Automated Generation of Non-Verbal Behavior for Virtual Embodied Characters

Werner Breitfuss
Department of Creative Informatics
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
+81-3-5841-6774
werner@mi.ci.i.u-tokyo.ac.jp

Helmut Prendinger
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, Japan
+81-3-4212-2650
helmut@ nii.ac.jp

Mitsuru Ishizuka
Department of Creative Informatics
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
+81-3-5841-6755
ishizuka@i.u-tokyo.ac.jp

ABSTRACT

In this paper we introduce a system that automatically adds different types of non-verbal behavior to a given dialogue script between two virtual embodied agents. It allows us to transform a dialogue in text format into an agent behavior script enriched by eye gaze and conversational gesture behavior. The agents' gaze behavior is informed by theories of human face-to-face gaze behavior. Gestures are generated based on the analysis of linguistic and contextual information of the input text. The resulting annotated dialogue script is then transformed into the Multimodal Presentation Markup Language for 3D agents (MPML3D), which controls the multi-modal behavior of animated life-like agents, including facial and body animation and synthetic speech. Using our system makes it very easy to add appropriate non-verbal behavior to a given dialogue text, a task that would otherwise be very cumbersome and time consuming.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – Interaction styles, Screen design, Theory and methods

General Terms

Human Factors

Keywords

Multimodal input and output interfaces, multi-modal presentation, animation agent systems, processing of language and action patterns

1. INTRODUCTION

Combining synthetic speech and human-like conversational behavior like gaze and gestures for virtual characters is a challenging and tedious task for human animators. As virtual characters are used in more and more applications, such as computer games, online chats or virtual worlds like Second Life, the need for automatic behavior generation becomes more pressing. Thus, there have been some attempts to generate non-verbal behavior for embodied agents automatically. Systems like the Behavior Expression Animation Toolkit (BEAT) allow one to generate a behavior script for agents by just inputting text [1]. The drawback of most current systems and tools, however, is that they consider only one agent, or only suggest behaviors such that the animator still has to select appropriate ones.

The aim of our work is to generate all non-verbal behavior automatically for conversing agents, so that someone writing a script to be performed by two agents can focus on creating the textual dialogue script and just feed it into the system. A salient feature of our system is that we generate the behavior not only for the speaker agent but also for the listener agent that might use backchannel behavior in response to the speaker agent. Employing two presenter agents holding a dialogue is advantageous, since watching (or interacting with) a single agent can easily become boring and it also puts "stress" on users, as they are the only audience. Furthermore, two agents support richer types of interactions and "social relationships" between the interlocutors. Also TV- commercials, games, or news use two presenters, because of the increased interaction possibilities and entertainment value. In this paper, however, we confine discussion to the case where one user just watches the performance (dialogue) of two virtual agents, and does not interact with them.

The paper is structured as follows. In Section 2 we discuss related work. Section 3 describes our system and the way gaze behavior and other non-verbal behavior is generated by means of a "walk through" example. Section 4 concludes the paper, and briefly discusses envisioned future developments of our system.

2. RELATED RESEARCH

Existing character agent systems already support the automated generation of some behaviors, such as automatic lip-synchronization. The next step is to automatically generate agents' conversational behavior from text. In this section, we report on some previous attempts, which combine various disciplines like computer animation, psychology, and linguistics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMI'07, November 12–15, 2007, Nagoya, Aichi, Japan.
Copyright 2007 ACM 978-1-59593-817-6/07/0011...$5.00.

The BEAT system [1] generates synthesized speech and synchronized non-verbal behavior for a single animated agent. It uses plain text as input, which is then transformed into animated behavior. First, text is annotated with contextual and linguistic information, based on which different (possibly conflicting) gestures are suggested. Next, the suggested behaviors are processed in a 'filtering' module that eliminates gestures that are incompatible. In the final step, a set of animations is produced that can be executed, after necessary adoptions, by an animation system. The BEAT system can handle any kind of text and generate a run-able agent script automatically. The system uses a generic knowledgebase where information about certain objects and actions is stored, and the selected gestures are specified in a compositional notation defining arm trajectories and hand shapes independently, which allows the animator to add new gestures easily, or adjust existing ones.

A different approach is suggested in [5]. This system supports the author in writing agent scripts by automatically generating gestures based on predefined rules, and using machine learning to create more rules from the set of predefined rules. It was used in the COHIBIT system, where the author first has to provide a script containing the actions for two virtual characters. In the next step the author writes simple gesture rules using his or her expert knowledge. Using this corpus of annotated actions the system can learn new rules. In the third step the system suggests the most appropriate gestures to the author, which are, after resolving conflicts and filtering, added to the already existing ones. Finally it produces a script with the gestural behavior of both virtual characters. Similar to our work, two agents are used, but since we want to reduce the workload to the minimum, our system does not require any input from the author except the dialogue to be presented by our characters.

