Building Spoken Dialogue Systems for Believable Characters

Johan Bos, Tetsushi Oka
ICCS, School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, United Kingdom
{jbos,okat}@inf.ed.ac.uk

1 Introduction

Believable characters are defined in the arts as autonomous characters with personality, capable of showing emotion, being self-motivated rather than event-driven, adaptable to new situations, maintaining consistency of expression, and able to interact with other characters. These requirements impose a number of conversational capabilities on a spoken dialogue interface with a believable character: the character should react on its name, it should give subtle signals when it is listening or wants to take the turn, and it should allow interruptions while it is talking.

Implementing these conversational aspects of believability presupposes a flexible architecture capable of dealing with asynchronous processes. Most of today's spoken dialogue systems exhibit a pipelined architecture and even software agent-based architectures with the potential to function asynchronously work in practice as a pipeline approach. Spoken interaction with a believable character is, in the simplest case, replacing the speech synthesis component with a module that captures both the animation of the character plus speech synthesis. It is obvious that processing on a purely sequential basis wouldn't meet the requirements for believability—this paper introduces a dialogue system based on an asynchronous agent architecture developed within the MagiCster project, aiming to display the aforementioned believable aspects.

2 Face-to-Face Spoken Dialogue Systems

Our spoken dialogue system prototype is built around the embodied believable conversational agent Greta (see Figure 1) developed in MagiCster (www.ltg.ed.ac.uk/magicster/). This animation character includes detailed emotion modelling, within the application domain of giving advice to teenagers about eating disorders (Pelachaud et al., 2002).

In the framework of this system, we focussed on implementing several subtle but effective non-linguistic aspects of conversation, which we believe contribute to an improved sense of believability: allowing interruptions when Greta is speaking; system back-channelling (generating uhm's, nodding) and showing attention when the user is speaking; signalling awareness of the presence/absence of the conversational partner.

3 OAA for Spoken Dialogue Systems

Realisation of these capabilities in a spoken dialogue system requires a flexible asynchronous architecture. The Open Agent Architecture (OAA, www.ai.sri.com/~oaa/) is a framework for integrating software agents, possibly coded in different programming languages and running on different platforms. The OAA framework forms a piece of middleware allowing smooth integration of software agents for asynchronous dialogue systems in a prototyping environment.

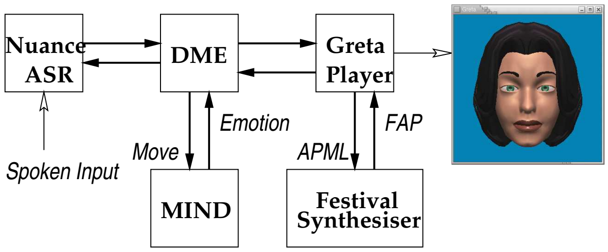

Figure 1: System Architecture

Figure 1 shows the components implemented as OAA agents. We adopted the grammar-based approach to language modelling that Nuance speech tools support (www.nuance.com). Using the slot-filling option of GSL, the value returned by the speech recogniser is the symbolic move of the utterance. The recognised string, the move, and an acoustic confidence value are sent as input to the dialogue move engine. Different speech grammars are loaded during different stages in the dialogue to increase performance of speech recognition.

The dialogue move engine (DME) controls all input and output relevant to the dialogue, interprets the user's move, and plans the system's next move. It can activate the ASR with a particular grammar, and receives information from the ASR (recognised moves, start of speech). It generates the system's move in the form of APML (Affective Presentation Markup Language), and sends this to the Greta Player. It is also able to tell the player to stop and it receives information when it is finished. The DME implements the information-state based approach to dialogue modelling.

The MIND component models emotions using dynamic belief networks, following the BDI (believes, desires, attentions) model (Pelachaud et al., 2002). Given a symbolic move sent by the DME, MIND decides which affective state should be activated and whether certain emotions should be displayed and how.

