

A Flexible Platform for Building Applications with Life-Like Characters

Thomas Rist

DFKI GmbH, IUI Lab
Saarbrücken, Germany
rist@dfki.de

Elisabeth André

Augsburg University, CS Dept.
Augsburg, Germany
andre@informatik.uni-augsburg.de

Stephan Baldes

DFKI GmbH, IUI Lab
Saarbrücken, Germany
baldes@dfki.de

ABSTRACT

During the last years, an increasing number of R&D projects has started to deploy life-like characters for presentation tasks in a diverse range of application areas, including, for example, E-Commerce, E-learning, and help systems. Depending on factors, such as the degree of interactivity and the number of the deployed characters, different architectures have been proposed for system implementation. In this contribution, we first analyse a number of existing user interfaces with presentation characters from an architectural point of view. We then introduce the MIAU platform and illustrate by means of illustrated generation examples how MIAU can be used for the realization of character applications with different conversational settings. Finally, we sketch a number of potential application fields for the MIAU platform.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: Interaction styles

General Terms: Human Factors

Keywords

interface agents, conversational embodied characters, intelligent systems for multimedia presentation

1. INTRODUCTION

The last decade has seen a general trend in HCI to make human-computer dialogue more similar to human-human dialogue. Computers are ever less viewed as tools and ever more as partners or assistants to whom tasks may be delegated. Trying to imitate the skills of human presenters, some R&D projects have begun to deploy embodied, life-like characters (also called animated agents) in wide range of different application areas including e-Commerce, entertainment, personal assistants, training and electronic learning environments.

Since it is unpractical or even not feasible for many application areas to control synthetic characters by hand, e.g., by relying on manually authored scripts, AI-based approaches - especially plan-based approaches - for automated character control are becoming increasingly popular in the field.

Looking at past and current projects conducted at DFKI's IUI Lab automated character control raises two basic issues from the point of view of system implementation. Firstly, automated scripting can be done from the perspective of a director who decides on what a character or a team of characters will do, i.e., how to present information about a certain subject matter, how to react to events in the character's environment, and how to respond to user input if supported. In contrast, one may adopt a character-centric, self-scripting approach. In this case, the character needs to have its own knowledge on how to behave in a reasonable way. Secondly, there is the issue of centralised versus distributed behaviour planning.

Our group has built various systems that deploy embodied presentation agents. Apart from different application fields and variations in the audio-visual appearance of the deployed characters, our previous systems also differ widely with respect to the above mentioned issues, i.e., scripting approach and style of computation. However, an analysis of our previous work suggested that it is indeed possible to develop a single platform which (a) can be used to construct a broad range of character applications, (b) even enables us to switch on-the-fly between the two basic scripting approaches, and (c) supports a clear separation between the specification of scripting knowledge (being a knowledge-engineering task), and the required computational machinery for behaviour generation (being an implementation task). In this paper, we first revisit some of our past character systems, discuss briefly their functionality as well as the underlying scripting approach and modularisation. We then introduce the MIAU platform and illustrate by means of illustrated generation examples how MIAU can be used for the realisation of applications that deploy characters in different conversational settings.

2. EARLIER SYSTEMS THAT EMPLOY LIFE-LIKE PRESENTATION CHARACTERS

There is a popular class of character systems in which a single character presents information to a user in the style

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in:

IUI'03, January 12–15, 2003, Miami, Florida, USA.
Copyright 2003 ACM 1-58113-586-6/03/0001

of a TV-presenter. An example of such a system is PPP (see Fig. 1) that employs a life-like character to explain technical devices and maintenance procedures [16].

From a functional point of view (see right-hand side of Fig. 1), the PPP Persona receives as input presentation goals (PG) and generation parameters (GP), such as the user's level of expertise and time constraints for the duration of the presentation to be generated. As output, the system delivers a specification of a multimedia presentation, called presentation *script* that will be forwarded to a dedicated player engine responsible for the synchronised display of all involved media objects including possibly animated illustrations of domain objects, text elements, as well as character animation and verbal speech output.