[3] investigated the many functions of gaze in conversation and its importance for the design of believable virtual characters. The gaze behavior of our agents is informed by empirically founded gaze models [4,8,12]. [4] analyzed gaze behavior based on two-person dialogs and found that gaze is used to regulate the exchange between the speaker and listener. In that work, different gaze patterns like the q-gaze (the speaker is looking at the person he/she is interacting with), and a-gaze (p is not looking at the interlocutor) were defined. It was found that the speaker looks at the listener while speaking fluently, but looks away when starting to speak or during hesitation (influent speech). In this way, speakers can keep the listeners attention or, by looking away, gain time to think about what to say next. Another finding is that mutual gaze can regulate the level of emotionality between interlocutors. The experiment described in [12] evaluates gaze behavior in multiparty environments, where four-person groups discussed current-affair topics in face-to-face meetings. Their results show that on average, interlocutors look about seven times more often at the speaker they listen to, than at others, and speakers looked about three times more at the addressed listener than at non-addressed listeners. Furthermore, the total amount of time spent gazing at each individual in a group of three is nearly 1.5 times higher than if visual attention of the speaking person were divided by three. These results are very relevant for our gaze algorithm since they give us the basis for a 'two agents' situation. And they also provide the needed information for our gaze generation rules. The work in [8] developed a model of attention and interest based on gaze behavior. An embodied conversational agent may start, maintain, and end a conversation dependent on its perception of the interests of the other agents.

Other related research was done is [2], which introduces a behavior synthesis technique for conversational agents in order to generate expressive gestures, including a method to individualize the variability of movements using different dimensions of expression. The work described in [6] presents a gesture animation system that uses results from neurophysiologic research and generates iconic gestures from object descriptions.

3. BEHAVIOR GENERATION SYSTEM

Our system consists of three different modules:

- Language Tagging module,
- Non-Verbal Behavior Generation module,
- Transformation to MPML3D module.

The Multimodal Presentation Markup Language is used to control the behavior of our 3D agents [7]. We choose a modular pipelined architecture to support future extensions. The code of the system is written in Java, and the XML format is used to represent and exchange data between modules.

The Language Tagging module takes the input dialogue text and uses the language module from the BEAT toolkit [1] to annotate linguistic and contextual information. Next, the Behavior Generation module adds non-verbal behavior like eye gaze and gestures to the annotated input sentence. In the final step, an MPML3D file is produced. The MPML3D player displays the embodied characters agents.

In our system, gaze patterns are generated for two different types of roles: (1) the speaker, i.e. the agent that is speaking and addresses the other agent, and (2) the listening agent. We can currently generate gaze behavior and gestures for these two roles, based on a fixed set of rules.

Gaze directions have certain probabilities of occurrence, which we derived from existing gaze models [4,8,12]. In order to avoid conflicts between certain gaze behaviors, like looking in two different directions at the same time, we assigned priorities to them. Typically, more specific gaze behaviors (such as looking at speaker/listener) have higher priority than e.g. looking around. Moreover, we prioritize gazes that occur before starting the utterance, i.e., speakers typically look away before starting a long utterance (in order to concentrate on planning their dialogue contribution).

The rule in Figure 1 (adapted from [1]) shows one example of how the gaze behavior for the speaker is generated.

```
FOR each THEMA node in the tree
  IF at the beginning of the utterance
  Or 70% of the time
      Look away from listener
FOR each RHEMA node in the tree
  IF at the end of the utterance
  Or 73% of the time
      Look at listener
```

Figure 1. Gaze generation for the speaker

In addition to generating the gaze behavior for the speaking agent we also have to consider the agent in the role of the listener. Since listeners typically look at speakers when they start an utterance (after taking floor) to demonstrate their attentiveness, we developed rule like the one if Figure 2.

```
FOR each THEMA node in the tree
  IF at the beginning of the utterance
  Or 80% of the time
      Look at speaker
FOR each RHEMA node in the tree
  IF at the end of the utterance
  Or 47% of the time
      Look at the speaker
```

Figure 2. Gaze generation for the listener

The Gestures of our agents are generated in similar manner, broadly following rules proposed in [1].

Let us now walk through one simple example utterance and see how our system works. As input we take the sentence: “This is just a small gaze example.” At first we send it to the Language Tagger module, which annotates the sentence with linguistic and contextual tags. The output of this process is shown in Figure 3. Here, “NEW” means that the word has not yet occurred in the conversation, and is thus a candidate for being accompanied by a “beat” gesture.

