator) and the project partners. The responsibility for the content lies with the authors.

9. REFERENCES

[1] G. Ball and J. Breese. *Embodied Conversational Agents*, chapter Emotion and personality in a conversational agent, pages 189 – 219. MIT Press, 2000.

[2] C. D. Carolis, M. Bilvi, and C. Pelachaud. APML, a Mark-up Language for Believable Behavior Generation. In *the AAMAS Workshop on "Embodied conversational agents – Let's specify and evaluate them!*, Bologna, 2002.

[3] P. Gebhard. ALMA - A Layered Model of Affect. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 29 – 36, Utrecht, Netherlands, 2005.

[4] P. Gebhard and K. H. Kipp. Are Computer-generated Emotions and Moods are plausible to humans? In *Proceedings of the Sixth International Conference on Intelligent Virtual Agents*, Marina del Rey, USA, 2006.

[5] P. Gebhard, M. Klesen, and T. Rist. Coloring Multi-Character Conversations through the Expression of Emotions. In *Proceedings of the Tutorial and Research Workshop on Affective Dialogue Systems*, pages 12 – 25, Kloster Irsee, Germany, 2004.

[6] J. Gratch. Émile: Marshalling passions in training and education. In C. Sierra, M. Gini, and J. S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 325–332, Barcelona, Catalonia, Spain, 2000. ACM Press.

[7] J. Gratch, J. Rickel, E. André, J. Cassell, E. Petajan, and N. I. Badler. Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, 17(4):54–63, 2002.

[8] J. Hulstijn. Dialogue Games are Recipes for Joint Action. In *Proceedings of the Gotalog Workshop on the Semantics and Pragmatics of Dialogues*, Gothenburg, Sweden, 2000.

[9] J. Lester, J. Voerman, S. Towns, and C. Callaway. Cosmo: A life-like animated pedagogical agent with deictic believability, 1997.

[10] M. Löckelt. Action Planning for Virtual Human Performances. In *Proceedings of the third International Conference on Virtual Storytelling*, Strasbourg, France, 2005. Springer.

[11] W. C. Mann. Dialogue Macrogame Theory. In *Proceedings of the 6th workshop on semantics and pragmatics of dialogue (EDILOG)*, pages 109–116, Edinburgh, 2002.

[12] M. Mateas and A. Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference, Game Design Track*, Mar. 2003.

[13] R. McCrae and O. John. An introduction to the five-factor model and its implications. In *Journal of Personality*, 60, pages 171 – 215, 1992.

[14] A. Mehrabian. Analysis of the Big-five Personality Factors in Terms of the PAD Temperament Model. In *Australian Journal of Psychology*, 48 (2), pages 86 – 92, 1996.

[15] A. Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. In *Current Psychology*, 14, pages 261 – 292, 1996.

[16] W. N. Morris. *The frame of mind*. Springer-Verlag, New York, 1989.

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

[18] N. Pfleger. Fade - an integrated approach to multimodal fusion and discourse processing. In *Proceedings of the Doctoral Spotlight Session of the International Conference on Multimodal Interfaces (ICMI'05)*, pages 17–21, Trento, Italy, 2005.

[19] P. Piwek, B. Krenn, M. Schröder, M. Grice, S. Baumann, and H. Pirker. RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA. In *Proceedings of the Workshop on Embodied Conversational Agents - Let's specify and evaluate them!*, 2002.

[20] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 1995.

[21] W. Swartout, J. Gratch, R. Hill, E. Hovy, S. Marsella, J. Rickel, and D. Traum. Toward virtual humans. *AI Magazine*, 27(2):96–108, 2006.