Enhancing Animated Agents in an Instrumented Poker Game

Marc Schröder[1], Patrick Gebhard[1], Marcela Charfuelan[1], Christoph Endres[1], Michael Kipp[1], Sathish Pammi[1], Martin Rumpler[2], and Oytun Türk[1]

[1] DFKI, Saarbrücken and Berlin, Germany, firstname.lastname@dfki.de
[2] FH Trier, Umwelt-Campus Birkenfeld, Germany, m.rumpler@umwelt-campus.de

Abstract. In this paper we present an interactive poker game in which one human user plays against two animated agents using RFID-tagged poker cards. The game is used as a showcase to illustrate how current AI technologies can be used for providing new features to computer games. A powerful and easy-to-use multimodal dialog authoring tool is used for modeling game content and interaction. The poker characters rely on a sophisticated model of affect and a state-of-the art speech synthesizer. Through the combination of these methods, the characters show a consistent expressive behavior that enhances the naturalness of interaction in the game.

1 Motivation

An application area with growing relevance for AI technologies are computer games. Specifically, driving the behavior of non-player characters raises a whole range of challenges, such as interaction design, emotion modeling, figure animation, and speech synthesis [1].

Some research projects address these problems in combination, such as the interactive drama game Façade [2] or the Mission Rehearsal Exercise Project [3]. However, relevant research is also being carried out in a range of individual disciplines, including believable facial, gesture and body animation of virtual characters [4], modeling personality and emotion [5], expressive speech synthesis [6] and control mechanisms for story and interaction [7].

In the project IDEAS4Games, we investigate how modern AI technologies can help to improve the process of creating computer games with interactive expressive virtual characters. Based on the experience of a computer game company (RadonLabs, Berlin), we identified four main challenges in creating computer games: (i) fast creation of game demonstrators; (ii) localization of game content; (iii) easy interaction; and (iv) consistent quality. We address these challenges by combining, in an integrated system, flexible multimodal dialog and story modeling, sophisticated simulation of affect in real time, expressive synthesis, and rendering of 3D virtual characters.

We explain the concept of the integrated game demonstrator before providing some details about the individual component technologies.

2 Poker Demonstrator

In order to demonstrate that it is possible to meet the requirements stated above, we created a poker computer game and presented it at the CeBIT 2008 fair to a large number of people to collect initial feedback.

By using real poker cards with unique RFID tags, a user can play draw poker [8] against the two 3d virtual characters Sam and Max (see Fig. 1). Sam is a cartoon-like, friendly looking character, whereas Max is a mean, terminator-like robot character. Both are rendered by the open-source 3d visualization engine Horde3D [9]. The human user acts as the card dealer and also participates as a regular player.

As shown in Fig. 1, we use a poker table which shows three areas for poker cards: one for the user and one for each virtual character. These areas of the table are instrumented with RFID sensor hardware, so that the game logic can detect which card is actually placed at each specific position. A screen at the back of the poker table displays an interface which allows users to select their actions during the game, using a computer mouse. These include general actions, such as playing or quitting a game, and poker game actions: bet a certain amount of money, call, raise, or fold. This screen also shows the content relevant for the poker game: the face of the user's cards, the number of Sam's and Max's cards respectively, all bets, and the actual money pot. The two virtual characters are shown above this interaction screen on a second 42" monitor.

When a user approaches the poker table and initiates a game, Sam and Max explain the game setup and the general rules. In a next step the user has to deal the cards. During the game the animated agents react to events, notably when

Fig. 1. Poker Demonstrator at the CeBIT 2008 fair

the user deals or exchanges their cards. Time is also considered – for example, Sam and Max start complaining if the user deals the cards too slowly, or they express their surprise about erratic bets.

Different poker algorithms are used to match Sam's and Max's individual character style. Sam, the human-like poker player, uses a rule based algorithm, whereas Max, the robot, relies on a brute-force algorithm that estimates a value for each of the 2.58 million possible combinations of five poker cards.