In PPP, we formalized action sequences for composing multimedia material and designing scripts for presenting this material to the user as operators of a planning system. Starting from a complex presentation goal, (e.g. to describe a technical device) the planner recursively performs a hierarchical goal decomposition until all subgoals have been expanded to elementary production, retrieval or presentation tasks (for details see [1]). The operators of the PPP system represent tactical knowledge about how to achieve presentation goals by means of a multimedia presentation including a character as an additional presentation device. Therefore, the operators are formulated from the point of view of a director who orchestrates the interplay of the character with the display of all other media objects. As to modularization of the planning mechanism, PPP uses just one planner for script generation.

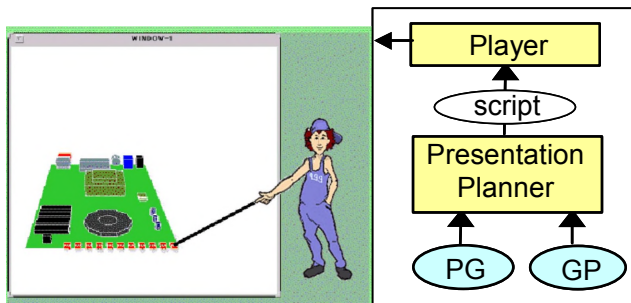


Fig.1. PPP Persona: Screenshot and architecture

A second class of systems we would like to include in our analysis promotes a conversational character as guiding interface metaphor. The stock agent “Rudi” (see Fig. 2) can be considered as a representative of this class.

Being connected to several online stock servers, a user can chat with Rudi about the latest developments of shares. As shown in the sketched architecture, Rudi's internal machinery is quite similar to that of the PPP Persona. However, the system also comprises a component for shallow NL parsing. The component tries to derive requests for new presentation goals and new settings of presentation parameters from a user's input message.

Next, we pick the so-called “eShowroom”, a virtual market place inhabited by life-like characters. In this system, a team of characters conveys information to the user by performing role plays (cf. Fig. 3).

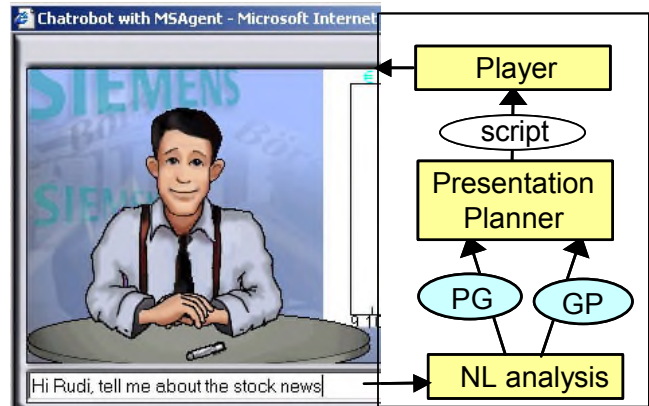


Fig.2. StockAgent: Screenshot and architecture

Similar as in PPP, a user specifies both a presentation goal and generation parameters prior to the presentation generation process. Since this time the behaviors of several characters have to be determined, a designer of such a system has the choice between taking a director's point of view or to adopt the self-scripting paradigm. In the first case, the task of the presentation planner is to work out one single script for the role play as a whole. Typically, such a script would include statements of the form: “character1 do x; character2 do y”. In contrast, when following the self-scripting approach, each character would have its own planner working out a script that specifies the character's individual contributions to the role play.

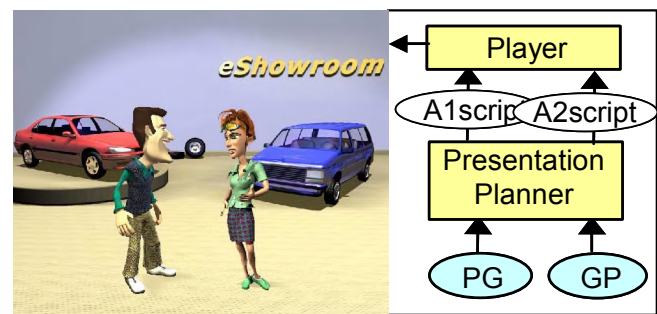


Fig.3. eShowroom: Screenshot and architecture

In our search for new presentation styles and further application fields, we allowed the user to participate in a role play (which in our case is mainly a multi-character conversation) while it is being performed. Rather than playing episodes to be watched by a user, such a scenario bears a lot of similarities to improvisational theatre. First of all, there is no pre-defined script. Instead the dialogue between the user and the characters evolves while time progresses. Furthermore, the scenario is open-ended.

