Multi-party Conversation for Mixed Reality

Markus Löckelt, Norbert Pfleger and Norbert Reithinger

Abstract—**The interactive scenarios realized in the two prototypes of Virtual Human require an approach that allows humans and virtual characters to interact naturally and flexibly. In this article we present how the autonomous control of the virtual characters and the interpretation of user interactions is realized in the Conversational Dialogue Engine (CDE) framework. For each virtual and real interlocutor one CDE is responsible for dialogue processing. We will introduce the knowledge needed for the CDE-approach and present the modules of a CDE. The real-time requirement resulted in the integrated processing of deliberative and reactive processing, which is needed, e.g., to generate an appropriate nonverbal behavior of virtual characters.**

***Index Terms*—Dialogue modeling, multiparty interaction, virtual characters, interactive narratives.**

I. INTRODUCTION

One of the main characteristics of Virtual Human is the lifelike interactivity of a group mixed of real people and believable virtual characters. The characters provide realistically all those types of interaction behavior that are expected by real humans and that typically depend upon the interaction situation. A salient example is the turn-taking behavior in real conversations where people look at each other while they talk.

The interaction aspect is even more prominent in the scenarios realized during the Virtual Human project. Both the physics lesson and the quiz show do not require a complex, drama-like narrative structure. The goals of the scenarios are quite obvious. The entertaining moments must be realized through ad-hoc reactions of the participants to the other's actions. Since humans are involved whose reactions are out of control of the game logic, this cannot be planned in advance at a central location. The virtual actors must employ local reaction strategies in order to achieve a high degree of naturalness.

The highly interactive scenarios realized in the two prototypes (see INTROARTICLE) raises the question of autonomy vs. guidance. There is a widespread discussion (see, e.g., [2, 3]) of the advantages and disadvantages of a strong guidance through a narration structure, which guarantees a coherent story experience, and the autonomous approach where each virtual character is allowed to act and react on its own behalf.

In this article we will introduce the technology behind the autonomous control of the virtual characters and will present the Conversational Dialogue Engine (CDE) framework, which realizes the control of the m virtual humans and the interaction with n users. The examples are drawn from the second example scenario, the quiz "Zweiundachtzig Millionen Bundestrainer" (eighty-two million national coaches, ZAMB).

We will first present some requirements for the processing of multiparty interactions in real-time in section Ⅱ and will introduce some example interactions in section Ⅲ. The following sections introduce the knowledge sources we use in the CDEs and provide an insight in the details of the main CDE components, namely the multimodal fusion and discourse engine FADE and the action manager.

II. MULTIPARTY INTERACTION BETWEEN HUMANS AND VIRTUAL CHARACTER

The interaction between a number of human users and virtual characters that converse in the scenario of a quiz show poses complex challenges on the knowledge representation, reasoning and processing within each character. As already mentioned, the characters are modeled as individual and semi-autonomous entities. This is necessary in order to be able to react as life-like as possible. However, not only the virtual characters need modeling, also for the human interlocutors we must be able to process their multimodal input-namely speech and gestures on objects in the virtual world.

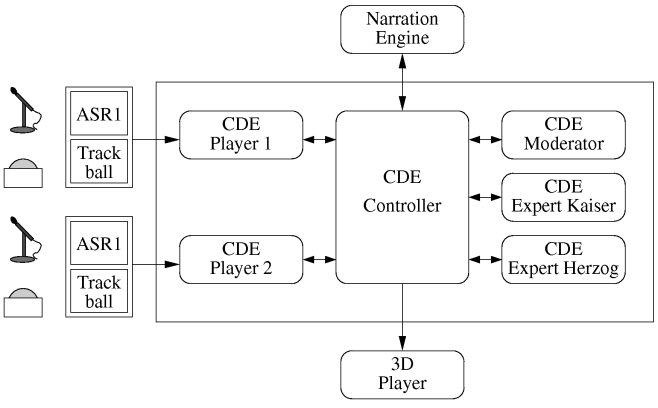

Fig. 1. Overview of the CDE architecture.

Central to our notion of processing are the conversational dialogue engines, or CDEs (see Fig. 1). For each of the real and virtual humans we instantiate one of these engines. In the figure, we have two CDEs for the human partners (user CDEs) and three for the virtual characters (character CDEs). User CDEs process the inputs from multimodal recognizers (shown left in

Manuscript Received on March 12, 2007.

Markus Löckelt has done work in the area of dialogue management, action planning, and cognitive modeling of computer characters. E-Mail: loeckelt@dfki.de.

Norbert Pfleger is working as a research scientist at the Intelligent User Interface lab at DFKI Gmbh, Saarbrücken. His main research interests are in the area of multimodal dialogue systems, multimodal fusion and discourse processing and conversational human-computer interaction. E-Mail: pfleger@dfki.de.

Norbert Reithinger majored in computer science and in 1983 earned his Diplom-Informatiker degree with excellence. He is member of ACM and the IEEE computer society. E-Mail: norbert.reithinger@dfki.de.

the figure), while character CDEs implement the reasoning and triggering of their actions. Via a controller component, the CDEs communicate with each other and with the narration engine and the 3D player component. To manage this complex task, we have to address various different areas of research:

Knowledge representation: The CDEs need to model different areas of expertise. The actors are part of a game world which includes the other dialogue participants in a virtual physical environment. Facts about these objects are covered by declarative world knowledge. The task knowledge is related to the objects relevant to the task, like football players with their strengths and weaknesses, the playing field, and facts about football matches. Finally, the characters need to know the rules and social conventions of interaction, which are called communicative knowledge. The different knowledge types are defined declaratively in an ontological representation.

For multimodal interaction, each character additionally needs to know the spatial relationships between the objects present in the physical surroundings. Based on this representation, the characters are able to resolve cross-modal and spatial references like "Put Ballack left of Klose."

Deliberative behavior: The participants must be able to deliberate on their declarative knowledge to purposefully take action to bring the story forward. The narration engine is concerned with the global situation in the scene and assigns story goals to the virtual characters, each of which maintains its own private view of the state of the world and infers from it a course of action to satisfy these goals. This private state, and the knowledge how actions can be used to modify it in the desired way, is modeled using building blocks in three layers, from activities corresponding to story goals, over dialogue games that represent interactions between participants. The dialogue games in turn comprise communicative acts that are exchanged between the participants and realized by the player.

Multimodal interaction: The mixed reality approach of VirtualHuman must employ as much natural modalities as possible to realize a convenient interaction experience. The real humans can address their virtual counterparts through speech and can interact with objects in the virtual world. In the physics lesson scenario, the users were able to manipulate charts, in the ZAMB game show they can select from a multiple choice menu in the first part, or they can grab a player's name from a list and place it on the playing field. The interpretation of speech and gestures needs advanced methods for speech integration, gesture analysis and modality fusion. For speech recognition we use the modules provided by our partner, for synthesis the commercial Scansoft product.

The virtual humans also communicate through various modalities. A virtual human addresses the other members of a group with speech and body gestures which must be combined to reach a high degree of naturalness. Gestures are also of utmost importance to signal turn-taking, attention, and emotions. The CDE for each virtual human must therefore listen also to the other virtual partners and signals constantly its part in the conversation.

