Proceedings of the Institute of Acoustics

THE SPEECH FILING SYSTEM: A TOOL FOR COOPERATIVE SPEECH RESEARCH

M D Edgington, C M Barnes, P D Stringer & D M Howard

Signal Processing: Voice and Hearing Research Group,
Department of Electronics, University of York, Heslington, York

ABSTRACT

The Speech Filing System (SFS) is a tool for manipulating processed audio data. It consists of a file format definition, 'C' source code for file interface and display interface routines, and a suite of utility and processing programs. It can run on a variety of hardware platforms, which along with the freely available source code, gives SFS the potential of providing an informal mechanism for easy dissemination of data and speech processing implementations through the speech community.

1. INTRODUCTION

The Speech Filing System (SFS) is a tool for the management of files containing speech and related data. It was originally developed at University College London under the Alvey project as 'SPARBASE', and has undergone several refinements since.

SFS is in use at York on a number of different platforms, from Personal Computers, to UNIX workstations. However, to allow very similar operation on all hardware, the interface has been deliberately left simple; command line interfaces are used for most processing programs, and a straightforward mouse interface is used in the interactive display program. There are no facilities for multi-window operation or mouse based cut-and-paste. While this may be seen as a limitation, it does lead to simple implementation on many hardware platforms, a significant advantage over many other systems.

The source code for the whole SFS system is freely available, allowing users to adapt parts of the system for use with their particular hardware facilities, for example display devices and data acquisition boards. The flexibility of the system, and the compatability over different hardware platforms, has made SFS particularly useful for collaboration between research projects, both within our group at York, and with external bodies.

The following section of this paper gives a short history of the motivation for and development of SFS. A description of some of the main features of SFS follows that, which is neccesarily only an overview of the system. Finally a simple example of the use of SFS is presented.

2. HISTORY OF SFS

The speech filing system resulted from an Alvey funded research project based at University College London on speech patterns and algorithmic representations (SPAR) [1]. At the outset of the SPAR project in the mid 1980s a survey of available speech analysis systems was carried out and none was deemed both convenient and appropriate for use in the project. One particular requirement was for portability of software and exchange of data files between project partners. Speech filing protocols available at the time were designed to create a new data file each time an analysis was carried out on an existing stored data item for the output of that analysis. In a research environment where analysis routines are being developed and new ideas are being constantly investigated, large numbers of data files are created during the research and program development stages of the work and directories can become very large. In particular, the maintenance of data files relies essentially on available system commands for handling files within directories.

The fundamental difference in the design of the SFS system is the use of a single data file containing all data sets associated with the originally captured signal. This data file expands when analyses are carried out on the dafa as the outputs are appended to the file. An item header block is associated with each item in the SFS file which enables convenient data management and an overall main header is used to log the source of the original data. To port the data with its analyses to another machine is simply a matter of copying one file. The main drawback with this system is that the files can become very large and they can soon outgrow a single floppy diskette. However, the availability of networked electronic file transfer and transfer via back-up systems such as tape and cartridge rather than floppy diskettes, means that the disadvantages of moving a single large file rather than many smaller files is far outweighed by the simplicity of local data housekeeping.

3. DESCRIPTION

A schematic diagram showing the structure of SFS is presented in Figure 1. In general the structure assumes that processing of data is separated from the display of data, precluding some forms of interactive data analysis. The following sections describe the main features of the file structure, processing history information, and display interfacing.

3.1 File Format

SFS normally stores all the data derived from a single speech recording in one file, although it is possible to hold the data in separate files and store index information in the main SFS file. The SFS file format comprises of a set of filing system headers and speech data structures. The *main header* occupies the start of every SFS database file and describes and records information concerning the creation of the file and its contents. It contains fields such as speaker identification, session reference date and code, recording conditions and token spoken.

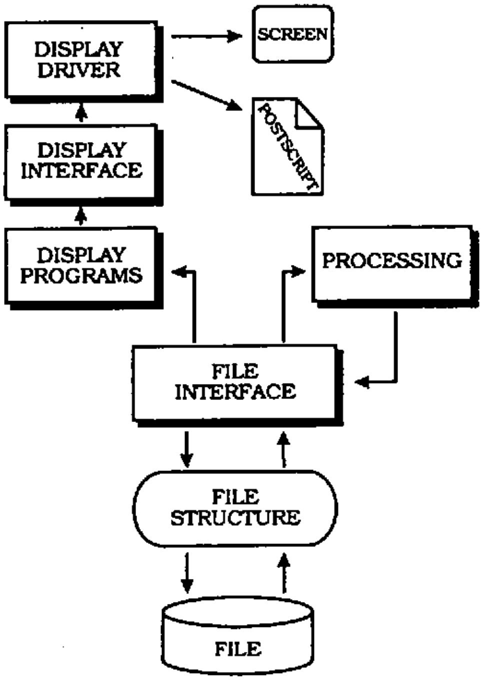

Figure 1: A schematic diagram of the SFS structure.

An SFS *item header* is used as the prefix for every data set in the file. It contains information about the format of the data set including its processing history, sampling frequency and data length.

A *link header* can replace an item header for some data sets. In this case the data set does not follow the header, but is stored elsewhere; the link header contains details of where the data set can be found.

3.2 Item Structures

Some of the important parameters in the SFS item header are described below.