Based on game events, the affect of each character is computed in real-time and expressed through the character's speech and body. The richly modeled characters, as well as the easy interaction with the game using real poker cards, have stirred up a lot of interest in the CeBIT 2008 audience (see Fig. 1).

3 Enhancing Agents

In the following, we explain the key properties of the technologies used in the demonstrator.

3.1 Interaction and Story Authoring

The behavior of the two virtual players has been modeled with the authoring tool SceneMaker, which separates dialog content and narrative structure [10].

Dialog content is organized in *scenes* – pieces of contiguous dialog. Scenes are defined in a multimodal script that specifies the text to be spoken as well as the agents' verbal and nonverbal behavior. The utterances can be annotated with dialog act tags that influence the computation of affect (see below). In addition, system commands (e.g., for changing the camera position) can be specified. Scenes are created by an author with standard text processing software. The major challenge when using scripted dialog is variation. The characters must not repeat themselves because this would severely impact their believability. For this purpose we use blacklisting: once a scene is played, it is blocked for a certain period of time (e.g., five minutes), and variations of this scene are selected instead. For each scene, several variations can be provided that make up a *scene group*. In our poker scenario there are 335 scenes organized in 73 scene groups. The number of scenes in a scene group varies between two and eight scenes, depending on how much variation is needed.

The narrative structure – the order in which the individual scenes are played – is defined by the *sceneflow*. Technically, the sceneflow is modeled as a hypergraph that consists of nodes and edges (transitions). *Supernodes* contain subgraphs. Each node can be linked to one or more scenes and scene groups. Different branching strategies (e.g. logical and temporal conditions as well as randomization) can be used by specifying different edge types [10]. The sceneflow is modeled using the sceneflow editor, our graphical authoring tool that supports authors with drag'n'drop facilities to *draw* the sceneflow by creating nodes and edges.

At runtime the sceneflow graph is traversed by selecting nodes and edges based on the current game state and the actions of the three players. The scenes that are selected and executed during such a traversal control the multimodal

behavior of the virtual agents. Transitions in the sceneflow are triggered either by the players' actions or as a result of context queries. In both cases sceneflow variables used for conditional branching are updated by an event handler.

Each time the user places or removes a card, an event is generated and sent to the poker event handler. The same happens when the user chooses an action (bet, call, raise, or fold) by pressing the respective button on the graphical user interface (see Fig. 1). The event handler receives these low-level events and updates the data model representing the game state as well as the graphical user interface. It also analyzes the situation and generates higher level events, e.g., that all cards have been removed from the table or that the user has changed the cards of a player in the drawing phase. At the end of this process, it updates the respective sceneflow variables, which may enable transitions in the sceneflow and trigger the next scenes.

Apart from scenes and scene groups the author can also attach commands to a node. These commands are executed by the sceneflow interpreter each time the node is visited. There are commands that modify the game state (e.g. selecting the next player after a scene has been played in which one of the players announces that he drops out) and commands that access the poker logic to suggest the next action (e.g. deciding which cards should be changed in the drawing phase and which action the virtual players should perform in the betting phase).

4 Affect Model

For the affect computation in real-time, we rely on ALMA, a computational model of affect [11]. It provides three affect types as they occur in human beings: (1) *emotions* reflect short-term affect that decays after a short period of time; (2) *moods* reflect medium-term affect, which is generally not related to a concrete event, action or object; and (3) *personality* reflects individual differences in mental characteristics and affective dispositions.

ALMA implements the cognitive model of emotions developed by Ortony, Clore, and Collins (OCC) [12] combined with the *BigFive* model of personality [13] and a simulation of mood based on the PAD model [14]. The three levels are interrelated: personality defines a default mood, and influences the intensities of different emotions; emotions as short term events influence the longer-term mood; and the mood, in turn, amplifies or dampens the intensities of emotional reactions to events.