Real-time interaction: Human communication is obviously a function of time. In a conversation we expect timely reaction of a dialogue partner. If, e.g., a reaction is delayed, we immediately attribute this delay a meaning, e.g., the partner may think about an argument. We also immediately watch the partner for clues in her facial and body expression whether the delay has, e.g., some additional emotional reason. In VirtualHuman we therefore have no choice but to interact in real dialogue time: Utterances and behaviors must be generated and analyzed in the time-spans that are expected by humans in natural inter-human communication. The CDE framework therefore is tuned to enable an immediate reaction usually below one second, which ensures that the perception of a real-time interaction is not affected [1]. This requirement of course influences the approach we can use internally for processing. Since the interaction flow must not be interrupted, algorithms that respect the real-time requirement are to be preferred to those that deliver theoretically better solutions but delay the interaction beyond acceptability. An example for this is the access and manipulation of the elements in the knowledge bases in the CDEs, which are stored as ontologies in RDFS format. The "default" tool for this would be a toolkit such as Jena (see http://jena.sourceforge.net). While Jena offers access to the full power of RDFS, this also means a considerable overhead on the performance of some operations. Realizing that we would not be able to provide reliable real-time performance when using Jena, and on the other hand, that we did not actually need the full feature set, we implemented a derived API called Jena-Lite using a JDOM representation. Jena-Lite only supports the required subset of the Jena features, but is optimized for speed and memory savings.

Reactive behavior: Reactive behavior comprises actions that are hard to control, 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. An addressee displaying backchannel feedback needs to know: (i) the exact location of where to feedback is appropriate (the point(s) within a turn at which an addressee can take over or can display backchannel feedback; see [18]) and (ii) the current status of the understanding process to determine the most appropriate response. The generation of backchannel feedback is triggered by FADE while the actual action is generated by the multimodal generation component. FADE needs to constantly monitor the perceived actions of the speaker and the other participants in order to determine the appropriate spots for displaying feedback.

III. INTERACTION EXAMPLES

Fig. 2 shows the studio setup of the first ZAMB game show part, where a virtual moderator and two virtual experts, Mrs. Herzog and Mr. Kaiser, interact with the two human participants. In the following example dialogue we demonstrate the verbal interactions together with some annotations about nonverbal activities. In the following sections we will refer to these dialogues while explaining the inner working of the CDEs[2]:

[2] The original dialogue is in German. For brevity reasons, we provide only the English translation.

(1) MODERATOR: ...Now look closely [shows video on screen]. What will happen next? The alternatives are [counting gesture] One - Ballack scores the goal, [co*unting gesture]* Two - the keeper does a parade, *[counting gesture]* Three - Ballack kicks the ball into the sky.

Fig. 2. Screen-shot of the first phase of the Virtual-Human system; from left to right: The moderator, the virtual expert Kaiser, and the virtual expert Herzog. See Color Plate 6.

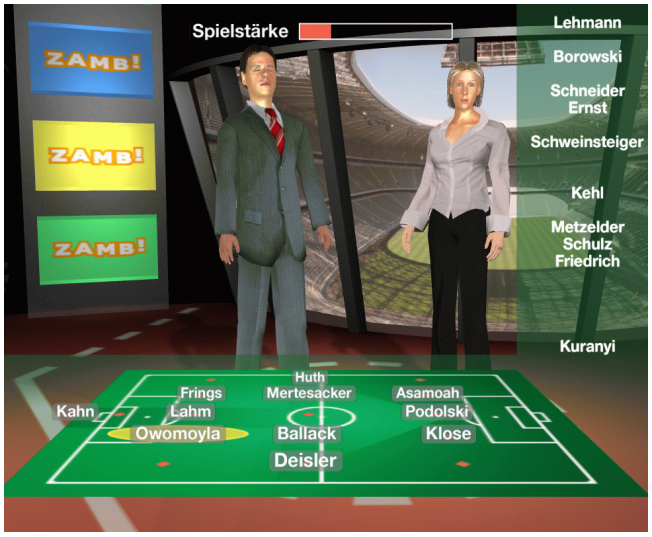

Fig. 3. Screen-shot of the second phase of the VirtualHuman system. See Color Plate 7.

(2) MODERATOR: [gazes at Kaiser] What do you think, Mister Kaiser?

(3) EXPERT KAISER: [gazes at moderator] I think Ballack scores the goal.

(4) MODERATOR: [appreciative gesture] Spoken like a real football trainer.

(5) MODERATOR: [gazes at user 1] Now, player one, what is your guess?

(6) USER 1: Mr. Kaiser, what do you think?

(7) EXPERT Kaiser: [blushes] I think the keeper does a parade.

(8) MODERATOR: An interesting opinion.

(9) MODERATOR: [looks again at user 1] Now it's your decision, player one.

(10) USER 1: I think Mr. Kaiser is right.

(11) EXPERT HERZOG: [gets angry] How can you believe this amateur!

(12) EXPERT KAISER: [smiles]

(13) MODERATOR: Alright, answer one.

Fig. 3, the setup of the second phase is shown where the winner of the first phase of ZAMB selects his favorite team with the support of Mrs. Herzog. The example interaction demonstrates some more complex interactions by the user, including spatial references:

(21) MODERATOR: Ok, let's get started.

(22) USER: Put [characters gaze at user] Kahn up as keeper.

(23) EXPERT HERZOG: [nods] That's an excellent move! You can't go wrong with Oliver Kahn as a goalie.

(24) MODERATOR: Well [nods] great, Kahn as keeper.

(25) USER: Mrs. [characters gaze at user] Herzog, give me a hint!

(26) EXPERT HERZOG: [smiles] I recommend a defensive strategy against Brazil. I would definitely put Ballack into the midfield.

(27) USER: Ok, let's do that.

(28) EXPERT HERZOG: [smiles, nods] You won't regret this move.

(29) MODERATOR: [nods] Great, Ballack as midfielder.

(30) USER: [hesitates, does not say anything]

(31) MODERATOR: [encouraging gesture] Don't be shy!

(32) USER: Hhm, [characters gaze at user] put Deisler to Ballack's right.

(33) MODERATOR: [shrugs] That is not possible, I'm afraid that place is already occupied.

IV. REPRESENTING KNOWLEDGE

The knowledge base for a flexible dialogue infrastructure like VirtualHuman needs to cover different areas, and it is advantageous to clearly separate the knowledge for each area to facilitate reuse [20]. The different types of knowledge are the following (see also Fig. 4):

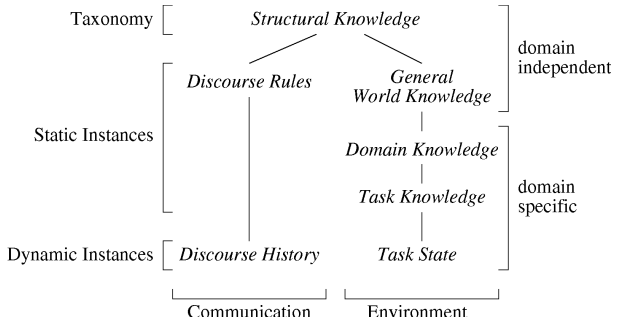

Fig. 4. The knowledge hierarchy of VirtualHuman.

