SPECIFICATION AND REALISATION OF MULTIMODAL OUTPUT IN DIALOGUE SYSTEMS

Jonas Beskow, Jens Edlund and Magnus Nordstrand

CTT (Centre for Speech Technology), KTH, Sweden
Drottning Kristinas väg 31, SE-100 44 STOCKHOLM
{beskow, edlund, magnusn}@speech.kth.se

ABSTRACT

We present a high level formalism for specifying verbal and non-verbal output from a multimodal dialogue system. The output specification is XML-based and provides information about communicative functions of the output without detailing the realisation of these functions. The specification can be used to control an animated character that uses speech and gestures. We give examples from an implementation in a multimodal spoken dialogue system, and describe how facial gestures are implemented in a 3D-animated talking agent within this system.

1. INTRODUCTION

Spoken dialogue systems incorporating some form of animated characters are becoming increasingly popular. There are many compelling reasons to include an animated agent in the interface. Since people have life-long experience at interpreting facial expressions and gestures, it is one of the most intuitive and non-intrusive interfaces imaginable. Using gestures, an agent can continuously provide the user with implicit feedback about the progress of the dialogue. This is an elegant way to alleviate problems with turn taking, resulting in a smoother dialogue flow, while at the same time making the system appear more responsive. The agent can help to direct the user's attention to specific areas on the screen using deictic gestures. Given proper speech-synchronised articulatory movements and emphatic gestures, the agent will boost the intelligibility of the spoken output [1].

Implementing gestures for an animated agent is however a time consuming task and the result may not always be portable or re-usable. Furthermore, there is a potential conflict between creating clear and unambiguous gestures and avoiding stereotypic or repetitive behaviour, which would make the agent less natural and lifelike.

In this paper we suggest a high-level abstraction layer for specifying verbal and non-verbal output in a way that frees the dialogue manager from having to know any details about the capabilities of the animated agent, which has two implications. Firstly it makes the dialogue system portable across different output channels - it is not tied to a specific animated agent with a given set of capabilities - any output module that can signal the communicative functions in a meaningful way will do. Secondly, once gestures for an animated agent have been implemented in a way that conforms to the specification, the gestures can be reused in other dialogue systems or domains.

2. RELATED WORK

Several models for automatic generation of gestures for animated characters in conversational systems have been proposed. [2] present static facial displays for signalling communicative functions in a dialogue system. [3] present a model for generating facial expressions and intonation from a common representation. [4] present an agent capable of signalling it's communicative goal for example by showing emotions in the face. [5] and [6] both describe complete frameworks for conversational dialogue systems incorporating animated agents capable of generating deictic gestures, turn- taking signals and emblematic gestures, relying on input from several sources. In contrast, our more limited model aims at separating the dialogue system from the realisation of output in order to facilitate rapid development and portability. The work in this paper builds on our experiences from previous attempts at integrating animated characters into conversational dialogue systems developed at CTT ([7], [8], [9] and [10]).

3. IMPLEMENTATION

The work described in this paper has been implemented in a multimodal spoken dialogue system, AdApt [11]. The AdApt system was developed at CTT with Telia Research as an industrial partner. It allows users to browse the real-estate market in downtown Stockholm, and features multimodal input and output. The input takes the form of speech and pointing/clicking on a map; output consists of lip-synchronised synthetic speech and facial gestures produced by an animated talking head, as well as interactive map displays. The AdApt system has a modular architecture (see Figure 1) that makes it a good test bed for exploring different aspects of multimodal input and output and performing user studies.

4. ANIMATED AGENT

The animated agent is based on a 3D parameterised talking head that can be controlled by a TTS system to provide accurate lip-synchronised audio-visual synthetic speech [12]. The facial model includes control parameters for articulatory gestures as well as facial expressions. Parameters in the former category include jaw opening, lip closure, labiodental occlusion, tongue tip elevation, lip rounding and lip protrusion while the latter category includes controls for raising and shaping of eyebrows, smile, eyelid opening, gaze and head movement. Gestures can be developed using an interactive parameter editor based on the WaveSurfer platform [13].

