Multi-Party Interaction With Self-Contained Virtual Characters

Markus Löckelt
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
loeckelt@dfki.de

Norbert Pfleger
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
pfleger@dfki.de

Abstract

We describe a layered approach for coordinating interactions of human users and virtual characters in a multi-modal dialogue system.

1 Introduction

Contributions in face-to-face conversations convey not only propositional but also interactional content. Interactional information contributes to the structural organization of the conversation. It regulates the transitions between speaker and hearer, helps to avoid overlapping speech, and supports the identification of intended addressees of a contribution. We illustrate some aspects of multi-party discourse by an example of a quiz dialog. It includes a virtual moderator, a human user (Chris) and a virtual character (Frank):[1]

(1) *Moderator*: [⊙ both candidates] "The next question: Who scored the last goal at the world championship 1990?"

(2) *Chris*: [⊙ moderator] "Franz Beckenbauer"

(3) *Moderator*: [⊙ Chris; Frank shakes head and raises finger] "well, no . . . " [⊙ Frank]

(4) *Frank*: [⊙ Chris] "Oh dear, no" [⊙ moderator] "He was the coach." [Moderator nods] "The correct answer is Andreas Brehme."

(5) *Moderator*: [⊙ Frank] "Yes, that will be one point" [points at Frank] "for Frank!"

[1] ⊙ means "looks at".

Following (Duncan, 1972), conversations are organized in turns where participants coordinate their actions in order to achieve a smooth turn exchange. This takes place by means of a rule based signaling of what the individual participants want to do next. A hearer wanting to take the speaking turn can e. g. signal this by an upraised finger, sometimes accompanied by an audible intake of breath, see the beginning of turn (3). Even though there are several other ways to encourage a speaker to finish talking, a speaker who perceives these signals is able to infer the intention of the hearer and react accordingly (the moderator yielding the turn at the end of (3)). We model dialog exchange as being structured in rule-governed game-like sequences of dialog moves. When being addressed in a game move, a character has a choice of legal reply or followup moves, among which it selects one based on the current situation and its current goals. In turn (4), Frank determined that Chris has answered incorrectly. He decides to take over the pending response move; the moderator agrees by gazing at him.

2 Conversational Dialog Engines

Modules called Conversational Dialog Engines (CDEs) interact to realize the dialog capabilities of our system. All actions of a single virtual character are controlled by a dedicated CDE representing it. Human users of the system are also represented by their own CDEs, resulting in two classes of CDEs: CDEs creating the behavior of the virtual characters (*Character-CDEs*) and CDEs recognizing and analyzing the contributions of a hu-

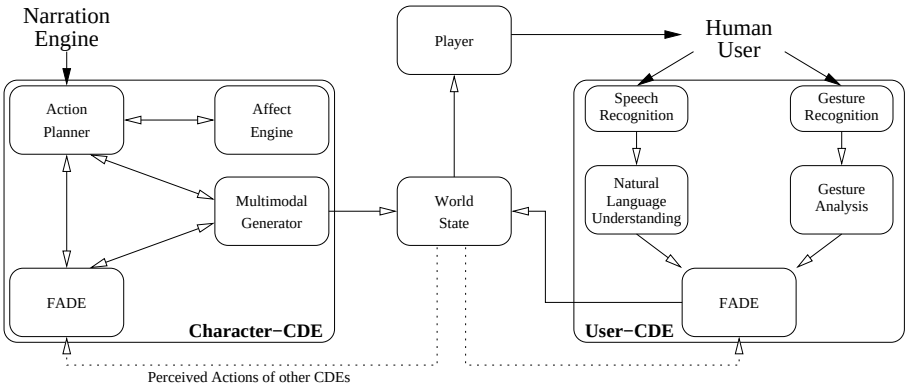

Figure 1: Components of Character-CDEs and User-CDEs

man user (*User-CDEs*).

Both CDE classes perceive, process and generate all character actions represented in the same ontology based data format. An abstraction of the actual state of the world can be perceived and manipulated by the CDEs. The abstract world state is interpreted by the 3D player to produce the actual visualization. A participant's contribution to an interaction is represented by an instance of a dialog act, e. g., *request*, and an embedded semantic representation of that utterance. Also, the internal knowledge of the virtual characters is represented in terms of this ontology. The Character-CDEs (as depicted on the left side of Fig. 1) consist of a fusion and discourse modeling engine (FADE), an affect engine, an action planner, and a multimodal generation component. In contrast, a User-CDE basically serves as a perception and translation module converting the actions of a human user into the ontology based representation the CDEs employ to communicate. A User-CDE comprises an ASR, a natural language understanding component, a gesture recognizer and analyzer, and a fusion and discourse modeling engine (FADE).

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

The local turn context provides a comprehensive model of the current conversational situation. It models all participating co-interactors with respect to their current role and their respective internal states. This enables a CDE to interpret the perceived interactional contributions with respect to the current state of the conversation. One example would be a virtual character raising a finger into the visual field of another agent. This could mean 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 discourse history keeps track of the ongoing discourse and provides a comprehensive history of the individual discourse contributions. This enables the generation component to produce referring or elliptical expressions.