Structural knowledge describes the taxonomic relations between concepts (types of objects). It is concerned with concept hierarchies (e. g. is-a relations) and relations between instances of concepts (e. g. has-a relations). For VirtualHuman we store the knowledge in the W3C recommendation RDFS

and use the speed optimized Jena-Light framework for processing.

General world knowledge concerns factual information about the world, which includes objects, their attributes and relations between them. It is extended by domain knowledge which is about the concrete domain of the application. For example, for the VirtualHuman system that is concerned with the domain of football, the domain knowledge of the CDEs includes the concept of football player with has attributes such as the player's name, physical fitness, or nationality. It specifies that the FootballPlayer category is a subcategory of Human and has subcategories like Defender and Goalkeeper. The knowledge base also contains objects that are concrete instances of the FootballPlayer concept. There is no absolute necessity for a separation of general world knowledge and specific domain knowledge. However, it makes it possible to share and re-use existing bodies of domain-independent general world knowledge across different applications. We used the freely available Protégé editor (see Fig. 5 for the definition of a football player) to define an extension of the base ontology given in [4].

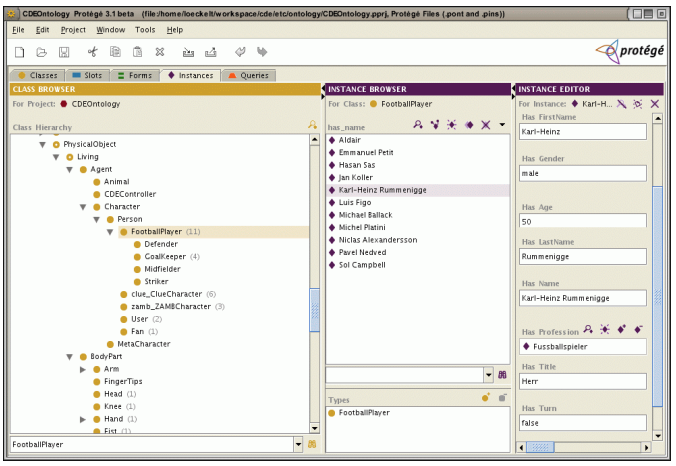

Fig. 5. The definition of a football player with Protégé.

Task knowledge connects the interaction and the general world and domain knowledge with respect to the goals of the system. In VirtualHuman, it encompasses what the different quiz games involve and how the characters go about realizing their roles. Task knowledge is defined in terms of goals and the actions that are necessary to achieve them, how different actions change the state of the CDE's goals, and what they mean for the character's future actions and behavior. Task knowledge can be seen as a special case of world knowledge.

Knowledge about the discourse rules defines appropriate contributions in a given context, and their meaning in that context. Coherent interaction has rules, and the individual actions have an effect. For example, a question will be about some subject which must be known and identifiable from the world knowledge. It influences the knowledge state of the question's addressee, e. g. in that it is inclined to return an answer.

The discourse history models the content and structure of instances of actual dialogical interactions between the interlocutors. Information that is obtained in the course of the dialogue may also be accepted permanently by dialogue participants to become part of their world or domain knowledge. The discourse history records the communication in the system.

In parallel to the discourse history, the situation also includes information that represents the *task state.* This information is influenced by the discourse, but separate from it. The task state may also change due to, e.g., physical actions of the characters, or external events that do not have anything to do with the interaction per se.

Some of these knowledge types are defined prior to a conversation and are common to all CDEs, like e.g. general facts about football or conversation. Others, like the discourse history, are created during an interaction and define the flow of the ongoing activities.

V. INSIDE THE CONVERSATIONAL DIALOGUE ENGINES

In VirtualHuman, the goal is to accompany the life-like visual modeling of the characters with realistic behaviors of the simulated participants. They have to act as individuals, each with a separate private view of the virtual environment, as well as own beliefs, goals, and intentions. In addition, the setting dictates that they also have to perform as virtual actors under the regime of a narrative control acting as a director. This director guides the characters through the plot by assigning goals to achieve. For example, during the first phase of ZAMB, the moderator is first given the goal to introduce the other characters and the quiz to the users, and later to present the videos with the appropriate questions, at the same time, the expert characters are directed to provide advice to the user.

To realize this flexibility, each character is modeled by a separate conversational dialogue engine (CDE) with a set of possible communicative behaviors at its disposal, which are available to reach the goals set by the director. A central controller module manages the communicative exchange between the CDEs and the other modules in the system. How the CDEs in VirtualHuman are connected is shown in Fig. 1.

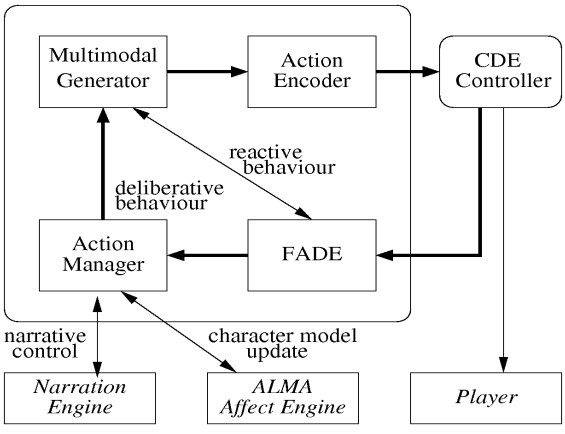

Fig. 6. The internal structure of a CDE.

Fig. 6 shows an inside view of a character's CDE. The main data flow goes from the virtual environment (managed by the controller) via FADE, the action manager, the multimodal generator, and the action encoder back to the CDE controller.

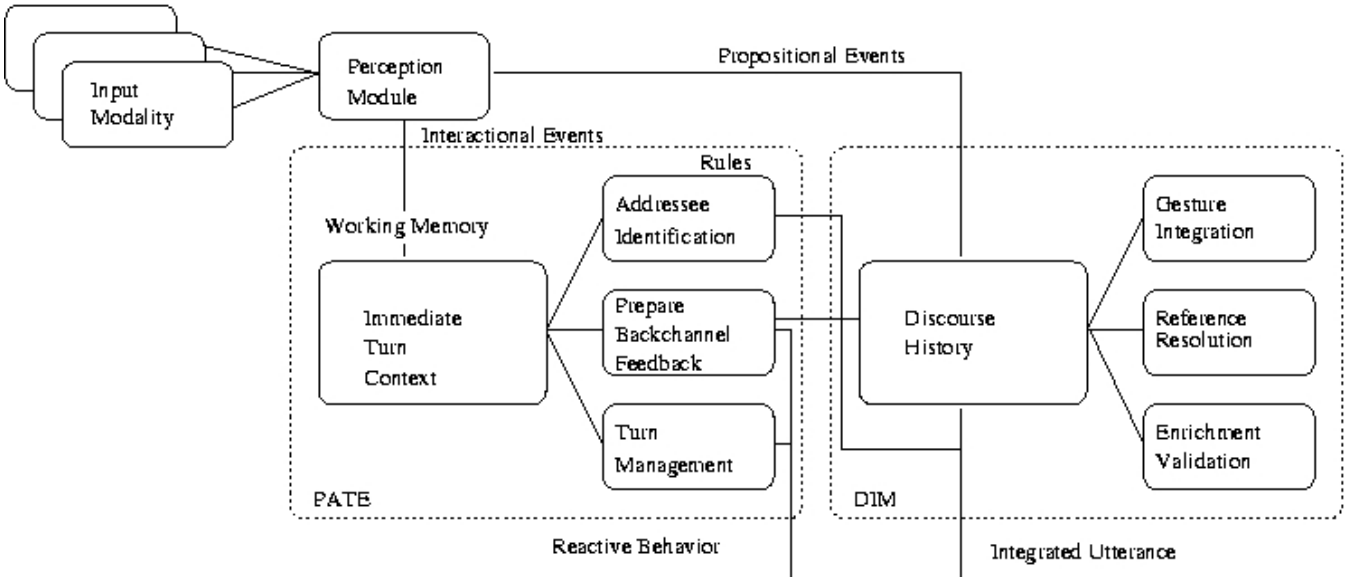

Fig. 7. The basic architecture of the reactive multimodal fusion and discourse processing component FADE.