ALMA enables the computation of 24 OCC emotions with *appraisal tags* [15] as input. Elicited emotions influence an individual's mood. The higher the intensity of an emotion, the higher the particular mood change. A unique feature is that the current mood also influences the intensity of emotions. This simulates, for example, the intensity increase of *joy* and the intensity decrease of *distress*, when a individual is in an *exuberant* mood. Mood is represented by a triple of the mood traits pleasure (P), arousal (A), and dominance (D). The mood's trait values define the mood class. If, for example, every trait value is positive (+P,+A,+D), the mood is *exuberant*.

The current mood and emotions, elicited during the game play, influence the virtual poker characters' affective behavior. *Breathing* is related to the mood's arousal and pleasure values. For positive values, a character shows fast distinct breathing. The breathing is slow and faint for negative values. *Speech quality* is related to the current mood or an elicited emotion. In *relaxed* mood, a character's speech quality is *neutral*. In *hostile* or *disdainful* mood or during negative emotions, it is *aggressive*, in *exuberant* mood or during positive emotions it is *cheerful*. In any other mood, the speech quality is *depressed*.

Initially, Sam's and Max's behavior is nearly identical, because they have the same default *relaxed* mood that is defined by their individual personality. However, Sam is slightly more *extroverted*, so his mood tends to become more *exuberant*. Max, on the other side, has a tendency to be *hostile*. This is caused by his negative *agreeableness* personality definition. This disposition drives Max towards negative mood (e.g. *hostile* or *disdainful*) and negative emotions faster than Sam.

In the poker game, events and user actions influence the selection of scenes and their execution. As a consequence the actions of Sam and Max as well as their appraisal of the situation changes. For example, if Sam loses the game because he has bad cards, a related scene will be selected in which he verbally complains about that. An *appraisal tag* [**BadEvent**] used in the scene lets Sam appraise the situation as a bad event during the scene execution. In the affect model, this will elicit the emotion *Distress*, and if such events occur often, they will lead to a *disdainful* or *hostile* mood. In general, appraisal tags can be seen as a comfortable method for dialog authors to define on an abstract level input for a affect computation module. For doing this, no conceptual knowledge about emotions or mood is needed.

The poker game scenario covers the simulation of all 24 OCC emotions. For example, prospect-based emotions such as *hope*, *satisfaction*, or *disappointment* can be triggered during the card change phases – Sam and Max sometimes utter their expectation of getting good new cards. Those utterances also contain the appraisal tag [**GoodLikelyFutureEvent**]. As a consequence, *hope* is elicited. After all cards have been dealt out, Sam's and Max's poker engine compares the current cards with the previous ones. Depending on the result either the [**EventConfirmed**] or the [**EventDisconfirmed**] appraisal tag is passed to the affect computation. In the first case (cards are better), *satisfaction* is elicited, otherwise *disappointment*. Appraisals leading to all other emotions are included in the scene logic in a similar way.

5 Expressive Synthetic Voices

Convincing speech generation is a necessary precondition for an animated agent to be believable. That is true especially for a system featuring emotional expressivity. To address this issue, we have investigated the two currently most influential state-of-the-art speech synthesis technologies: unit selection synthesis and

statistical-parametric synthesis. A suitable speech synthesizer should produce speech of *reliable quality* which at the same time shows a *natural expressivity*.

Unfortunately, in state-of-the-art speech synthesis technology, the two criteria are difficult to reconcile. Unit selection [16] can reach close-to-human naturalness in limited domains, but it suffers from unpredictable quality in unrestricted domains. Statistical-parametric synthesis [17] has a very stable synthesis quality, but typically sounds "muffled" because of the excessive smoothing involved in training the statistical models.

In IDEAS4Games, we investigated two methods for approximating the criteria formulated above. For our humanoid character, Sam, we created a custom unit selection voice with a "cool" speaking style, featuring high quality in the poker domain and some emotional expressivity; for our robot-like character, Max, we used a statistical-parametric voice, and applied audio effects to modify the sound to some extent in order to express emotions.