Action Planning The action planner is the deliberative unit for a character that devises the actions that are necessary to stepwise achieve its narrative goals. Each action planner operates as an independent agent whose deliberative process roughly follows a cycle where the narration engine indicates plot goals for one or more characters, the CDEs enact dialog moves to fulfill them, and report back success or failure. When additional processes are spawned, this can happen either directly consequential of the original goal (e. g. an obligation to answer a question) or as

a result of the internal state of characters, (e. g. complaining that questions are too difficult). Dialog management usually adopts one of several established approaches, 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 (see (Larsson, 2002)). The suitability of an approach depends on the characteristics of the application. The interactions for a simple ticket-ordering application might map quite naturally to form-filling fixed data structures, but complexer scenarios call for more versatile interactions and representations. Our domain shows mixed characteristics, and we also use a combination of methods. Our scenario contains elements that have little variation and can be scripted (e. g. greetings), but the user interaction and autonomous behavior by the virtual characters also allow for flexible deviations interweaved into the story controlled by the narration engine. Both types of tasks share a common task model, the process, but the dialog games can be initiated using either a finite-state model, or a plan-based approach which is adapted from the system described in (Wahlster, 2003) to work with multi-party dialogs.

3 Three Levels of Processing

Purely Unconscious Behavior The lowest level of behavior comprises reactive actions of the characters. If, e. g., a character perceives another character has just started to speak, it should react by gazing at the speaker. Another example is *idle behavior* a character displays when there is nothing else to do (e. g., short intakes of breath or self-adaptors). Idle behavior can be willfully suppressed if participants in a conversation want to show inattentiveness they can refuse to gaze at the speaker, and is triggered by FADE or the Affect Engine. FADE monitors the perceived changes in the environment and ensures that the character displays proper behavior. The Affect Engine in turn controls the idle behavior and facial expressions of the characters with respect to emotional state (e. g. angry facial expression). The respective actions of a virtual character are triggered by interfaces to the multimodal generation component (see Fig. 1).

Semi-Conscious Behavior The semi-conscious 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 a *transition relevance place* (TRP) (the point within a turn at which an addressee can take over or can display backchannel feedback; see (Sacks et al., 1974)) 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 TRP in the speaker's turn. It also needs to monitor the current status of the natural language understanding.

Another instance of semi-conscious behavior is related to the process of requesting the turn as displayed by Frank in example turn (3). Here Frank knows the answer but the moderator is holding the turn at the moment (to get the turn Frank raises his index-finger). When the moderator notices, he yields the turn to Frank by stopping to speak and looking at him. 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 turn (4) but before it starts to generate and output this sentence it checks with FADE who is holding the speaking turn. If it is the character itself, the action planner's request can be realized directly. However, in this case, FADE informs the generator that the moderator is holding the turn. Based on the initial generation job and the current affective state, the generator selects appropriate actions.

Deliberative Behavior The top level of behavior control executes *processes* to achieve goals, which can be triggered externally, e. g. by a narrative control instance. Characters will also autonomously adopt goals to fulfill social obliga-

tions, e. g. conforming to a dialog game, or to honor internal (e. g. emotional) state. Deliberative behavior itself decomposes into three levels: Dialog acts, dialog games, and processes.

The lowest level comprises a set of *dialog acts*, the atomic communicative units between CDEs. We use a set of acts similar to those in (Poesio and Traum, 1998); examples are *opening* (greeting), *info-request* and *answer*. The propositional content of dialog acts refers to ontological object instances. The dialog acts themselves do not carry interlocutor obligations. *Dialog games* form the middle level. They specify exchanges of dialog act moves governed by rules, and the alternative moves legal in a situation. An *InformationSearch* game, for example, states that an initial *info-request* may allow for an *answer* making an assertion in response, a statement that one does not know the answer, or a refusal to answer. Dialog games can be combined by several operations, e. g. appended or nested, to form composite games (see e. g. (McBurney and Parsons, 2002)). Dialog game specifications need not be the same across characters (e. g. an unfriendly character need not know how to respond to an *opening*, and may ignore it). If a character participates in a game, it accepts the obligation to make only legal moves according to (its own version of) the rules of the game. The conventional part of the game definition—stating which moves are legal to make at any point of the game—is shared among all characters, and takes the form of a finite-state-automaton, where transitions are labeled with preconditions and postconditions. From the narration engine's point of view, a *process* appears as a parametrized black box. A *QuizQuestion* process, for example, would be parametrized by (i) instances of ontological objects filling *roles* specifying the moderator, the contestants, the subject, and possibly the presentation style, (ii) narrative constraints, e. g. a timeout, (iii) the content of the dialog history, (iv) the character's private world view, and (v) a set of *traits* for the character, which can be static (e. g., an intelligence value) or dynamic (e. g., the affective state). The process also needs a method of evaluating the appropriateness of answers. As stated before, the internal process implementation can use a finite-state representation for simpler tasks, or be plan-based if more flexibility is necessary. A process goal from the narration engine can result in several sub-processes for other participating characters, as in our example. The contestants are obliged to answer the moderator: The question in turn (1) is not directed towards a specific character. Any dialog participant can decide to join the game, and *Chris* does so first.

4 Conclusion

Our four-year project has passed its halfway point, for which we completed a demonstrator system implementing our first scenario. In the second project phase, more than one human user will be able to simultaneously participate in the dialog, using separated input devices.

Acknowledgements

This research is funded by the German Ministry of Research and Technology (BMBF) under grant 01 IMB 01A (VirtualHuman).

References

Starkey Duncan. 1972. Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Department of Linguistics, G¨oteborg University, Sweden.

Peter McBurney and Simon Parsons. 2002. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3):315–334, Summer.

Massimo Poesio and David Traum. 1998. Towards an Axiomatization of Dialogue Acts. In J. Hulstijn and A. Nijholt, editors, *Proceedings of TWENDIAL workshop*, Enschede.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A Simplest Systematics for the Organization of Turn-Taking for Conversations. *Language*, 50(4):696 – 734.

Wolfgang Wahlster. 2003. Towards Symmetric Multimodality: Fusion and Fission of Speech, Gesture, and Facial Expression. In A. G¨unter, R. Kruse, and B. Neumann, editors, *Proceedings of the 26th German Conference on Artificial Intelligence*, pages 1 – 18. Springer.