The action manager is interacting with the affect engine ALMA to exchange character model updates. The top-level goals are defined by the narration engine.

5.1 The Multimodal Fusion and Discourse Engine (FADE)

When two or more people engage in a conversation, they need to coordinate their contributions in order to achieve a successful and smooth exchange of information. This coordination takes place not only on a propositional level but also on an interactional level. Following [11], [12], contributions to conversations can be divided into propositional and interactional information. While propositional information contributes to the content of the conversation, interactional information contributes to the regulation and organization of the conversational process. Thus, meaningful speech and gestures that complement or elaborate on the speech content both contribute to the propositional aspect of contributions. Interactional information, however, is realized by means of a range of nonverbal behavior (like head nods indicating that one is listening, gazes, etc.) and para-verbal speech (like hmm, huh, etc.). Moreover, the more participants take part in a conversation, the more complex the coordination of the interaction gets.

A smooth exchange of turns requires signalling clearly when a character attempts to take the turn, so that the human user and the other participants will not interrupt while the utterance is planned and generated. Silent and filled pauses are often used between human interlocutors for these strategic purposes, e.g., taking, holding and yielding the turn [15, 14]. Consider, for example, turn (24) of our second dialog where the moderator takes the turn in order to evaluate the answer of the user:

MODERATOR: Well *[nods]* great, Kahn as keeper.

But since he is still busy with planning the utterance he uses "well, …" at the beginning of the turn to underline his claim for the turn. When generating longer utterances, the generation component exhibits a similar behavior. The kind of verbal filler used for the pauses depends on the dialogue act type to be generated; additionally, the system can provide explicit cues about the semantic content, e.g., whether an answer to a question is positive or negative. When the generator determines that it needs to fill a pause, it draws from a collection of set pieces to be inserted. This can include general utterances (like "hmm …") as well as more specific utterances depending on the discourse context ("one moment pleas..."', "what I want to know is …").

The identification of the intended addressee(s) is another important task that participants of conversations with more than two speakers have to perform continuously [16]. The main question is how speakers signal who the actual addressee(s) of their contribution are so that all present listeners can easily determine on a moment-to-moment basis who is supposed to react, i.e., take the floor, when the speaker has finished. To what extent a speaker needs to determine the addressee depends on the dialogue situation and the context within which an utterance takes place. Consider, for example, the question in (1) which leaves its addressee(s) open when viewed on its own.

MODERATOR: Let me know when you are ready.

However, when some contextual information is available - e.g., we are currently in phase two which means that only one human user is left. Then it is easy to infer that the moderator actually addresses this user. In phase one, however the moderator would be required to indicate which user he is referring to either by means of gazing behavior or through explicit reference, e.g.:

MODERATOR: Player one, let me know when you are ready.

Another key linguistic phenomenon of verbal interaction is the use of referring expressions like *Mr. Kaiser* (2) or *this amateur* (step 11) to denote or refer to one person. Both referring expressions denote one person named Kaiser (their referent). If two referring expressions denote the same entity,

they are said to *corefer*. A referring expression that refers to some entity or concept of the physical, situational or discourse context is called *deixis* or *deictic expression*. Examples of deictic expressions are personal pronouns (e.g., *I, you*, etc.), adverbial expressions (e.g., *here, now,* etc.), and demonstrative pronouns (*this, that,* etc.). However, the category deixis also subsumes referring expressions relating to an entity that has been introduced during previous discourse.

A general characteristic of deictic expressions is their immanent context dependency. Thus, referring expressions can only be interpreted with access to their context of use. Another aspect emphasizing the role of referring expressions is their contribution to efficient and coherent communication. Grice's maxims of quantity and manner (see [13]), for example, postulate that speakers should keep their contributions brief and as informative as necessary and this can only be achieved by using referring expressions.

On an abstract level, FADE consists of two processing layers (see Fig. 7): (i) a production rule system called PATE; *a Production rule system Based on Typed Feature Structures* (see [7, 8]) that is responsible for the reactive interpretation of perceived monomodal events, and (ii) a discourse modeler (called DiM) that is responsible for maintaining a coherent representation of the ongoing discourse and for the resolution of referring and elliptical expressions (see [9, 10]).

Making sense of perceived monomodal events consists of two aspects: (i) interpreting interactional signals in order to trigger appropriate reactions, and (ii) the integration of monomodal contributions that contribute to the propositional content of the turn. The Perception Module distinguishes the incoming monomodal events respectively and updates the immediate turn context and the DiM. Key to our approach is that all processing instructions necessary to interpret the interactional events can be expressed through production rules. The remaining integration task is handled by the discourse modeling subcomponent. For more information about FADE please see [19].

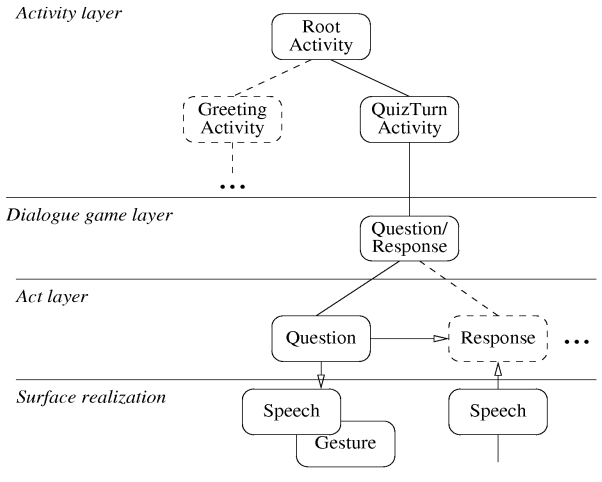

Fig. 8. Example for the three-levelled interaction structure.

5.2 The Action Manager

The elements of the communicative behaviors, and the interaction structure that results from their use, are organized hierarchically in three levels (see Fig. 8): *Dialogue acts* are used as the atomic units of communication between interlocutors, *dialogue games* specify rule-governed exchanges of dialogue acts, and *activities* use combinations of dialogue games to implement a character's goal-directed behavior. The underlying motivation for this structuring is to identify basic patterns occurring in storytelling that can be made into building blocks to be generalized and reused across similar situations.