Sam's voice was carefully designed as a domain-oriented unit selection voice [18]. Domain-oriented voices sound highly natural within a given domain; they can also speak arbitrary text, but the quality outside the domain will be seriously reduced. We designed a recording script for the synthesis voice, consisting of a generic and a domain-specific part. We used a small set of 400 sentences selected from the German Wikipedia to cover the most important diphones in German [19]. In addition, about 200 sentences from the poker domain were recorded, i.e. sentences related to poker cards, dealing, betting, etc. The 600 sentences of the recording script were produced by a professional actor in a recording studio. The speaker was instructed to utter both kinds of sentences in the same "cool" tone of voice. In a similar way, the same speaker produced domain-oriented voice databases for a *cheerful*, an *aggressive* and a *depressed* voice. We built a separate unit selection voice for each of the four databases, using the open-source voice import toolkit of the MARY text-to-speech synthesis platform [20].

In our application, most of Sam's utterances are spoken with the neutral poker voice. As the voice database contains many suitable units from the domain-oriented part of the recordings, the poker sentences generally sound highly natural. Their expressivity corresponds to the "cool" speaking style realized by our actor. Selected utterances are realized using the *cheerful*, *aggressive*, and *depressed* voices. For example, when Sam loses a round of poker, he may utter a frustrated remark in the *depressed* voice; if the affect model predicts a positive mood, he may greet a new user in a *cheerful* voice, etc.

The voice of the robotic character, Max, uses statistical-parametric synthesis [21] which we incorporated into our MARY TTS platform [22]. A voice is created by training the statistical models on a speech database. After training, the original data is not needed anymore; at runtime, speech is generated from the statistical models by means of a vocoder. HMM-based synthetic speech typically sounds muffled due to the averaging in the statistical models as well as the vocoding, but the quality is largely stable, independently of the text spoken.

Expressivity for Max' voice is performed using audio effects to modify the generated speech. At the level of the parametric input to the vocoder, we can

modify the pitch level, pitch range and speaking rate. In addition, we apply audio signal processing algorithms that modify the generated speech signal, using linear predictive coding (LPC) techniques. In this framework, our code provides a vocal tract scaler, which can simulate a longer or shorter vocal tract; a whisper component, adding whisper to the voice; a robot effect, etc. Expressive speech is generated by combining various effects – for example, *aggressive* speaking style is simulated by a lengthened vocal tract, lowered pitch, increased pitch range, faster speech rate, and slightly whisperised speech.

6 Conclusion

We have presented an agent-based computer game employing RFID-tagged poker cards as a novel interaction paradigm. It is built around the integration of three components: a powerful and easy-to-use multimodal dialog authoring tool, a sophisticated model of affect, and a state-of-the-art speech synthesizer.

In our authoring approach, the separation of dialog content and narrative structure makes it possible to modify these two aspects independently and without expert knowledge.

In speech synthesis, we have illustrated the trade-off between reliable quality and naturalness, and have offered two partial solutions. We have created a very high quality unit selection voice for our character Sam, but the quality comes at the price of high effort and limited flexibility. The HMM-based voice used for our character Max is flexible so that it can speak arbitrary text reliably, but the naturalness is limited.

Our computational model of affect is integrated with the dialog script through the use of appraisal tags which can be authored without any deep knowledge about a model of emotion or mood. The continuous computation of short-term emotions and medium-term moods allows for a smooth blending of different aspects of affective behavior that help to increase the believability and the expressiveness of virtual characters.

Overall, we have shown how state-of-the-art research approaches can be combined in an interactive poker game to show new possibilities for the modeling of expressive virtual characters and for future computer game development.

Acknowledgements .

The work reported here was supported by the EU ProFIT project IDEAS4Games (EFRE program) and by the DFG project PAVOQUE.