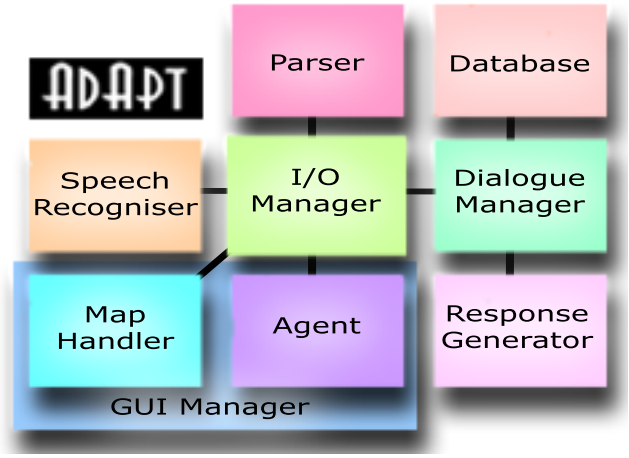

Fig. 1. The AdApt system architecture

5. GESTURE SPECIFICATION

Most dialogue systems, and indeed most interactive systems in general, are based around an event-driven model, i.e. actions that are carried out by the system are triggered by some kind of event. The events occurring in a dialogue system can either be the direct result of a user action (such as speaking) or they could be internally generated during the system's data processing. A spoken dialogue system has, at the bare minimum, one user event, which we can call "speech done" - the user has finished speaking and the utterance is available to the system. The system will process the utterance and respond in some way, after which it will wait for the next "speech done" event. This one-event-per-turn model is sufficient for a simple system, but we would want an animated agent to give as much feedback as possible during all the different stages in a turn, not just during the system's speech output.

A human taking part in a dialogue continuously uses input from many sources in different modalities to decide how to proceed: visual cues such as gaze, facial expression, hand- and body gestures, speech-related cues such as phrasing and intonation, and the actual spoken content. A spoken dialogue system goes through several stages of processing during a dialogue turn, i.e. speech recognition, parsing, etc. Between these, the system may provide feedback to the user, but during each processing stage, the system is more or less unable to do so. The following system events occur during a dialogue turn in the AdApt system:

1. *Start of speech* - the speech recogniser has detected that the user is speaking.
2. *End of speech* - the speech recogniser has detected that the user has stopped speaking.
3. *Recognition done* - the speech recogniser has processed the utterance and passed the result on to the parser.
4. *Semantics done* - the parser has processed the recogniser output. The parser will categorise an utterance as either closing (the utterance can be interpreted in its own right) or non-closing (more input is needed to make sense of the utterance). Closing utterances will be passed on to the dialogue manager. Non-closing utterances causes the system to go back to listening.
5. *Planning done* - the dialogue manager has decided what to do next.

All of these events represent points in time where feedback to the user could be given. Based on the capabilities and requirements of a dialogue system, we have divided non-verbal output into two basic categories. We call the first category *event gestures* and the other *state gestures*.

5.1. Event gestures

There are instants during a dialogue turn when the system will need to produce transient output. By transient we mean that the duration of the output is finite and can be predicted (as opposed to the states described below, which can be of arbitrary length). An event gesture is realised and then disappears. Emphasis on a particular word during speech output could be indicated using an appropriate gesture, such as a head nod, eyebrow raise or an eye widening. Agreement could be indicated using a head nod or a slow blink instead of speech. These *event gestures* leave the agent in the same position it was before the event took place.

5.2. State gestures

Some of the things one would want the agent to communicate are poorly modelled by transient gestures. If we want to signal that the agent is performing an action, e.g. Listening, searching a database ("thinking") or just being idle, we need the signals to be visible for an arbitrary amount of time. These kinds of behaviour are encoded as *states*. A state represents what the system is doing at a given moment, and has the following properties:

- The agent must always be in one and exactly one state at any given time
- A state lasts until another state is entered

In the AdApt system, states are used for feedback concerning the dialogue flow. The states *idle*, *listening*, *talking*, *busy*, and *continued attention* are used.

6. GESTURE LIBRARY

The events and states described above provide all the semantic information the animated agent needs in order to produce meaningful gestures, and the set of defined event and state gestures constitute all the information that needs to be encoded both on the agent side and in the dialogue system. How the gestures should actually look is up to the agent. In our implementation, this information is encoded in a gesture library.

6.1. Gesture realisations

At the lowest level of the library are descriptions of the actual gesture realisations. For most states and events there are multiple realisations with subtle differences. For our parametrically controlled animated agent, the descriptions are coded in terms of parameter tracks. For other types of agents, they would be coded in other ways, for example as 2D-animation sequences. Our gesture realisation descriptions include the time offset to the *stroke* of the gesture, i.e. the point in time where the centre of gravity of the gesture occurs. This information is used to synchronise the timing of gestures and other events such as stressed syllables of emphasised words.

	Dialogue system event	State	Event gesture
USER:	Start of speech	listening	
den röda lägenheten...	End of speech		
the red apartment...	Recognition done		has_heard
	Semantics done (fragment)	continued attention	
USER:	Start of speech		
...har den öppen spis?	End of speech		
...does it have a fireplace?	Recognition done		has_heard
	Semantics done (yn-question)	busy	
	Planning done		
SYSTEM:	Speech synthesis, background attribute set to positive	talking	emphasis
ja den **röda** lägenheten har öppen spis			
yes *the* ***red*** *apartment has a fireplace*	Synthesis done	listening	

Table 1. A turn in the AdApt system.