For an illustration, Fig. 8 shows the structure of the interaction for the Moderator character immediately after turn (5) in the example of section Ⅱ, namely the question

MODERATOR: Now, player one, what is your guess?

After (5), the moderator expects an answer from the first player, which would be the corresponding "Response" to the "Question". In the example, the user does not satisfy this expectation, but rather decides to first start a *sub-game* by asking the opinion of one of the experts:

MODERATOR: What do you think, Mister Kaiser?

Only after this sub-game has been carried out, the original Question-Answer game is continued with the (indirect answer) of user 1

USER: I think Mr. Kaiser is right.

The example also illustrates that the participants who are not directly involved in a particular exchange nevertheless can overhear its content and uses it later in their own processing. In utterance (7), the expert remembers the question that has been posed to the user, and in (10), the moderator can infer what content the user agrees to.

Communicative acts in the acts layer of Fig. 8 are the atomic units of communication between the participants. The set of possible communicative acts is shared and agreed upon between all CDEs. The set of useful acts depends on the particular scenario. We extended the basic structure from commonly used tag sets for dialogue acts (e. g. [21]). The exact set of acts is not intended to be complete from a speech-theoretical point of view. Rather, we use these acts as a starting point to assemble the interaction types we need for our scenario. Communicative acts specify, as ontological objects, the semantic content to be conveyed (e. g. the focus of a question) as well as preconditions and postconditions. Preconditions must hold for an act to be usable, while postconditions are assumed to hold afterwards. For example, to make a (honest) "Inform" act about a fact F, a precondition would be that the initiating CDE must believe that F holds, and afterwards it may (naively) assume, as a postcondition, that everyone who heard the act also believes in F. However, this is only a subjective assumption, and, depending on the circumstances, some overhearers might also choose to ignore or reject F.

The approach of seeing of dialogical interaction as a game, which is the next layer, has a long history (e. g. [22, 16]). In linguistics, dialogue games are used predominantly in an analytical fashion to show the structure of dialogues. In our approach, dialogue games are used generatively to produce the exchanges of communicative acts between the virtual and real participants, and to generate reasonable expectations for future

utterances. They also function as a device to coordinate the joint actions of the participants [5]. As is commonly the case with games, ours consist of sequences of moves along with rules stating constraints on legal moves for the participants, in their roles as *initiator* or *responder*, to make in a given situation. To make a move, a participant sends an instantiation of a dialogue act to another participant.

Such a game can be depicted as a finite state automaton. Fig. 9 shows two different dialogue games that the Moderator can use to produce the interaction (5)-(13), where terminal states are shown darkened. In the first game, the initiator expects a Response move from the responder after the initial Question. It is also socially acceptable if the responder utters a refusal to answer, or states that he does not know the answer. The two latter choices will terminate the game immediately, while in the first case the initiator will close the game by evaluating the answer.

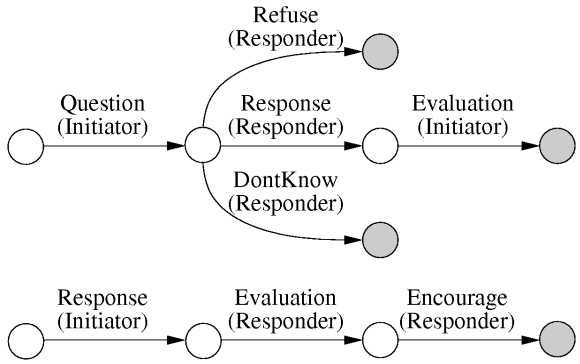

Fig. 9. Two example dialogue games.

The second game is meant to handle the case that another participant answers the Question during the first game. Therefore, it has a precondition that the first game G is currently active, and the Encourage move is addressed at the Responder of G. To ask the question (5) to the user, the moderator will initiate the first game, with the user in the role of the responder. When the user chooses to ask the expert instead of answering right away, the moderator ignores this question (because it is not addressed to him), but overhears the response from the expert. This response is not used to advance the first game, since it does not come from the expected responder in the active game (the user), but the moderator can start a new instance of the second game, initiated by the expert, as a sub-game of the first. Its precondition is satisfied, and it will have the moderator in the role of the responder. He will give an evaluation of the response, followed by encouraging the user. The second game will then terminate, and the first game, which was not finished, can be continued.

Composite games are created by sequencing, embedding, iterating or parallelizing other games and dialogue acts – cf. [6]. Like the dialogue acts, the basic game set is shared knowledge (the games can also be seen as "social conventions"), but characters may add private preconditions and post conditions and have several versions of a game for different conditions. Even though the shared knowledge about the social conventions underlying dialogue games is used to coordinate the joint action, the caveat that the game post conditions are not guaranteed to hold after a game does still apply. For example, a post condition to a Question-Response game may be that the initiator has learned the answer to the question, but there generally may be more than one way for a game to terminate, in this case, the responder might not know the answer at all. When there is a choice between several moves during a game, the character making the turn examines the preconditions of the possible moves, and selects the one that fits best to the current situation: the most constrained move that is still fulfilled is taken. To predict moves by others, a character can use the same method using its own version of the game. Conditions can be logical requirements or context conditions. Logical requirements are checked against the private world state of the character. For example, a greeting move can be prohibited if the character remembers that it has already greeted the potential addressee.

```
<directionML>
 <setGoal>
  <has_name>lineup</has_name>
   <Participant> … Moderator …  </Participant>
  <Participant> … Lebacher …
  </Participant>
  <Participant> … User1 … </Participant>
  <Timeout>360</Timeout>
  <NoResponseEvent>30</NoResponseEvent>
  <Event refId="lastEvaluation"/>
  <Goal>
   <zamb_Lineup>
    <has_moderator>
     <Character> … Moderator … </Character>
    </has_moderator>
    <has_expert>
     <Character> … Lebacher … </Character>
    </has_expert>
    <has_contestant>
     <Character> … User1 … </Character>
    </has_contestant>
    <has_opponent>
     <FootballTeam>
      <has_name>Brasilien</has_name>
     </FootballTeam>
    </has_opponent>
    <has_lastEvaluation>
     <Evaluation id="lastEvaluation"/>
    </has_lastEvaluation>
   </zamb_Lineup>
  </Goal>
 </setGoal>
</directionML>
```

Fig. 10. Example of a directionML message.

A precondition can also use static and dynamic character traits to constrain the choice of a move. This way, a character can refuse to answer a question from a character he is (currently) hostile towards even if he knows the answer. Game parameter conditions check the content of moves, e.g., whether the content type of a question is related to football or not. A condition can also specify constraints related to the state of the interaction, e.g., that no utterance has been made for a given time. Additionally, a probability weight can be used to assign a

choice relative weight, introducing an element of chance in the interaction. A move is selected from a set of conditional alternatives as follows: All choices with unfulfilled logical requirements or character trait conditions get removed.

This set is reduced to its members with the most specific applicable parameter conditions. Finally, a random selection is made between the remaining choices, where the chance of a member is proportional to its relative probability weight.

```
<Question>
 <has_initiator> ... </has_initiator>
 <has_addressee> ... </has_addressee>
 <has_content>
  <ListElement>
   <has_listPosition> ... </has_listPosition>
   <has_content>
    <Response>
     <has_content>
      <Parade>
       <has_agent>
        <GoalKeeper> ... </GoalKeeper>
       </has_agent>
       <has_style> ...
       </has_style>
      </Goal>
     </has_content>
    </Response>
   </has_content>
  </ListElement>
 </has_content>
</Question>
```

Fig. 11. Example dialogue act representing the contribution of a virtual character.