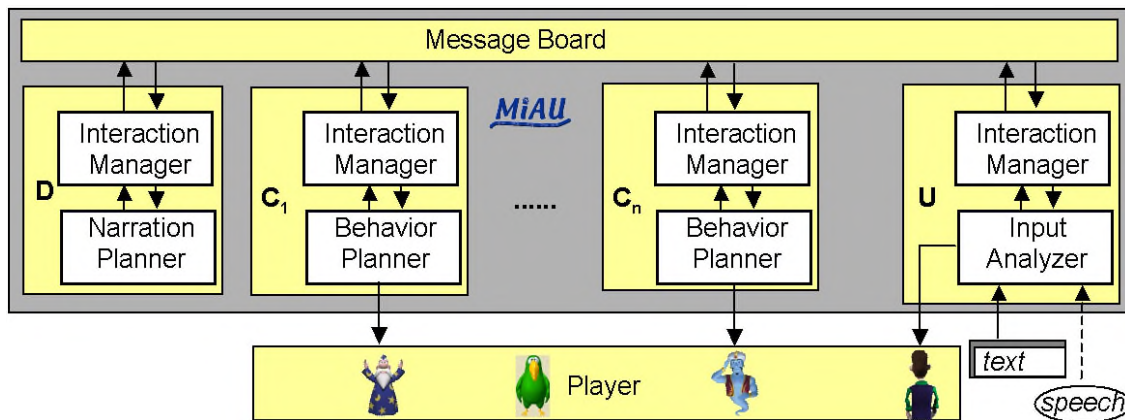


Fig. 4. Architecture of the MIAU Platform

Neither the characters nor the users are able to tell what exactly may happen next.

When starting work towards such a scenario, we first decided to equip each of the involved characters with a separate behavior planner. The first design of this platform has been sketched in [3]. However, this early architectural approach towards an interactive multi-character system relied on a number of hardwired synchronization mechanisms to orchestrate the overall behavior of the characters. The MIAU platform that will be described in the next section overcomes these limitations. As a result, it can not only be used to build systems that convey information to the user in the style of an interactive performance, but also allows for the realization of all system types mentioned above.

3. OUTLINE OF THE MIAU PLATFORM

The architecture of the MIAU platform is shown in the upper part of Fig. 4. We adopt the metaphorical distinction between a character's brain and a character's body which is typically reflected in an architecture by a separation of components for behavior planning on the one hand, and a character player component on the other hand. We further assume that the player will receive commands for direct execution from the superordinate behavior determining part. The MIAU platform itself abstracts from the player technology used for character animation, speech synthesis, and receiving user input. Rather, the platform consists of the following components:

For each character $C_1 \dots C_n$ MIAU foresees so-called *character components* containing a separate behavior planner as well as a separate interaction manager. The behavior planner has the task to decompose complex discourse goals into basic acts that can be executed by the character player. The interaction manager, in a certain sense, corresponds to a dialogue manager as found in NL dialogue systems since it is responsible for keeping book on interaction states and the interaction history. However, in MIAU the interaction manager realizes a character's internal interface for communication with other system components by means of formalized communication acts.

To allow a user to alter settings for the performance, to take an active part in a performance, or even to intervene in the role of a director or co-director, the platform also incorporates a U box, the so-called *user component*.

However, since this time the user decides on what to do, we don't foresee a planner, but an input analyzer for mapping user input onto formalized communication acts. The internal communication with other system components is handled by the interaction manager similar to the structure of a character component. In case the user is represented in the scenario by an embodied (and possibly animated) avatar, the avatar may be employed to audio-visually indicate his or her input activity. For instance, if a text widget is used for acquiring user input, the user's avatar may speak the input sentence. Currently, we restrict ourselves to a single participating user only. Nevertheless it seems possible to extend the architecture for multi-user, multi-character scenarios by adding more user components.

In addition to character components and the user component, MIAU incorporates also a D-box, the so-called *director component*. In contrast to the characters, the director does not participate in performances and therefore has no embodiment. Rather, this component is foreseen to enable some degree of centralized control on the overall interactive performance. While the director also comprises a planner, this time the planner is used in order to influence the course of the performance depending on the degree of centralized control wanted for a certain application. Internal communication with other components is again handled via the interaction manager.

