BARGE-IN REVISED

B. Kaspar, K. Schuhmacher, S. Feldes
Deutsche Telekom Berkom GmbH
Research Group Speech Processing
D-64295 Darmstadt, Germany
email {kaspar,schuhm,feldes}@tzd.telekom.de

ABSTRACT

We consider speech dialogues, allowing for simultaneous input (via speech recognition) and output (via speech synthesis or pre-recorded prompts), often referred to as "barge in". We start with a collection of dialogue situations, where simultaneous input and output is useful. It is argued, that a variety of possible system behaviour is necessary in order to take into account these situations adequately. We then define a formalism, that allows to control this system behaviour. We end up with reporting some experience gathered both in lab tests and a in real world pilot.

1. INTRODUCTION

In human conversation, it is quite usual, that both partners speak simultaneously, at least partially. One may experience this in cases, when double talk is explicitly suppressed by the communication channel. In order to take account of this user behaviour, many state-of-the-art dialogue systems exhibit an option of simultaneous speech input and output, mostly under the name "barge-in" (synonyms used include "Cut-Through", "User Priority Mode", "Talk-over"); for examples, see [1,2,3,4]. The intended use of such a facility is, e.g. getting rid of system beeps or free users from the discipline of waiting for the end of long prompts.

A user trying different systems capable of "barge-in" may, however, experience a variety of system behaviours, e.g. the speech output may stop immediately when he/she starts speaking, may stop with delay or not at all. Scanning system descriptions, it is hard to find any precise description of what system behaviour is meant with "barge-in". In the following, we try to fill this gap :

We describe different dialogue situations, where simultaneous input and output is desired. We then define a set of system behaviours, appropriate for these dialogue situations, based on different influencing factors. In the next chapter, a formalism is outlined to control the system behaviour. We conclude with some experience, partly derived from real-world tests.

2. DIALOGUE SITUATIONS

Before listing different situations, we restrict the system behaviour, that is covered by our considerations, by the following constraint :
User input is interrupted in cases of time-out only, that is, if the user does not react to a question in time or his/her answer is too lengthy. We do not consider cases where the system "barges in" based on the content of the user utterance; (s. [5] for such an application).

The user may "barge-in" in the following dialogue situations :

- The user wants to stop a lengthy system output. A common situation is, that an (experienced) user wants to stop a help prompt. Another option is, to move forward or backward in lists of announcements.
- The user intentionally reacts to a system prompt, before it is finished. Again, this might be an option for an experienced user, who already has heard all information needed.
- The user unconsciously answers, before the prompt is finished. This behaviour is experienced quite often. It might be diminished, but not excluded, by well designed prompts, outlining its essence at the end. Such a situation may arise more often in mixed-initiative dialogues.

The dialogue system may react differently to these user actions:

- It may be deaf during (part of) the output, not reacting to input at all.
- It may be listening during (part of) the output, but nevertheless finishing the output.
- It may be listening and interrupting the output on demand, possibly after some reaction time.

Allowing for barge-in from the system designer point of view should depend on whether the user has heard all information necessary to continue the dialogue. Reactions on barge-in cases should take into account some reaction time as inherent in usual dialogues.

Of course, the problem remains, to tell the user, at what time he/she is allowed to barge in.

3. TECHNOLOGY REQUIREMENTS

3.1 General

We do not describe speech technology in general, here. Rather, we list some requirements for components, that are not commonplace, but necessary in order to allow for the system behaviour as described below. One common requirement is, that these components are able to operate in asynchronous mode and are interruptible. (The first condition might be somewhat relaxed.)

3.2 Speech detection

This component decides, whether incoming sounds are classified as speech. It may also decide, if speech has ended. See, e.g. [2]. In order to simplify explanations, we assume in the following, that speech detection appears as a "sleep-mode" of the recogniser.

3.3 Speech recognition

The recogniser should have the option, to reject a user utterance. This allows, among others, for the implementation of a "re-entering-mode", causing the recogniser, to start again automatically after having heard a out-of-vocabulary word. Furthermore, the recogniser or it's controlling software should exhibit a time-out facility, stopping the recognition if the user utterance is too long.

3.4 Speech Output

Our considerations are independent of the type of output (synthesised or pre-recorded). What is desirable, however, is a signalling telling the application, that the output is "almost finished". This is easy for fixed prompts, where the length can be measured in advance, but harder for synthesisers.

3.5 Echo Cancellation

Echoes are inevitable, if recogniser and output device run in parallel. Current echo cancellation devices work reliably for line echo and hybrids, even though there is some weakness in case of non-linear distortions. There is no solution to compensate for room echoes (at the far end) reliably over the telephone line, unless this is solved within the handset. Echo cancellation is enabled, whenever the recogniser is listening and is therefor not mentioned in the following.

4. SYSTEM DESIGN

4.1 Features to be controlled

The features to be controlled include timing, interruptability of the output, and triggering.