On the top level of the dialogue model in Fig. 8, characters are executing *Activities* to achieve the goals of the narrative. They use preconditions and postconditions like the dialogue games, and also can exist in several versions with different condition sets. Again, it is not guaranteed that an activity will succeed, since it would require complete certainty about the knowledge and cooperation of other participants. Activities achieve their communicative tasks by using single dialogue acts or playing dialogue games, or delegate the task to sub-activities that can be selected based on preconditions as described in the previous subsection. The model treats activities as black boxes that can be used to perform any kind of computation necessary to determine what games or acts to use. In its simplest form, an activity can just use a fixed sequence of acts or follow the rules of a game, which then results in scripted behavior.

Activities can be triggered by the narration engine. The narration engine can also state parameters and success conditions on the execution of activities (e.g., the football lineup in phase 2 of VirtualHuman must complete within a variable timeout). Fig. 10 shows an example of a "SetGoal"-message from the narration engine that sets a joint goal for three participants, namely the team lineup from phase two. Along with the participants, the goal specifies the roles they are to take in the activity and additional parameters, such as the opponent team, and a timeout for the activity. The SetGoal also contains an "Event" tag that refers to a slot in the activity object ("lastEvaluation"), which has the effect that the CDE controller will notify the narration engine of all evaluations of the human player's moves. When an activity terminates, the CDE controller sends a feedback message to the narration engine which may contain requested return values from the world state of the CDEs (e.g., the current score during the quiz).