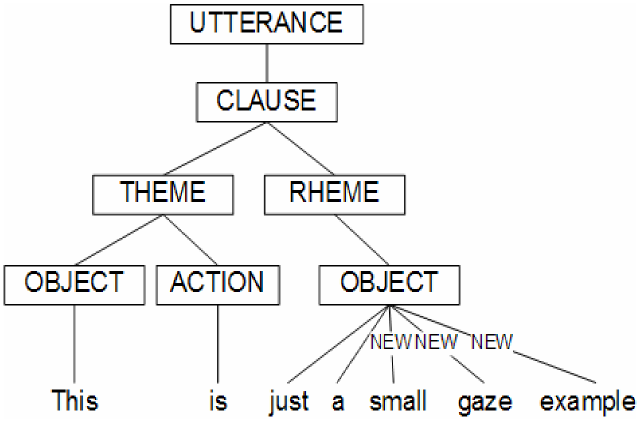

Figure 3. The output tree of the language module

In the next step, we pass this newly constructed tree to our Behavior Generation module. It first generates a new tree with the gaze behavior and gestures (and speech parameters) for the speaker and a second tree for the listener. The tree for the listening character has the same structure as the speakers’, but contains the nodes for the non-verbal behavior that should be displayed by the listener agent.

Figure 4 shows the speaker’s tree, which was generated by our system for the sentence used in this short example. As we can see, the root node of the tree is the utterance, and there is a speech pause between the theme and rheme of the sentence (see [1] for a discussion of speech parameters). The gaze behavior “Gaze away” and “Gaze at listener” is derived from the previously discussed rule (Figure 1). The gesture behavior is generated according to dedicated gesture generation rules of the Behavior Generation module. In our example, a beat gesture is selected to accompany the word “just”, and an iconic gesture (for describing something small) is suggested to co-occur with the phrase “small gaze example”.

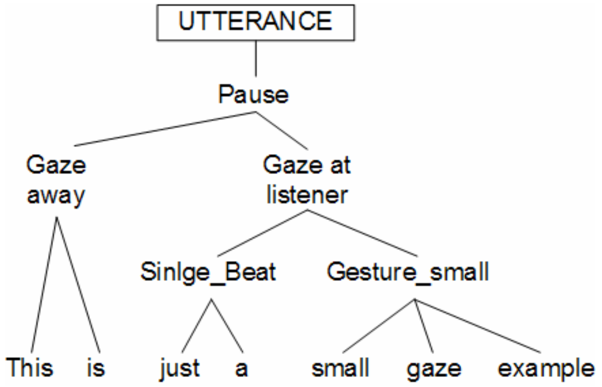

Figure 4. Tree for the speaker behavior

The behavior tree of the listener agent is generated similarly to that of the speaker agent (see Figure 5). It is based on the same tree that is output from the Language Tagging module of the speaker agent, but applies listener behavior generation rules instead of speaker rules. Again, we start with root node “UTTERANCE”. During the speaker’s speech pause, there is no behavior for the listener agent is defined. The listener’s gaze behavior is added according to the rule in Figure 2, i.e. the listener is looking at the speaker when the utterance begins. Accordingly, our system creates the label “Gaze at speaker”. Since the listener agent is paying attention to the user, it continuous to look at the speaker also in the “rheme part” of the utterance.

Thereafter, appropriate gestures are suggested for the listener agent. Whereas no gesture is suggested for the phrase “just a”, the phrase “small gaze example” is accompanied by head nods. In our system, a head nod is a basic gesture type for the listener. It is the gesture with the lowest priority and is used when no other, more specific gesture can be suggested. In the future, a dedicated “backchannel” knowledge base will be created to insert listener head nods in an informed, systematic manner.

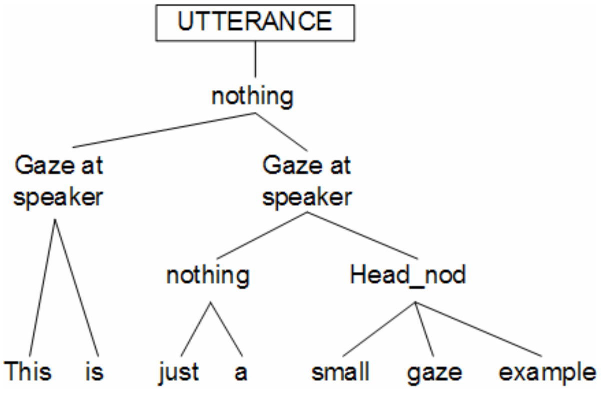

Figure 5. Tree for the listener behavior

After speaker and listener behavior trees are created, they are passed to the Transformation module, which compiles them into a synchronized MPML3D XML file. Figure 6 shows our agents performing the example sentence.