Finally, the MIAU platform comprises a message board which is shared among the different components for the exchange of internal communication acts.

4. REALIZING DIFFERENT MODES OF OPERATION

The flexibility of the MIAU platform lies in the fact that different modes of operation can be achieved by choosing a certain task division between the character components on the one hand, and the role of the director component on the other hand. Specifying such a task division in MIAU does

not require changes in the implementation of the platform, but is a pure knowledge engineering task that manifests itself in writing plan operators for the components. In the sequel, we first introduce a rough classification of operators and then discuss some *modi operandi* in more detail.

Building upon our earlier work on automated script generation [1], an operator represents a mapping rule that takes as input an abstract and usually complex communicative goal and either: (a) maps it onto a set of less complex communicative goals (goal decomposition), or (b) specializes the goal by mapping it onto a more specific goal (goal specialization), or (c) maps the goal directly onto a command or a command sequence which can be passed to the character player component for execution (goal realization).

To allow for some character-specific variations when mapping basic communicative acts onto player commands, we introduce the operator type *instruction*. Unlike player commands, instructions still leave some room for individually rendering communicative acts. For instance, the instruction to perform a greet act could result in a wave gesture or a greet speech act, but also in a bow gesture in case of a Japanese character application.

In contrast to instructions, abstract communicative acts require deliberative planning and decision making to decompose and specialize the goal into subgoals until eventually a set of player commands has been worked out. For details on this hierarchical planning process we refer to [1]. To account for presentations given by a team of agents, we also introduced *dialogue acts*, such as *responding to a question* or *making a turn*, which refer to the interaction between the individual agents [2]. In the MIAU platform we make use of a specific kind of communicative act - the so-called *dialogue control acts* that essentially serve to make sure that a conversation develops in a certain direction.

Mode M1: Director Scripts all Characters in Detail

In the so-called "director mode" (or M1 for later reference), the role of MIAU's director component can be compared to that of a screen writer who generates a script for the single actors of a play. That is the characters only follow instructions while all deliberate planning processes are left to the director.

Technically speaking, the director posts a number of complex communicative goals, decomposes them into more elementary goals and writes the resulting instructions onto the message board. All instructions are annotated with internal identifiers for the characters so that there is no need for further coordination among them. All characters read the message board, pick those instructions which are addressed to them, perform a character-specific mapping of the instructions onto player commands, and pass the commands to the player for execution.

Depending on the used player technology for character animation, the instructions may also include some timing

information. For instance, when using the MS Agent player technology [13], a simple time stamp suffices to compute the proper sequence of command execution in the player.

It deserves mentioning that this modus can also be applied to implement characters that interact with a user. The basic idea is not to script the whole presentation in advance, but to invoke the director's planner after each new user turn.

M2: Self-Scripting with Directed Turn Management by Director

In this modus, the characters will not receive simple instructions, but read complex communicative goals from the message board. They perform their own goal decomposition process, and can even post new goals onto the message board to be processed by the other characters, or the user. For instance, a character's dialogue strategy may involve to set up the goal that an answer to a specific question should be provided.

In principle, all characters may try to get a goal from the message board. Whether it will do so or not, depends on its motivation. In particular, we consider the character's competence, its status, its personality and interest profile. For instance, extrovert agents with a high status have a higher motivation to join a discussion than introvert agents with a low status. Since all characters may apply for goals, some coordination mechanism is necessary. In M2, the director takes over this task. It checks which goals have been posted and has the authority to pick goals and assign them to the single characters for further processing. This scenario bears some similarity to a school situation where a teacher gives the word to one of the students who signaled their willingness to make a contribution.

There are a number of criteria that the director may take into account for decision making, such as the dialogue state. For instance, if someone has asked a question, the director will make sure that the question will be answered by the next turn rather than allowing new questions being posed. Another criterion is to whom the last turn was addressed. If a goal on the message board is annotated with a specific character, this agent will get the right of speaking with a higher probability than any other character. Finally, the director gives special priority to those goals that correspond to user input events.