5.3 The multimodal generator

The multimodal generator of a character CDE takes an ontological instance of a dialogue act(see Fig. 11) and turns it into PML syntax (see the corresponding chapter in this volume). The output of the generator contains the spoken utterance, as well as tightly synchronized nonverbal actions (gazes, adaptors, emblematic, iconic, deictic and beat gestures). In the dialogue act, some of the content may be marked as optional and can then be realized, depending on the speaker's emotional state and the discourse context. An utterance will turn out differently, 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" ... >
   <text>b) the goalkeeper saves the
    ball with his fingertips </text>
  </speak>
  <animate id="ag0"
    alignTo="s0" alignType="starts">
   <gesture refId="gazeAtUser1"/>
  </animate>
  <animate id="aa0" ...>
   <gesture refId="countTwo"/>
  </animate>
  ...
 </character>
</actions>
```

Fig. 12. Example PML output of the multimodal generator.

Since the generation takes place in real-time, we use several means to cope with time-critical aspects. First of all, generation is bound to be fast and efficient. Secondly, we estimate 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 express pensive behavior. Our multiparty scenario asks for a turn-taking approach incorporating general gazing behavior as well as actions to take and yield turns. When the generator detects that its CDE 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. Most gestures have to be synchronized with a constituent in the speech utterance. A gesture states the kind of alignment (e.g. Starting with a word) as well as the time frame during which it should be performed. Especially in the case of iconics, metaphorics, and

emblems, one can identify a gesture's meaning. This gesture should coincide with a constituent of related meaning (e.g. counting gestures are likely to co-occur with an utterance that refers to an enumeration).

VI. REALIZING REACTIVE AND DELIBERATIVE BEHAVIOR

6.1 Interpreting and Generating Turn-Taking Signals

It is crucial for a participant of a multi-party interaction to understand turn-taking signals displayed by the other participants, as well as to display appropriate signals for the other participants. Moreover, timing has a great impact on the naturalness of this behavior. A backchannel feedback that is realized only a little too late might interrupt or at least confuse the current speaker and cause a temporary break-down of the turn-taking system.

For the current version of the VirtualHuman system we focused on the reactive generation of gaze behavior. A participant that perceives, for example, the onset of a verbal contribution of another character usually (but not always) reacts by gazing at the speaker. This is realized by means of a set of specialized production rules of FADE. If appropriate, FADE directly sends a request to the multimodal generator without consulting the dialog manager. The speaker also displays gaze behavior, however, with slightly different intentions. Speakers, in turn, gaze alternately at the participants who they want to address.

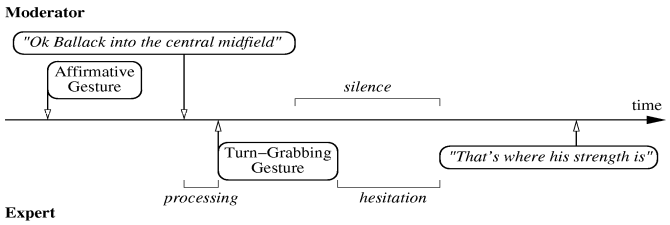

Fig. 13. Temporal diagram of a turn-request displayed by a virtual character.

Another instance of reactive turn-taking behavior is related to the process of requesting the turn. When a virtual character wants to take the turn while another character or the user is still holding the turn, it needs to signal this wish (see Fig. 13). On the technical side, this display of a turn requesting signal is managed and triggered by the multimodal generation component. First, the generator receives a request from the action planner to generate a turn but before it starts to generate and output this sentence it checks with FADE who is currently holding the speaking turn. If it is the character itself or if the turn is available, the action manager's request can be realized directly. However, if another participant holds the turn, FADE informs the generator that the floor is not available. Based on the initial generation job and the current affective state, the generator then selects appropriate actions in order to signal the turn-request to the current speaker.

6.2 Identifying the Intended Addressees

*The task of identifying the intended addressee(s) is particularly relevant for dialogue systems with more than two participants and is not required for dyadic systems. As discussed in the previous section, the intended addressee(s) are signaled in differ*ent ways by speakers. Some of the common addressing techniques are explicit while others are rather tacit and require some contextual reasoning in order to determine the intended addressee.

The following list describes the individual rules that realize the identification of the intended addressees in the VirtualHuman system:

- *Vocative*: If the utterance of the speaker contains a vocative, the participants that are denoted by the vocative are the addressees of that utterance.
- *Speaker mentions participant*: If the speaker explicitly names a participant in the utterance, this participant is most likely not the intended addressee.
- *Contextual factors 1*: If none of the previous rules can be applied, the previous addressee is the current speaker and the dialog act of the previous utterance is a subtype of Request, then take the previous speaker as addressee.
- *Contextual factors 2:* If none of the previous rules can be applied and one of the previous addressees is the current speaker, then take the previous speaker as addressee.

6.3 Realizing Deliberative Behavior

During the execution of activities, the action manager of a CDE changes between different modes in an action cycle. Since there can be more than one activity concurrently active at a time, the CDEs can be in different modes for each activity. These are called "deliberation", "initiate move" and "consume move".

In deliberation mode, the action manager examines the current state of the world, drawing inferences (i.e., information state updates) from it, and possibly adapting intentions to initiate new dialogue games to move the activity towards its goal. It is also possible to start sub-activities or sub-games to be completed before the current activity can continue. The inference procedure is influenced by the current private world state, the affective state, and the remaining conditions for goal completion. It employs a mixture of Java code snippets, action scripts and the integrated JSHOP2 planner [33], which can use the dialogue games with their preconditions and postconditions as plan operators. However, since the postconditions are not guaranteed to hold, execution monitoring must be employed in the latter case to re-check the state of the world after each step. If the action manager determines that the activity is complete, it sends a goal feedback message, possibly including additional information about the activity's final information state, to the narration engine and terminates the activity.

If a dialogue game G is currently executed, it depends on its state whether the action manager initiates or consumes a move. At each point in a dialogue game, there is a set of legal continuation moves whose initiative lies at either the initiator of the game, or its counterpart, the responder. If a communicative act is received from another participant that can be matched with such a continuation move, it is consumed by the associated activity, the information state is updated according to the semantic content of the act and the post conditions of that move, and the game is advanced. On the other hand, when a set of

moves with satisfied preconditions is available where the CDE has the initiative, it selects one of them to instantiate as detailed above. A corresponding communicative act is created and sent to the other participants, the game is advanced and the information state updated. After consuming or initiating a move, the activity returns to deliberation mode.

6.4 Resolving Spatial References

Resolving spatial references depends on the point of view the speaker takes to encode the referring expression. This point of view is called the *frame of reference* (see [17]). The frame of reference a speaker takes directly influences the selection of a particular referring expression, e.g., everything that is on my left is on the right of someone standing in front of me. Levinson distinguishes three main frames of reference: *intrinsic*, *relative* and *absolute*. When using an intrinsic frame of reference, the speaker takes the point of view of the relatum (i.e., the object that is used to locate the target object). In a relative frame of reference, the speaker takes an outside perspective (e.g., his own point of view, or that of someone else). Within an absolute frame of reference, everything is located with respect to the geographic north. While the latter frame of reference is always unambiguous the former two might introduce some ambiguities that need to be resolved.

The resolution of referring expressions involves the following aspects: (i) an up-to-date representation of the physical environment, (ii) knowledge of the currently active type of frame of reference and (iii) a mapping function that converts spatial references to locations or objects in the scene.

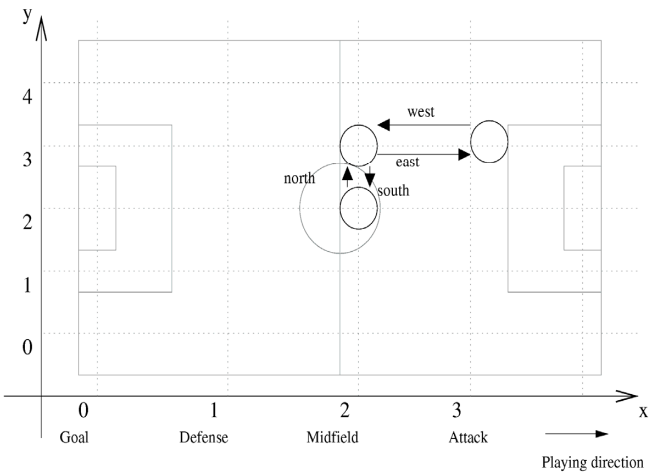

Fig. 14. Internal representation of the physical environment describing the football field.

In order to resolve spatial references, FADE first determines the currently activated physical environment and its corresponding active frame of reference and then maps the referring expression to an absolute location. If, for example, the user commands the system to "*Put Metzelder to Ballack's left*", the system first searches for the current position of the player *Ballack* in the physical environment (see Fig. 14). Then it retrieves the orientation of that player and maps the referring expression to one of the absolute identifiers. At this point we assume a currently active frame of reference of type *intrinsic*, otherwise the system would need to determine the orientation of the speaker and then compute the mapping. In any case, the mapping function takes the referring expression (*left-of*) and the orientation of the relatum *eastern*) which would result in an off-set of 1. This means, we need to go *one* neighbor feature further to get the correct neighbor given the orientation. Normally (i.e., if the player would be oriented to the north), left-of would be mapped to the western neighbor, however, in our case we need to go one neighbor further which is the northern neighbor. If the player faces westwards, the mapping function would return an off-set of 3 which means left-of is now the southern neighbor. Fig. 15 shows the configuration of the football field after the user's utterance has been processed.