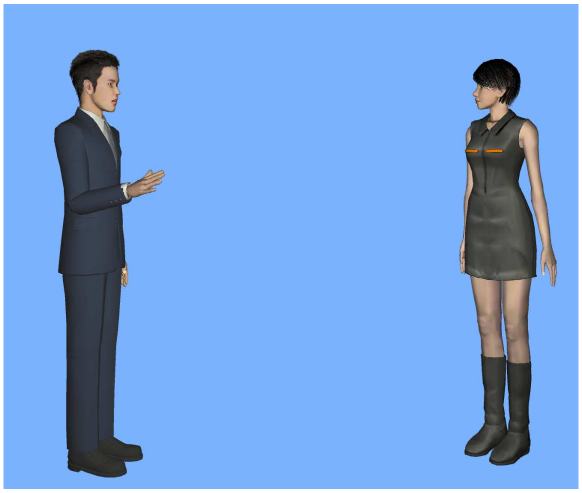

Figure 6. MPML3D Agents enacting the example

4. CONCLUSIONS

There is ample evidence that agent-based multimodal presentations can entertain and engage the user, and are also an effective way to mediate information [10]. In this paper, we described our system that automatically generates gaze and gestures for two agents, in the roles of speaker and listener. It uses a dialogue script as its only input (from the content creator), and transforms it into a run-able multimodal presentation using two highly realistic 3D character agents.

In our future work, we plan to analyze the emotional content of text based on the work described in [11], and add emotional expressions to the agents' behavior in order to improve the naturalness of the performed dialogue. The emotion expressed in a sentence will also affect voice parameters, gaze, and gesture behavior. Conversational behavior is also influenced by the social role (instructor-student, employer-employee, etc.), the cultural background, and the personality of the interlocutors. Another venue of research relates to including a model of the user as a listener, who might be addressed by the agents.

Our next step, however, will address easier issues. Besides extending the set of behavior generation rules for the listener agent, we want to align the behavior of the agents with respect to a slide show and virtual objects in a 3D environment. Here, we have to analyze phrases like "if you look at the slide" and generate appropriate behavior for the speaker and listener agent. Among others, the selected gaze behavior has to be timed and directed to specific locations in the 3D environment. In this way, "joint attention" (gaze) behavior will be implemented.

For all of our ideas, the focus will remain on the exploration of ideas that ultimately lead to a minimal workload for content creators, while ensuring high-quality, professional output in the form of natural and enjoyable multimodal presentations.

5. REFERENCES

[1] Cassell, J., Vilhjálmsson, H., and Bickmore, T.: BEAT: the Behavior Expression Animation Toolkit. In: Proceedings of SIGGRAPH 2001, pages 477-486 (2001)

[2] Hartmann, B., Mancini, M., Buisine, S., and Pelachaud, C.: Design and evaluation of expressive gesture synthesis for embodied conversational agents. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. ACM Press (2005)

[3] Heylen, D.: Head gestures, gaze and the principles of conversational structure, In: International Journal of Humanoid Robotics Vol. 3 Nr. 3, pages 241-26 (2006)

[4] Kendon A.: Some functions of gaze-direction in social interaction. In Acta Psychologica 26, pages 22-63, North-Holland Publishing Co. (1967)

[5] Kipp M.: Creativity meets automation: Combining nonverbal action authoring with rules and machine learning, In: Proceedings of the 6th International Conference on Intelligent Virtual Agents (IVA 2006), Springer, pages 230-242 (2006)

[6] Kopp S., Tepper, P., and Cassell, J.: Towards integrated microplanning of language and iconic gesture for multimodal output. In: Proceedings International Conference on Multimodal Interfaces 2004, ACM Press, pages 97-104 (2004)

[7] Nischt M., Prendinger, H., André, E., and Ishizuka, M.: MPML3D: a reactive framework for the Multimodal Presentation Markup Language. In: Proceedings 6th International Conference on Intelligent Virtual Agents, Springer, pages 218-229 (2006)

[8] Peters C., Pelachaud C., Bevacqua E., and Mancini M.: A model of attention and interest using gaze behavior. In: Proceedings of 5th International Conference on Intelligent Virtual Agents 2005, pages 229-240 (2005)

[9] Prendinger, H. and Ishizuka, M., editors. Life-Like Characters. Tools, Affective Functions, and Applications. Cognitive Technologies. Springer, Berlin Heidelberg, (2004)

[10] Rist T., André, E., Baldes, S., Gebhard, P., Klesen, M., Kipp M., Rist, P., and Schmitt, M.: A review of the development of embodied presentation agents and their application fields. In: Prendinger and Ishizuka [9], pages 377-404

[11] Shaikh M., Prendinger H. and Ishizuka M.: A Cognitively Based Approach to Affect Sensing from Text. In: Proceedings 10th International Conference on Intelligent User Interfaces, ACM Press, pages 349-351 (2006)

[12] Vertegaal, R., Weevers, I., Sohn, C., and Cheung, C.: Gaze-2: conveying eye contact in group video conferencing using eye-controlled camera direction. In: Proceedings of the SIGCHI Conference on Human factors in Computing Systems (CHI'03), pages 521–528, ACM Press (2003)