M3: Fully Self-Scripting Characters

In this mode, the characters are again responsible for the posting and complete realization of presentation goals. In contrast to the previous mode M2, however, we don't involve the director at all. Thus, there is no superior control mechanism that handles turn-taking or assigns goals or instructions to the characters. Rather the assignment (and the complete realization) of communicative goals has to be handled by the characters themselves. In addition to M2, the characters not only have to self-assess their motivation to take the turn, but also to negotiate with the other characters to actually obtain it. As a first step, we implemented a protocol for turn-taking management that

shares some similarities with a game. Assume that at a certain point in time, a goal G has been written on the message board, and a new turn needs to be assigned. Following the game-style turn-assignment protocol, each character C proceeds as follows:

(i) C assesses its own motivation for processing G . A value from the interval $[0 \dots 1]$ may be used to express this. When finished, C signals to the other characters that it has completed its self-assessment process.

(ii) As soon as all characters have completed step (i), C as well as all other characters reveal their self-assessment values on the message board.

(iii) The character with the highest value will win. That is, it is allowed to take the goal and process it. If there is no unique winner, the characters may initiate a second round, or they may reach a decision in a randomized manner.

However, if new input from the user is received before the winning character has completed processing the posted goal G , G may become obsolete or at least less relevant. In this case, the new input will cause an update of the message board and a new round of the turn-talking negotiation game needs to be initialized.

M4: Self-Scripting with Director's Interventions

While the self-scripting approach M3 is attractive with regards to the scalability of the number of employed characters, it is easier to ensure coherency of a multi-party conversation when switching to the director's mode M1. As already illustrated by mode M2, a compromise can be achieved by a task division between the director and the characters. In M2, the director was only responsible for handling turn-taking among the characters. It did neither post new goals nor process goals on its own. However, it is also possible to have the director intervene in the course of a conversation, e.g., to increase coherency or to bring in new topics. In such a scenario, the role of the director would be similar to the role of the Helper Agent proposed in [8] for a multi-user chat application. To realize this mode, we equip the director with a set of *dialogue control acts*. In general, dialogue control acts lead to new communicative goals for the single agents that give the conversation a new direction. Note that this is a kind of indirect intervention. The director decides that there is a need for intervention, but leaves the realization of the corresponding dialogue acts to the single agents.

5. MIAU APPLICATION: CAR TALK

To demonstrate the flexibility of the MIAU platform, let's turn to the Inhabited Market Place and more specifically to an electronic car showroom. In this application domain, a single character or a team of characters are deployed to inform a user about the features of a certain car. The planners in this application are realized as BDI-clients in the JAM framework [7], and we use the MS Agent controller [13] for character animation.

5.1 Monologues

In the simplest case, the presentation will be given by a single character. For example, using the character profile interface in the left screenshot of Fig. 5, we may select the character Merlin to present a car from the point of view of a seller (i.e., for Merlin's role in the performance, we tick the option "seller"). In addition, we can specify Merlin's attitude towards the car, several personality traits, and Merlin's status relative to other characters (which is, however, not relevant in the case of performances with a single character).

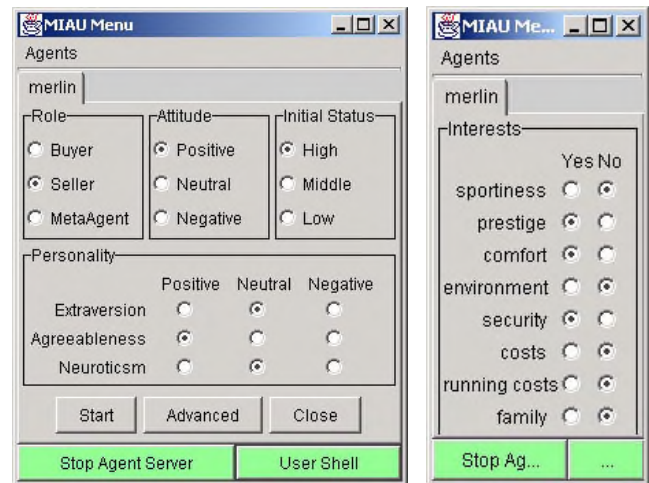


Fig. 5. Specifying a character profile in MIAU

The menu shown in the right screenshot of Fig. 5 allows us to specify Merlin's interest profile for the car domain. According to the chosen settings, Merlin pays particular attention to prestige, comfort, and security.