6.2. Structure of the library

The gesture library contains a separate entry for each event and state. For events, a set of alternative gesture realisations is defined. Gestures defined for a particular event will typically have similar semantic meaning - some gestures might only have subtle differences (e.g. in duration) whereas others may differ in style (such as a head nod vs. an eye widening gesture to signal emphasis). Each gesture is given a weight to make it more or less likely to occur. By allowing alternative realisations, the agent will seem less repetitive and more natural in its behaviour.

In order to deal with the arbitrary length of states, these are divided into three segments: *enter, sustain* and *exit*. For each of the segments, one out of a set of alternative gestures will be chosen, as with the events. Gestures in the *enter* and *exit* segments are performed once (on state entrance and exit respectively), while the *sustain* gestures are executed at random intervals during the duration of the state. *Enter*- and *exit* gestures are paired in such a way that if a particular enter gesture is picked, the corresponding exit gesture will be chosen. This makes it possible for the *exit* gesture to restore the parameters that have been changed by the *enter* gesture.

6.3. Choosing between multiple realisations

In the current implementation, selection of a particular gesture is done in a weighted random fashion, based on the weights specified for each entry in the library. Although the gestures in each group are supposed to be semantically equivalent, there might be external factors making a particular gesture inappropriate at some given point. When choosing a realisation for an emphasis event, an emphatic nod is not well suited if the utterance is of negative nature - an eyebrow raise or lowering would fit better. To deal with this, we allow background information to influence the weights of the gestures, making them more or less likely in a particular context. Background information is best described as global variables that are orthogonal to states and events. An example of a background variable is *responsetype*, which can be specified per utterance and is either *positive*, *negative* or *neutral*.

6.4. Gesture co-articulation

Up until now we have considered each gesture realisation as being independent of preceding and following gestures. This is however an oversimplification - just as with speech, there is co-articulation among gestures. If a head nod is followed by a look-right-gesture, it would be unnatural if the agent returned to the neutral pose (straight ahead, which is the ending pose of the nodding gesture) before starting to turn the head sideways to the right. The natural thing would be to go more or less directly from the low point of the nod to the looking-right pose. To achieve this behaviour we have implemented a co-articulation algorithm that merges gestures that are overlapping or adjacent in time. The algorithm will always preserve the area around the stroke of each gesture, but segments before and after this area are subject to reduction. Reduced parts of the track are interpolated with a smooth spline curve.

6.5. XML-based API

The output from the dialogue manager takes the form of an XML-formatted output description. The output will normally be realised as a combination of speech and gestures, but it may be only gestures. Output to the agent is marked by the tag `<response>`. If the response contains text data, it will be synthesised and spoken by the audio-visual TTS system. The response tag can be given an attribute named `background` that can be used to set background variables that affect the gesture selection process as described in section 6.3.

State changes and events can be inserted at any point in the response, denoted by the tags `<state>` and `<event>`. If a tag occurs in the middle of the text, it will take effect when the following word in the text is spoken by the TTS. More precisely, the stroke of the gesture associated with the tag will be synchronised with the first stressed vowel of the word, or the beginning of the word if it is unstressed. If there is no text specified in the response (i.e. a non-verbal response), any state changes or events will be executed immediately.

7. AN EXAMPLE

Table 1 shows a typical question-answer turn in the AdApt system. The user and system utterances are listed in the leftmost column. The parser interprets the first user utterance as fragment of an utterance [14], in this case a topicalisation followed by a pause. The “red” in the user utterance refers to a colour coded apartment icon on a map next to the animated agent on the display. When the system receives an incomplete utterance, it enters the state *continued*

attention and waits for more user input. The enter gesture of the *continued attention* state could be a head tilt or a slight head lowering while the agent still keeps its gaze at the user. When the user's second utterance is parsed, the system recognises the combination of the two utterances as a yes/no-question, and starts planning a response. This involves a database search and text generation based on the result of the search. It goes into the *busy* state to show that it is about to reply. The enter gesture of the *busy* state could be to move the gaze away from the user, or some other gesture designed to help the agent hold the floor. During the system's spoken response, the agent is placed in the *talking* state, which is a neutral state with a low blink frequency. When the system utterance is complete, the system goes back to the *listening* state to signal that it is once again ready for input. Sustain gestures, e.g. blinking, are not shown in the table, but are performed throughout the dialogue. Their frequency and exact realisation depends on the present state.

