WORKING NOTE D7

Alan Creak
17 June 1986

PROPOSALS FOR A COMMUNICATOR USING A SYNTHESISED VOICE

THE POSITION.

We want to develop a computer-driven device which will use a speech synthesiser to speak words or phrases selected by the person using the system. The computer and synthesiser are defined : Amstrad CPC664 and Amsoft SSA2 (? – that's the number mentioned in the manual) respectively. There is a suggestion that the main use of the device will be for conducting telephone conversations; that's relevant, because it means that clarity is essential – there's no chance of augmenting the spoken words with gestures or facial expressions.

Some software has already been written. Software to drive the synthesiser is supplied (SSA1) and a Basic programme which uses this (RDEV1) has been started.

There is no documentation for any of this, except for the Amstrad introductory manual. (Amstrad CPC664 User Instructions (1985))

SOME NOTES.

The existing Basic programme is written as a (stumpy) tree of menus. It begins by displaying a selection menu; selecting one of the "dishes" from that menu produces another menu of phrases which can be said. The whole is rigidly encoded within the Basic programme.

This seems to us to be less than adequate. The number of phrases which are easily accessible by such means is restricted, as the programme must fit completely within the computer; it may be enough to express primary needs for help, food, and service, but it's unlikely to extend satisfactorily into a system which can handle even a restricted telephone conversation. As well as that, the system is inflexible : the person using it cannot edit the existing phrases, or insert new ones as they become useful, without fairly careful fiddling about with the Basic programme itself.

The sound generated by the synthesiser leaves a lot to be desired; but until we can find some documentation for it, we can't tell whether there's hope for improvement. We would like to be able to control pitch, speed, and inflection.

The easiest language to use is Amstrad Basic. CP/M is available, so Microsoft Basic is presumably possible in principle : that would offer random access files, which aren't provided in the Amstrad version. It would not necessarily give easy access to the Amstrad system instructions, nor would there necessarily be a simple way of linking a programme to the SSA1 software. The manual also mentions an assembler, but this is also – apparently – a CP/M thing, and other documentation is needed.

PROPOSALS.

We think that a better system could be constructed along these lines :

1. The menus should be kept as data files, not built into the programme. This means that
   - o the programme can be used with other sets of menus if required;
   - o it becomes much easier to change the menus and to add new bits;
   - o there's rather more space for the menus, so more variety is possible.

2. A selection of common phrases should be available all the time – things like "yes, please", "no, thank you", "good morning", "please say that again".

3. It should be possible to represent "multiple" phrases : "I am hungry | thirsty | tired | hot | cold | _" shouldn't all need to be included separately.

4. There should be provision for repeating the last phrase spoken, in case the person at the other end of the telephone doesn't catch it. (The synthesised voice isn't always clear.) It should also be possible to repeat *part* of the spoken phrase.

5. Another programme will be required for constructing and editing the phrases to be spoken, and other items which can appear in the menus. It should be possible to enter a phrase and its phonetic equivalent, and to experiment with both until a satisfactory result is obtained – then the whole can be incorporated into the menu.

6. SPECULATIVE : In many cases, a sentence can be transformed into another useful sentence by a simple standard change. Consider :

   The dog chased the cat.
   Did the dog chase the cat ?
   Can the dog chase the cat ?
   The dog chased the cat.

   If it were possible to exploit such transformations easily, a much wider vocabulary of phrases would be accessible from a comparatively limited volume of text.

WHAT NEXT ?

We intend to start work on a system incorporating points (1) to (5) of those listed above. We'll keep thinking about (6). For the present, at least, we'll stick to Amstrad Basic, but we foresee some difficulties with the file structures needed which may lead us to think again.

We'll begin by making a simple prototype, which we can then discuss with Eddie. This will look fairly like the present version, though it'll be rather more complete. We don't want to spend a lot of time constructing software that turns out to be unusable !