Fig. 6. Merlin giving a sales presentation

After setting Merlin's profile and pressing the "Start" button, the character Merlin shows up in the electronic show room. Since no other character has been "configured" so far, Merlin takes the initiative, welcomes the user and starts with a sales presentation (see Fig. 6).

The contents of Merlin's car presentation is determined by: (a) the set of plain facts about the car that are represented in the system's car database, (b) Merlin's interest profile for cars, (c) Merlin's attitude towards the car, and (d) the specified personality traits. For example, given Merlin's

interest in comfort and his positive attitude towards the particular car on display, he points out that the car has a number of features contributing to comfort. Linking the above mentioned factors together in a way so that a coherent monologue can be generated, however, requires an appropriate set of monologue generation operators. In the case of generating monologues, these operators can be part of the director's repertoire of operators, or likewise belong to Merlin's own set of self-scripting operators. While both approaches are possible, in the system session from which the current screenshots have been taken Merlin relies on the self-scripting approach.

5.2 Sales Talks Among Seller and Buyer Agents

Since monologues tend to be long-winded, let us add another character to the scenario, say Peedy in the role of a potential buyer. To involve Peedy, we first specify a character profile. To obtain an interesting car sales dialogue between Merlin and Peedy, we choose controversial settings this time. That is we indicate that Peedy has a negative attitude towards the car on display, and compared to Merlin, a complementary interest profile with regards to cars in general (see Fig. 7).

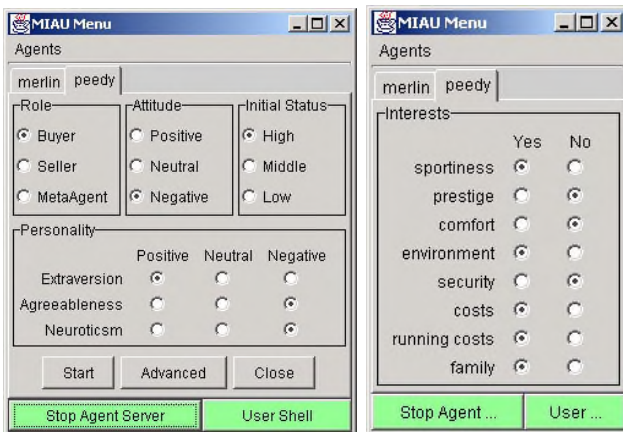


Fig. 7. Specifying Peedy's character profile

Right after pressing the "Start" button for Peedy, the character shows up. Realizing the new situation, Merlin, still in "monologue" mode, now switches to "dialogue" mode, greets Peedy, and asks what he can do for it. Playing the role of an interested customer, Peedy starts asking questions in order to learn more about those features of the car that relate to its interest profile. For instance, since Peedy is interested in the dimension "environmental friendliness", it wants to know whether the car has a catalytic converter (cf. Fig. 8). According to the car database, this is not the case. Having a general negative bias towards the car, Peedy expresses its disappointment explicitly (cf. Fig. 9). This type of conversation will continue unless (a) all car features have been mentioned that relate to the interest profiles of the involved characters, or (b) another change in the conversational setting occurs. Such a change can be initiated by the user (as in the case of activating Peedy), or likewise by the director, or any of the

performing characters. For instance, to add a variant to the question-answering scheme, the director may block Merlin to answer a question himself, but have him consult another seller character. Such a scenario is illustrated in Fig. 10, where Merlin calls his "colleague" Robby who is familiar with technical details, such as the car's fuel consumption.



Fig. 8. Buyer Peedy poses a question



Fig. 9. Buyer Peedy provides negative feedback



Fig. 10. Merlin calls his colleague Robby for advice

5.3 Allowing for User Participation

In the scenarios presented so far, the user's role was restricted to select characters for the performance and to set character profiles, but there was no user participation in the

performance itself. However, if a user wishes to engage in the car sales conversation, too, s/he can start an avatar that represents her/him in the scenario. As shown in Fig. 11, the user can now participate in the ongoing conversation by typing questions and comments into the text input widget.

However, it is up to the user to decide on the level of active participation. One extreme is that from now on the user asks all questions. However, if the user wants to relax again and watch the characters talk, s/he may remain silent. In this case, Peedy continues asking questions.



Fig. 11. A user avatar has joined the conversation.