8. CONCLUSIONS AND FUTURE WORK

The system presented here is a first attempt at a generalised description formalism for multimodal output from a dialogue system. Our experiences from the implementation in the AdApt system indicate that it is successful in its pursuit, namely to form an abstraction layer between the dialogue manager and the output module, so that the dialogue manager does not need to know about the capabilities of the output module. The output module - for example the animated agent - is responsible for suitable realisation of the communicative functions requested by the dialogue manager.

Since the output description does not assume anything about the capabilities of the output device, it is fully possible to realise the output in some other way than through the gestures in an agent. An alternative might be to use familiar GUI metaphors, such as an hourglass for the *busy* state or a blinking red lamp for *listening* (recording) [15]. This would allow output generation on hardware incapable of rendering the animated agent, such as present day cell phones or PDAs.

Our current implementation of the animated agent uses a library of handcrafted gesture descriptions, grouped by communicative function. This is a very flexible model, since it allows us to model different attitudes, manners, personalities, moods or the socio-cultural identity of the agent simply by defining a new set of gesture descriptions (at least theoretically, assuming that the communicative functions are invariant). However, creating gesture realisations is a laborious process, and to convincingly model e.g. attitudes and emotions would require extensive studies of real-life subjects. A faster and more accurate way of obtaining the gesture realisations would be to record facial movement of an actor using a motion capture system such as [16]. Work towards this end is in progress at CTT.

9. ACKNOWLEDGEMENTS

This research was carried out at the Centre for Speech Technology, a competence center at KTH, supported by VINNOVA (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organisations.

10. REFERENCES

[1] E. Agelfors et al., "Synthetic Faces as a Lipreading Support," in *Proceedings of ICSLP*, Sydney, Australia, 1998.

[2] K. Nagao and A. Takeuchi, "Speech dialogue with facial displays: Multimodal human computer conversation," in *Proceedings of the 32nd ACL'94*, 1994, pp. 102–109.

[3] C. Pelachaud and S. Prevost, "Sight and sound: Generating facial expressions and spoken intonation from context," in *Proceedings of the 2nd ESCA/AAAI/IEEE Workshop on Speech Synthesis*, New Paltz, NY, Sept. 1994, pp. 216–219.

[4] I. Poggi and C. Pelachaud, "Performative Facial Expressions in Animated Faces," in *Embodied Conversational Agents*, J. Cassell et al., Ed., pp. 155–188. MIT Press, Cambridge, MA, 2000.

[5] K. R. Thórisson, "A Mind Model for Multimodal Communicative Creatures and Humanoids," *International Journal of Applied Artificial Intelligence*, vol. 13, no. 4-5, pp. 449–486, 1999.

[6] J. Cassell et al., "Human Conversation as a System Framework: Designing Embodied Conversational Agents," in *Embodied Conversational Agents*, J. Cassell et al., Ed., pp. 29–63. MIT Press, Cambridge, MA, 2000.

[7] J. Bertenstam et al., "The Waxholm system - a progress report," in *Proceedings of Spoken Dialogue Systems*, Vigsø, Denmark, 1995.

[8] J. Beskow, K. Elenius, and S. McGlashan, "Olga - a dialogue system with an animated talking agent," in *Proceedings of Eurospeech'97*, Rhodes, Greece, Sept. 1997.

[9] J. Gustafson, N. Lindberg, and M. Lundeberg, "The August dialogue system," in *Proceedings of Eurospeech'99*, Budapest, Hungary, Sept. 1999.

[10] B. Granström, D. House, and M. Swerts, "Multimodal feedback cues in human-machine interactions," *Prosody 2002*, Aix-en Provence, France. under review, 2002.

[11] J. Gustafson et al., "Adapt - a Multimodal Conversational Dialogue System in an Apartment Domain," in *Proceedings of ICSLP 2000*, Beijing, China, Oct. 16-20 2000, pp. 134–137.

[12] J. Beskow, "Animation of Talking Agents," in *Proceedings of AVSP'97*, Rhodes, Greece, 1997, pp. 149–152.

[13] J. Beskow and K. Sjölander, "Wavesurfer - an Open Source Speech Tool," in *Proceedings of ICSLP 2000*, Beijing, China, Oct. 16-20 2000, vol. 4, pp. 464–467.

[14] L. Bell, J. Boye, and J. Gustafson, "Real-time Handling of Fragmented Utterances," in *Proceedings of the NAACL Workshop on Adaption in Dialogue Systems*, Pittsburgh, PA, June 2001.

[15] J. Edlund and M. Nordstrand, "Turn-taking gestures and hour-glasses in a multi-modal dialogue system," *IDS'02*, Kloster Irsee, Germany, under review, 2002.

[16] `http://www.qualisys.se`.