Fig. 15. The configuration of the football field after processing the user utterance: "Put Metzelder to Ballack's left." See Color Plate 8.

VII. COMPARISON WITH RELATED WORK

The realization of life-like virtual characters is an emerging technology for entertainment and tutorial scenarios. VirtualHuman combines technologies from multimodal task-oriented dialogue systems and interactive narratives. It is novel in the combination of multimodality, multi-party interaction, and deep knowledge modeling. Comparable task-oriented systems do only handle two-party human computer interaction. This is also the case for SmartKom, which uses predecessors of VirtualHuman´s dialogue management modules [26]. The WITAS system has a similar task structure using a tree of activities and dialogue moves [28]. SmartWeb features a thorough ontological modeling of the domain, but only handles question-answer dialogues [33].

Interactive narratives tend to focus on the believability of the characters and restrict their autonomous reasoning, to avoid problems with actions that might disturb the story. Façade [29] is an example that creates an immersive narrative, but at the expense of the ability of the user to act out her intentions. The ambitious mission rehearsal exercise [30] has a similar setup to VirtualHuman, but does not feature rich multimodal interaction.

Also related is the emerging IN-TALE system [31] that employs a director entity controlling the story.

VIII. SUMMARY

In this paper we demonstrated how we realized the natural and flexible character-based interaction in VirtualHuman. Most important to our approach is the autonomous processing realized in the Conversational Dialogue Engine (CDE) framework. The knowledge-based approach for deliberative and reactive behavior is the basis for the successful realization of the two scenarios of VirtualHuman. The autonomous processing both enables the natural generation of nonverbal behavior of virtual characters, including turn-taking gestures.

The system was demonstrated at various occasions. Most challenging were two exhibits, one during the full duration of CeBIT 2006 in Hanover, and one at an event related to a FIFA World Cup game in Kaiserslautern. During both occasions, the general public was invited to participate in the ZAMB game. Despite the challenging acoustic environment at both locations, the system operated well and the reaction of the players and spectators was very positive. A video demonstrating the system can be found on the project's web site at www.virtual-human.org.

IX. ACKNOWLEDGMENT

We would like to thank all our colleagues in VirtualHuman. Special thanks go to our student assistants Benjamin Kempe, Ehsan Gholamsaghaee, and Mehdi Moniri. The work presented here was funded by the German Ministry for Education and Research under grant 01 IMB 01A. The responsibility for the content lies with the authors.

REFERENCES

[1] R. B. Miller. Response Time in Man-Computer Conversational Transactions, in Proceedings of the *AFIPS Fall Joint Computer Conference (volume 33)*, pp. 267-277, 1968.

[2] M. O. Riedl, C. J. Saretto and R. M. Young. Managing Interaction between Users and Agents in a Multiagent Storytelling Environment, in Proceedings of the *2nd International Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[3] Markus Löckelt, Elsa Pecourt and Norbert Pfleger. Balancing Narrative Control and Autonomy for Virtual Characters in a Game Scenario, in Mark T. Maybury, Oliviero Stock and Wolfgang Wahlster, editors, Proceedings of the *First International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN) 2005*, pp.248-252, Madonna di Campiglio, Italy, 2005.

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

[5] 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.

[6] P. McBurney and S. Parsons. Games that Agents Play: A Formal Framework for Dialogues Between Autonomous Agents. Journal of Logic, Language and Information, vol. 11, no. 3, pp. 315-334, 2002.

[7] N. Pfleger and J. Schehl. Development of Advanced Dialog Systems with PATE, in Processing of the *International Conference on Spoken Language Processing (Interspeech 2006/ICSLP)*, pp. 1778-1781, Pittsburgh, PA. , 2006.

[8] B. Kempe. PATE—a Production Rule System based on Activation and Typed Feature Structure Elements, Bachelor thesis, Department of Computer Science, Saarland University, (2004).

[9] 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)*, pp. 17–21, Trento, Italy, 2005.

[10] N. Pfleger and M. Löckelt. A Comprehensive Context Model for Multi-party Interactions with Virtual Characters, in Proceedings of the *6th International Conference on Intelligent Virtual Agents (IVA 2006)*, pp. 157-168, Marina del Rey, CA, 2006.

[11] J. Cassell, O. Torres and S. Prevost. Turn Taking vs. Discourse Structure: How Best to Model Multimodal Conversation, in Y. Wilks, editor, Machine Conversations, pp. 143-154. Kluwer, The Hague, 1999.

[12] J. Cassell. More than Just Another Pretty Face: Embodied Conversational Interface Agents, *Communications of the ACM*, vol. 43, no. 4, pp. 70-78, 2000.

[13] H. Grice. Logic and Conversation, in P. Cole and J. Morgan, editors, Syntax and Semantics, vol. 3, pp. 42-58. Academic Press, New York, NY, 1975.

[14] A.-B. Stenströom. An Introduction to Spoken Interaction, Longman Group, London, UK, 1994.

[15] V. H. Yngve. On Getting a Word in Edgewise, in Papers from the 6th Regional Meeting of the Chicago Linguistics Society, pp. 567-577, Chicago Linguistics Society, 1970.

[16] H. Clark and T. B. Carlson. Hearers and Speech Acts. Language, vol. 58, pp. 332-373, 1982.

[17] S. C. Levinson. Space in Language and Cognition, Press Syndicate of the University of Cambridge, Cambridge, UK, 2003.

[18] H. Sacks, E. A. Schegloff and G. Jefferson. A Simplest Systematics for the Organization of Turn-taking for Conversations. Language, vol. 50, no. 4, pp. 696-734, 1974.

[19] N. Pfleger. Context-based Multimodal Interpretation: An Integrated Approach to Multimodal Fusion and Discourse Processing, Phd thesis, to appear, 2007.

[20] Annika Flycht-Eriksson. A Survey of Knowledge Sources in Dialogue Systems, *Electronic Transactions on Articial Intelligence*, vol. 3, no. D, pp. 5-32, 1999.

[21] G. Core Mark and F. Allen James. Coding Dialogues with the DAMSL. Annotation Scheme. in R. Traum David, editor, Working Notes: AAAI Fall.
Symposium on Communicative Action in Humans and Machines, pp. 28-35, Menlo Park, CA, 1997. American Association for Articial Intelligence.

[22] C. Mann William. Dialogue Games: Conventions of Human Interaction, in *Argumentation*, no. 2, pp. 512-532, 1988.

[23] Lauri Carlson. Dialogue Games. Synthese Language Library. D. Reidel Publishing Company, Dordrecht, The Netherlands, 1983.

[24] W. Wahlster(editor). SmartKom: Foundations of Multimodal Dialogue Systems, Cognitive Technologies Series, Springer, Berlin, Germany, 2006.

[25] M. Löckelt. A Flexible and Reusable Framework for Dialogue and Action Management in Multi-party Discourse, PhD thesis, 2007, to appear.

[26] O. Lemon, A. Gruenstein and S. Peters. Collaborative Activities and Multi-tasking in Dialogue Systems-towards Natural Dialogue with Robots, *Traitement automatique des langues*, vol. 43, no. 2, pp. 131-154, 2002, Special issue on dialogue.

[27] M. Mateas and A. Stern. Facade: An Experiment in Building a Fully-realized Interactive Drama, in *Game Developers Conference*, *Game Design track*, San Jose, CA, USA, 2003.

[28] W. R. Swartout, J. Gratch, R. Hill and et al. Towards Virtual Humans, in *Working Notes of the AAAI Fall Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research*, 2004.

[29] M. O. Riedl and A. Stern. Believable Agents and Intelligent Scenario Direction for Social and Cultural Leadership Training, in Proceedings of the *3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pp. 195-206, Darmstadt, Germany, 2006.

[30] D. Nau, T. -C. Au, O. Ilghami and et al. SHOP2-An HTN Planning System, *Journal of Artificial Intelligence Research*, no. 20, pp. 379-404, 2003.

[31] Norbert Reithinger, Simon Bergweiler, Ralf Engel, Gerd Herzog, Norbert Pfleger, Massimo Romanelli and Daniel Sonntag. A Look Under the Hood-Design and Development of the First SmartWeb System Demonstrator, In: *Proc. ICMI 2005*, Trento (Italy). pp. 159-166.

Markus Löckelt was born in Zweibrücken, Germany. He studied Computer Science at the University of Saarbrücken and received his Diploma degree in 2000.
He is working as a research scientist at the Intelligent User Interfaces lab at the German Research Center for Artificial Intelligence (DFKI) in Saarbrücken. His research interest is on improving human-computer interaction; he has done work in the area of multimodal dialogue systems, in particular dialogue management, action planning, and cognitive modeling of computer characters. He participated in the development of the Oz programming language and several national and international projects, including Verbmobil, SmartKom, MIAMM, and VirtualHuman.

Norbert Pfleger was born in Bad Dürkheim, Germany. He studied Computational Linguistics at the University of Saarbrücken and received his Diploma degree in 2002. He is working as a research scientist at the Intelligent User Interface lab at DFKI Gmbh, Saarbrücken. His main research interests are in the area of multimodal dialogue systems, multimodal fusion and discourse processing and conversational human-computer interaction. He participated in various national and international projects like SmartKom, COMIC, SmartWeb, VirtualHuman, and i2home.

Norbert Reithinger was born near Nuremberg, Germany and went to the University of Erlangen-Nuremberg. He majored in computer science and in 1983 earned his *Diplom-Informatiker* degree with excellence
After a short stay with the research lab of a computer company in Nuremberg, he joined the artificial intelligence group of Prof. Wahlster at Saarland University as a research assistant. He was responsible for the area of multimodal generation in various projects. In 1991 he earned his computer science doctorate *magna cum laude*. 1993 he joined DFKI GmbH, Saarbrücken, as a project manager for discourse processing in the Intelligent User Interfaces department. He was responsible scientific manager for various large scale projects like Verbmobil, SmartKom, SmartWeb, and VirtualHuman, which were funded by the German Ministry for Education and Research. He also participated in various international projects, funded by the European Union and others. In 2000, he was honored as a DFKI Research Fellow for his contribution to AI.
Dr. Reithinger was co-organizer and member of the program committees of numerous conferences and workshops. He co-authored more than 50 articles in journals, books and refereed conferences. He is member of ACM and the IEEE computer society.