5.4 Sending Characters Away

As it is possible to add further seller and buyer characters to the scenario, it is also possible to remove already performing characters. For instance, if the user would send all characters as well as her/his avatar away but Peedy, the conversational setting switches back to a monologue situation. Since Peedy plays the role of a buyer, however, it would start dreaming about its perfect car.

6. RELATED WORK

We discuss some representative examples of conversational embodied agents and classify them according to the four modes (M1-M4) of operation introduced before.

Most work so far concentrates on the realization of single presentation agents or dialogue agents, i.e. agents that correspond to mode M1. Prominent examples include the Internet Advisor Cosmo [10], the REA real estate agent [4], and the GRETA medical advisor [14], each relying on a sophisticated models of multimodal communication. Speech output, facial expression, and gestures generated by these systems start from the assumption that the agent is talking to a user who is facing the agent. Communication among virtual characters is not considered. In contrast, Agneta and Frida [6] is a character couple commenting on web pages visited by a user. The application corresponds to mode M1.

However, the system relies on pre-authored scripts. Therefore, the role of the director has been taken over completely by a human author.

Traum and Rickel [17] have addressed the issue of automatically generated multi-party dialogues in immersive virtual environments. In the context of a military mission rehearsal application, they address dialogue management comprising human-character and character-character dialogues. The characters are based on the Steve architecture which has been enhanced by a multi-modal dialogue model. There is no superior component that handles the turn management. To explicitly select the next speaker, the authors have introduced a specific turn-taking action called "Assign-Turn". In this system the characters are realized as completely autonomous agents, and the application essentially corresponds to mode M3.

Another multi-party application has been proposed by Isbister and colleagues [8]. Unlike the above-mentioned applications, their work concentrates on social interaction between several humans in a video chat environment which is supported by so-called Helper Agent. Helper Agent is an animated, dog-faced avatar that tracks audio from two-person conversations and intervenes if it detects longer silences. The task of the Helper Agent may be compared to that of our director in mode M4 with the difference that the director provides conversational support in more indirect manner. Instead of making utterances itself as the Helper Agent, it posts new conversational goals which are then realized by the participating agents.

Our work has also been inspired by research on interactive drama (see for instance [9,11]) that aims at integrating a user in a scenario – either as an audience member or an active participant. To allow for user interaction, systems usually incorporate decision points in a narrative-style script [12] or model their characters as autonomous agents that select and instantiate actions under consideration of dramatic constraints, such as the plot of a story or the characters' role and personality [5]. Similar to our own approach is Young's work [18] who introduces an execution manager that monitors the user's actions and makes real-time decisions about the appropriate system response in case these actions deviate from the planned narrative structure. The role of this execution manager essentially corresponds to that of our director in mode M4.

Besides building concrete agents, researchers focus on the development of application-independent technology for the implementation of such agents. One of the most prominent frameworks is the Collagen system [15] which is based on a model of collaboration between a human and a computer agent. An interesting feature of the framework is the fact that it allows for different degrees of system activity. For example, the default strategy in a help system is to have a task performed by the Collagen agent. This strategy would, however, be less appropriate in a tutoring application where it is important that the student learns how to perform a task. Unlike the Miao platform, Collagen focuses on two-party

conversation between a human and a computer agent even though an extension to multi-party applications seems obvious and feasible.

7. CONCLUSIONS

In this paper, we presented the MIAU platform that can be used for building a broad range of applications with life-like characters. We illustrated the flexibility of the platform by changing the conversational setting several times during an ongoing performance. Such a flexibility may pave the way for new engaging presentation formats. However, building an interesting character application that will be appreciated by their users is foremost a challenging design task. For instance, the number of characters, their roles as well as the foreseen role for the user need to be carefully chosen for a particular application. MIAU has proven useful to application builders since it can be used as a tool for rapid prototyping. At DFKI, we deploy the MIAU platform in a number of ongoing character projects including the EU funded projects Magicster and NECA. Furthermore, we use the platform for the implementation of different experimental settings to study the impact of different kinds of multi-agent dialogue on the user. For instance, in one of the settings we aim at a comparison of dialogues with specialists and dialogues with generalists. The outcome of this study will be reported in a future report.

Planned extensions to the MIAU platform comprise the development of more sophisticated interaction models, taking into account multimodal communication channels, and the participation of several users in a performance.