Output and input may exhibit the following timing patterns :

- Sequential : input is allowed, when output has finished, only.
- Overlapping : input is allowed before output has finished (including the special case, that it even may finish, before output is finished). In order to specify the amount of overlapping, a delay has to be introduced.

Additionally, timing might be influenced by controlling the input window via timers.

A demand to stop the output may be triggered by the following events of the input device, (s. figure 1).

- On speech detection
- On speech recognition (whenever something has been recognised, even if it is rejected).
- On special keywords (whenever given keywords have been recognised, e.g. "wake-up").

As a side effect, these methods cause different delays in the interruption of the output.

The output may or may not react on demands for interruption, based on the current setting.

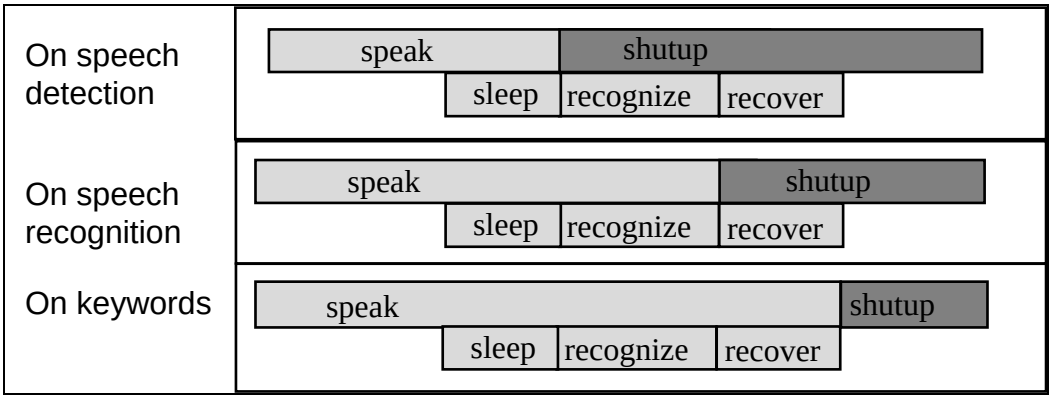

Figure 1 : Triggering

4.2 Control formalism

The processes involved are

- Speech Synthesis
- Speech Recognition

Additionally, we need the following timers

- DelayTimer
- TimeoutTimer (for Dialogue Time-out)
- RecoverTimer (e.g. for processing steps after word recognition)

All processes should be capable of running in asynchronous mode.

The parameters that can be set according to the dialogue situation are :

- Delay : Time period from start of speech output to enabling of speech detection
- RecoverTimeout : Time period allowed for speech recognition, to supply a result after timeout (more important in phrase recognition).
- DialogueTimeout : Maximal time period before start of user utterance.
- RecognitionTimeout : Maximum duration of user utterance.
- OutputInterruptability : control flag, whether the output should stop in a barge-in case or not.
- Recognition mode (triggering).

The system control can now be described in terms of a finite-state-machine, as outlined in figure 2, with the following simplifications :

- The special cases with recognition only or synthesis only are missing.
- The influence of both triggering mode and output interruptibility is hidden in the drawing.

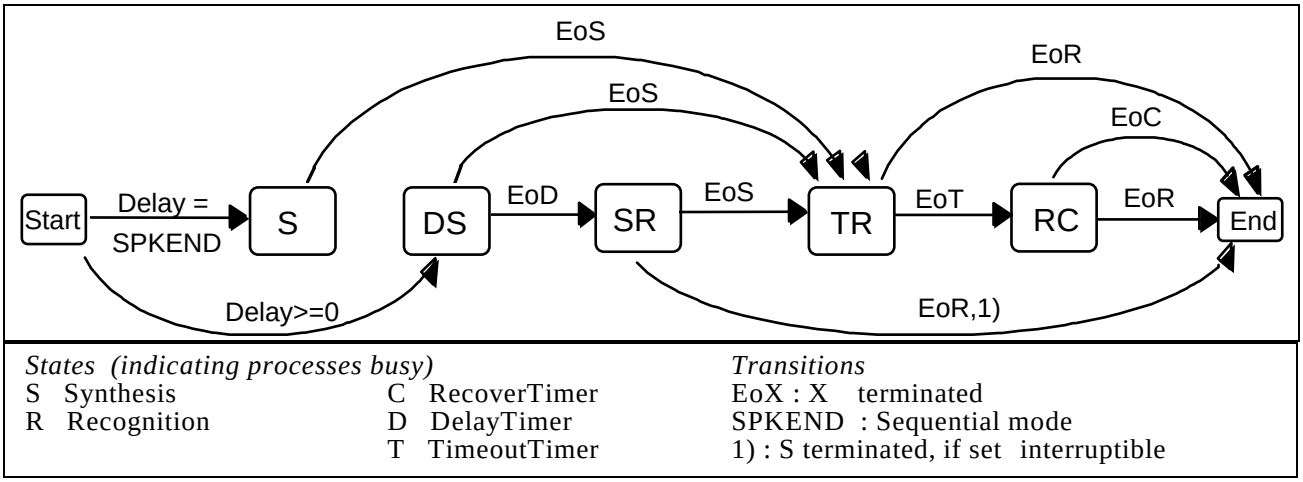

Figure 2 : Finite state machine

5. EXPERIENCE AND CONCLUSION

5.1 Tests

The formalism as described above was used to handle different dialogue situations in the SPRADIAK system [1]; speech recognition is implemented in form of simple keyword spotting. There are no beeps after the prompts. The system was tested in the lab and during a pilot for the German directory assistance service of Deutsche Telekom.