References

1. Gratch, J., Rickel, J., André, E., Cassell, J., Petajan, E., Badler, N.I.: Creating interactive virtual humans: Some assembly required. IEEE Intelligent Systems **17** (2002) 54–63
2. Mateas, M., Stern, A.: Façade: An experiment in building a fully-realized interactive drama. In: Game Developers Conference, Game Design Track. (2003)

3. Swartout, W., Gratch, J., Hill, R., Hovy, E., Marsella, S., Rickel, J., Traum, D.: Toward virtual humans. AI Magazine **27** (2006) 96–108
4. Martin, J.C., Niewiadomski, R., Devillers, L., Buisine, S., Pelachaud, C.: Multimodal complex emotions: Gesture expressivity and blended facial expressions. International Journal of Humanoid Robotics, Special Edition "Achieving Human-Like Qualities in Interactive Virtual and Physical Humanoids" (2006)
5. de Rosis, F., Pelachaud, C., Poggi, I., Carofiglio, V., de Carolis, B.: From Greta's mind to her face: Modelling the dynamics of affective states in a conversational embodied agent. Int. Journal of Human Computer Studies **59** (2003) 81–118
6. Schröder, M.: Emotional speech synthesis: A review. In: Proceedings of Eurospeech 2001. Volume 1., Aalborg, Denmark (2001) 561–564
7. Prendinger, H., Saeyor, S., Ishizuk, M.: MPML and SCREAM: Scripting the bodies and minds of life-like characters. In: Life-like Characters – Tools, Affective Functions, and Applications. Springer (2004) 213–242
8. Wikipedia: Draw Poker, http://en.wikipedia.org/wiki/Draw_poker. (2008)
9. Schulz, M.: Horde3D – Next-Generation Graphics Engine. Horde 3D Team, http://www.nextgen-engine.net/home.html. (2006–2008)
10. Gebhard, P., Kipp, M., Klesen, M., Rist, T.: Authoring scenes for adaptive, interactive performances. In: Proc. of the 2nd Int. Joint Conference on Autonomous Agents and Multi-Agent Systems, ACM (2003) 725–732
11. Gebhard, P.: ALMA - a layered model of affect. In: Proc. of the 4th Int. Joint Conference on Autonomous Agents and Multiagent Systems, ACM (2005) 29–36
12. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge, MA (1988)
13. McCrae, R., John, O.: An introduction to the five-factor model and its applications. Journal of Personality **60** (1992) 175–215
14. Mehrabian, A.: Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. Current Psychology: Developmental, Learning, Personality, Social **14** (1996) 261–292
15. Gebhard, P., Kipp, K.H.: Are computer-generated emotions and moods plausible to humans? In: Proc. of the 6th International Conference on Intelligent Virtual Agents (IVA 2006), Marina Del Rey, California, Springer (2006) 343–356
16. Hunt, A., Black, A.W.: Unit selection in a concatenative speech synthesis system using a large speech database. In: Proceedings of ICASSP 96. Volume 1., Atlanta, Georgia (1996) 373–376
17. Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T.: Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In: Proceedings of Eurospeech 1999, Budapest, Hungary (1999)
18. Schweitzer, A., Braunschweiler, N., Klankert, T., Möbius, B., Säuberlich, B.: Restricted unlimited domain synthesis. In: Proc. Eurospeech 2003, Geneva (2003)
19. Hunecke, A.: Optimal design of a speech database for unit selection synthesis. Diploma thesis, Universität des Saarlandes, Saarbrücken, Germany (2007)
20. Schröder, M., Hunecke, A.: Creating German unit selection voices for the MARY TTS platform from the BITS corpora. In: Proc. SSW6, Bonn, Germany (2007)
21. Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A., Tokuda, K.: The HMM-based speech synthesis system version 2.0. In: Proc. of ISCA SSW6, Bonn, Germany (2007)
22. Charfuelan, M., Schröder, M., Türk, O., Pammi, S.C.: Open source HMM-based synthesisser for the MARY TTS platform. In: Proceedings of the 16th European Signal Processing Conference (EUSIPCO 2008). (2008) submitted.