8. ACKNOWLEDGMENTS

This work has been funded partially by the BMBF (project MIAU), the EC (projects NECA and MagiCster) and the Federal State of Bavaria. Thanks to Peter Rist for designing the characters shown in Fig. 1-3.

9. REFERENCES

1. André, E., Rist, T., and Müller, J. 1999. Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence* 13:415–448.
2. André, E., Rist, T., van Mulken, S., Klesen, M., and Baldes, S. 2000. The Automated Design of Believable Dialogues for Animated Presentation Teams. In: Cassell, J., Sullivan, J., Prevost, S. and Churchill, E. (eds.): *Embodied Conversational Agents*, 220-255, Cambridge, MA: MIT Press.
3. André, E., Baldes, S., and Rist, T. 2002. From Simulated Dialogues to Interactive Performances. In: Marik, V., Stepankova, O., Krauwurmova, H., and Luck, M. (eds.): *Multi-Agent Systems and Applications II*, 107-118, N.Y.: Springer.
4. Cassell, J., Bickmore, T., Camphell, L., Vilhjalmsson, H., and Yan, H. 2000. The human conversation as a system framework: Designing embodied conversational agents. In: Cassell, J., Sullivan, J., Prevost, S., and Churchill, E. (eds.): *Embodied Conversational Agents*, 29-63, Cambridge, MA: MIT Press.
5. Hayes-Roth, B., van Gent, R., and Huber, D. 1997. Acting in character. In: Trappl, R., and Petta, P. (eds.): *Creating personalities for synthetic actors*, 92–112. New York: Springer.
6. Höök, K., Persson, P., and Sjölander, M. 2000. Evaluating Users' Experience of a Character-enhanced Information Space. *AI Communications*, 13(3), 195-212.
7. Huber, M. 1999. JAM: A BDI-theoretic mobile agent architecture. Proc. of 3rd Conf. on *Autonomous Agents*, 236–243. N.Y.: ACM Press.
8. Isbister, K., Nakanishi, H., Ishida, T., and Nass, C. 2000. Helper Agent: Designing an Assistant for Human-Human Interaction in a Virtual Meeting Space. Proc. of CHI 2000, 57-64, N.Y.: ACM Press.
9. Klesen, M., Szatkowski, J., and Lehmann, N. (2001). A Dramatised Actant Model for Interactive Improvisational Plays. Proc. of the 3rd Intelligent Virtual Agents Workshop, 181-194, N.Y.: Springer.
10. Lester, J. Voerman, J.L. Towns, S.G., and Callaway, C.B. 1999. Deictic Believability: Coordinated Gesture, Locomotion, and Speech in Lifelike Pedagogical Agents. *Applied Artificial Intelligence* 13:383–414.
11. Machado, I., Paiva, A., and Prada, R. 2001. Is the Wolf Angry or ... Just Hungry? Proc. of *Autonomous Agents*, 370–376. N.Y.: ACM Press.
12. Mateas, M. 1997. An Oz-Centric Review of Interactive Drama and Believable Agents. Technical Report CMU-CS-97-156, School of CS, CMU, Pittsburgh, PA.
13. Microsoft Agent: Software Development Kit 1999. Microsoft Press, Redmond Washington.
14. Pelachaud, C., Carofiglio, V., De Carolis, B., de Rosis, F. and Poggi, I. 2002. Embodied Contextual Agent in Information Delivering Application. Proc. of AAMAS 02, 758-765, N.Y.: ACM Press.
15. Rich, C., Sidner, C.L. and Lesh, N. 2001. COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. In: *AI Magazine*, 22(4): 15-25.
16. Rist, T., André, E., Müller, J. 1997. Adding Animated Presentation Agents to the Interface. Proc. of IUI 97, 79-86, N.Y.: ACM Press.
17. Traum, D. and Rickel, J. 2002. Embodied Agents for Multi-Party Dialogue in Immersive Virtual Worlds. In: Proc. of AAMAS'02, 766-733, N.Y.: ACM Press.
18. Young, M. 2002. The Cooperative Contract in Interactive Entertainment, In: Dautenhahn, K., Bond, A.H., Canamero and Edmonds, B. (eds.): *Socially Intelligent Agents: Creating Relationships with Computers and Robots*, 229-234, Dordrecht: Kluwer.