Greta is the talking head that plays FAP files based on timing information received from the Festival synthesiser. Greta combines verbal and nonverbal signals in an appropriate way when delivering information, achieving a very rich level of expressiveness during conversation (Pasquariello and Pelachaud, 2001). Using the APML markup language, it is able to express emotions, and synchronise lip and facial movements (eyebrows, gaze) with speech.

The Festival synthesiser (`www.shlrc.mq.edu.au/festdoc/festival/`) takes as input an APML file and produces both a waveform and timing information structured according to syllables and words. This information is needed by the Greta player to determine gestures and other movements, including facial movements such us eyebrow lifting and movement of lips and other articulators (Pasquariello and Pelachaud, 2001).

4 Information States in DIPPER

The information-state approach to dialogue modelling (Larsson and Traum, 2000) combines the strengths of dialogue-state and plan-based approaches, using aspects of dialogue state as well as the potential to include detailed semantic representations and notions of obligation, commitment, beliefs, and plans. It allows a declarative representation of dialogue modelling and is characterised by the following components: (1) a specification of the contents of the information state of the dialogue; (2) the datatypes used to structure the information state; (3) a set of update rules covering the dynamic changes of the information state; and (4) a control strategy for information state updates.

Our implementation of the information-state approach, DIPPER, follows the TrindiKit (Larsson and Traum, 2000) closely, by taking its record structure and datatypes to define information states. However, in contrast to the TrindiKit, in DIPPER all control is conveyed by the update rules, there are no additional update and control algorithms, and there is no separation between update and selection rules. Furthermore, the update rules in DIPPER do not rely on Prolog unification and backtracking, and allow developers to use OAA-solvables in the effects (Bos et al., 2003).

Update rules specify the information state change potential in a declarative way: applying an update rule to an information state results in a new state. An update rule is a triple $\langle$*name*, *conditions*, *effects*$\rangle$, The conditions and effects are defined by an update language, and both are recursively defined over terms. The terms of the update language allows one to refer to a specific value within the information state, either for testing a condition, or for applying an effect. In DIPPER this is done in a functional rather than a relational way as implemented by the TrindiKit, effectively abstracting away from Prolog syntax and discarding the use of Prolog variables.

We took the update rules of the GODIS system (Larsson and Traum, 2000) as a basis for our dialogue system for Greta. To implement the aspects of conversational believability in our system, several update rules were added, working mostly on the attentional state such as user-awareness and back channelling.

5 Conclusion

Implementing complex spoken dialogue systems—systems that go beyond the rather straightforward pipelined architectures—poses several requirements on the flexibility on dialogue modelling and the underlying architectures. We presented a system involving spoken interaction with an embodied character showing several believable characteristics of conversational behaviour, using the DIPPER framework for building spoken dialogue systems (`www.ltg.ed.ac.uk/dipper/`). We argued that OAA combined with an information-state theory of dialogue modelling is a good way of managing asynchronous processes that are imposed by these phenomena. We further claimed that an optimisation of the TrindiKit, where all dialogue controlled is specified by update rules, in combination with a language for update rules that abstracts away from Prolog and is tighter integrated with OAA, improves the facilities in the developer's workbench for complex spoken dialogue systems.

References

Johan Bos, Ewan Klein, Oliver Lemon, and Tetsushi Oka. 2003. Dipper: Description and formalisation of an information-state update dialogue system architecture. In *4th SIGdial Workshop on Discourse and Dialogue*. Sapporo, Japan.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, 5(3–4):323–340.

S Pasquariello and C. Pelachaud. 2001. Greta: A simple facial animation engine. In *6th Online World Conference on Soft Computing in Industrial Appications*.

C. Pelachaud, V. Carofiglio, B. De Carolis, F. de Rosis, and I. Poggi. 2002. Embodied contextual agents for information delivery applications. In *Proceedings of AAMAS'02, Bologna*.