We list some informal results concerning barge-in.

- Unconscious barge-in cases might happen everywhere within the dialogue, even if the prompts are designed such as to avoid this. (Similar observations were made testing the system FAUST [6]).
- Delay is, as far as possible, set to "NearEnd". This decision was not guided by principles of the dialogue design, but rather by the weakness of the echo cancelling device with respect to certain line effects.
- Triggering on speech detection proved to be unfeasible. False alarms as well as sudden reaction (by interrupting the speech output) lead to confusion on the user side. The system should exhibit some reaction time, as usual in human conversation. Triggering on speech recognition is a natural way of introducing this behaviour.

5.2 Proposal for naming conventions

Based on the above experience, three standard configurations for different "barge-in"-behaviour can be derived. We also propose a naming convention, thus attributing distinct system behaviour to the previously synonyms "barge-in", "talk-over", and "cut-through".

Barge-In : provided for the case of early answering. It is recommended, to provide the technique silently, not encouraging users explicitly to "barge-in". (In the same sense, keyword- or phrase spotting should be provided silently.)

Delay is set to "NearEnd". Output is not interrupted.

Talk-over : provided for expert users or the case, where the user is explicitly informed about this option. Its main purpose is, to abbreviate the dialogue. A typical situation is, that the user does not listen to all the given choices of a menu, as he/she already knows them.

Delay is set according to the information inside the prompt. Triggering is on recognition. Output is interrupted.

Cut-through : provided again for the case, where the user is explicitly informed about this option. The main purpose is to stop help information or to navigate in lists. In these situations, the vocabulary usually makes implicitly clear, that an interruption is intended; (e.g. "Stop", "Go on", "Forward").

Delay is set according to the information inside the prompt. Triggering is on keywords. Output is interrupted.

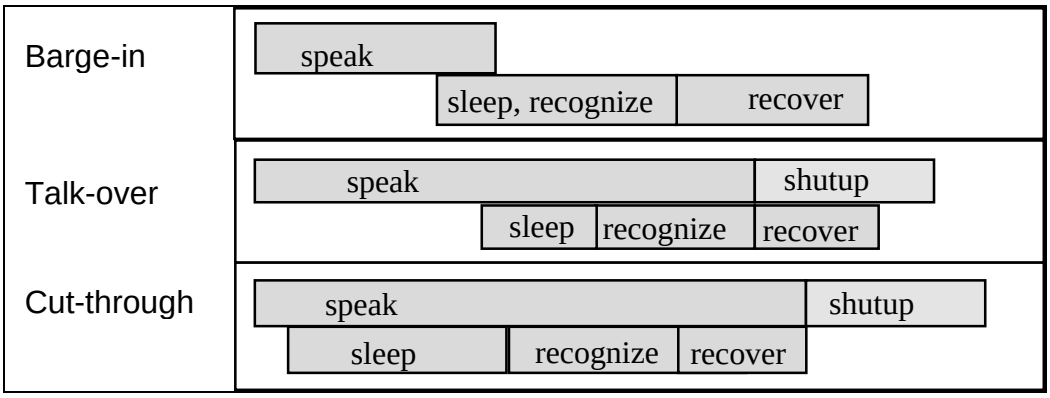

Figure 3 : Prototypical situations

6. REFERENCES

[1] B. Kaspar, G. Fries, K. Schuhmacher, A. Wirth: SPRADIAK- Directory Assistance Pilot. To appear in proceedings VOICE'97.
[2] K.J.Power : The listening telephone - automatic speech recognition over the PSTN. BT Technology Journal Vol14No.1, 1996, pp.112.
[3] H. C. Leung, J.R. Spitz : Interactive Speech and Language Systems for Telecommunications Applications at NYNEX. Proceedings IVTTA'94, pp.49.
[4] S. Springer, S. Basson, A. Kalyanswamy, E. Man, D. Yashchin : The MoneyTalks Interactive Speech Technology Assesment. Proce. Eurospeech '95, pp.1939.
[5] Y. Okato, K. Kato, M. Yamamoto, S. Itahashi : Insertion of Interjectory Response Based on Prosodic Information. Proc. IVTTA'96, pp. 85.
[6] B. Kaspar, G. Fries, K. Schuhmacher, A. Wirth: FAUST - A Directory Assistance Demonstrator. Proc. Eurospeech'95, pp. 1161.