History This field records the processing history of the data set which follows in the file. Common information stored in the history field may include:

- the name of the program that generated the data set.
- the input items used the generate this item.
- flags and other data describing how the data was processed.

Process date The date and time when the data set was generated. This is field is automatically generated by SFS.

Data types Items are given a basic type by SFS, which is determined by the type of data they hold. Some of the data types are appropriate for data sets consisting of a stream of samples, others impose a particular record structure on the data set. It is possible for a user to define a new data type if necessary, although one of the 30 types already defined is usually appropriate for the work being conducted. Some of the more common data types are:

- speech pressure waveform
- laryngograph waveform
- fundamental period markers
- fundamental frequency trace
- annotations
- greyscale display
- formant frequencies and bandwidths
- general acoustic tracks

Frame duration For unstructured data types such as speech and laryngograph items, this field records the time interval between consecutive data frames, i.e. the sampling period.

Machine This field contains the machine format for data stored in the item. For example 68000 based systems store data hi-byte first, whereas 8086 based systems store data lo-byte first. This flag is decoded automatically by the SFS routines, making storage differences transparent to the user.

3.3 Processing History

As described above, each data set has a history field in its header record. The history details the program which created the data set, and any data sets which were used in the generation. Thus it is possible to automatically create an 'audit trail' providing a complete history of all the processing stages leading to the current data. This feature is particularly useful when analysis routines are being investigated and developed, where a precise record of the analysis history is essential. The history information can be presented in two forms: the raw history, which simply produces a copy of the history fields, or a labelled history, which uses text labelling to produce a text interpretation of the history.

As an example consider the following raw history:

```
1.   SPEECH  (1.01) 36322 frames from speech
2.   SPEECH  (1.02) 11350 frames from ifd(1.01;low-pass=4500Hz)
3.    COEFF (11.01)   868 frames from spectran(1.02;window=8,overlap=6)
4.  DISPLAY  (9.01)   868 frames from dicode(11.01;dbr=50.00,nump=128)
```

The text labelling routines refer to a label file which defines the form of the labels. This file indicates that, for example, `ifd` is an interpolation and decimation program that changes the sampling rate of speech items, and provides the text description "`up/down sampled`". From the history, `ifd` operated on item `1.01`, which has the history "`speech`". There is no entry in the label file for "`speech`", so it is simply copied, producing the description "`up/down sampled speech`". The labelled history version of the file used above becomes:

```
 1.  SPEECH (SP.01) 36322 frames from speech
 2.  SPEECH (SP.02) 11350 frames of up/down sampled speech
 3.   COEFF (CO.01)   868 frames of spectra
 4. DISPLAY (DI.01)   868 frames of spectrogram
```

For complex files, labelled histories provide a very useful and necessary outline of the file contents.

3.4 Display Interface

SFS was written to use a set of standard procedure calls for interaction with graphics output devices and input devices. These procedures access tables which point to appropriate functions for the current device. Some functions, such as straight line drawing, are mandatory, while others are optional. If a particular function is not available for a device, this is indicated in the table. In this way the software can be written to be independent of a particular graphics device. For example, there is a standard call to draw a circle. The appropriate table is examined, and if there is a valid function entry, the function is called. If the device does not support circle drawing, then the calling routine decomposes the circle into a set of straight lines, and uses the line drawing procedure.

Although the extra levels of software slow down the drawing rate of graphics programs, all programs can produce graphics on any supported device, (without recompilation) simply by changing an environment variable switch. We have written a device interface for a PostScript file generator, so hardcopy is easily generated using exactly the same software as used for screen output.

4. EXAMPLE OF USE

This section follows an idealised analysis session using SFS. For the following example, the display device was set as a PostScript file, which was included directly into this paper.

Firstly the datafile is created, and general information put into the file header:

```
Main Header
File id       : PDS-395 created Mon Sep 28 11:52:48 1992 by Paul Stringer
Database      : SPIN 2.1        Speaker        : mde
Session       : spin tests      Session Date   : 08/8/92
Token         : one two         Environment    : deadened room
```

The acoustic waveform is captured using the hardware dependent input device, and stored as the first item in the file. For further processing, the sample rate needs to be reduced to 10 KHz, so an interpolation and decimation program is used. There are now two speech items in the file. The next stage is to generate a spectrogram, which is done in two steps, producing a sequence of spectra as an intermediate step. The spectra item is not needed, so an SFS utility program is used to remove it from the file, although its header information is retained because it forms part of the processing history of the spectrogram. A display of the file contents after this stage is shown in Figure 2.

Further processing is performed on the file to generate the energy contour of and average zero crossing rate of the speech. Both of these data sets are stored as general acoustic track items, which simply consists of a sequence of floating point numbers. An LPC analysis of the speech generates raw formant estimates, which are stored in a formant structure. The displayed file now looks like Figure 3.

5. CONCLUSION

This paper has briefly described the main features of SFS. As a tool for collabaration between research projects, it has been found to be particularly useful, due especially to the hardware independence of the system. We would recommend its use as one of the range of tools available in any speech research laboratory.

ACKNOWLEDGEMENTS

Thanks are due to UCL and Mark Huckvale for enabling the transfer of SFS to other platforms.

REFERENCES

[1] Huckvale, M., Brookes, D., Dworkin, L., Johnson, M., Pearce, D., Whitacker, L. "The SPAR speech filing system." *Proceedings of the European Conference on Speech Technology,* **1**, pp305–317. (1987)

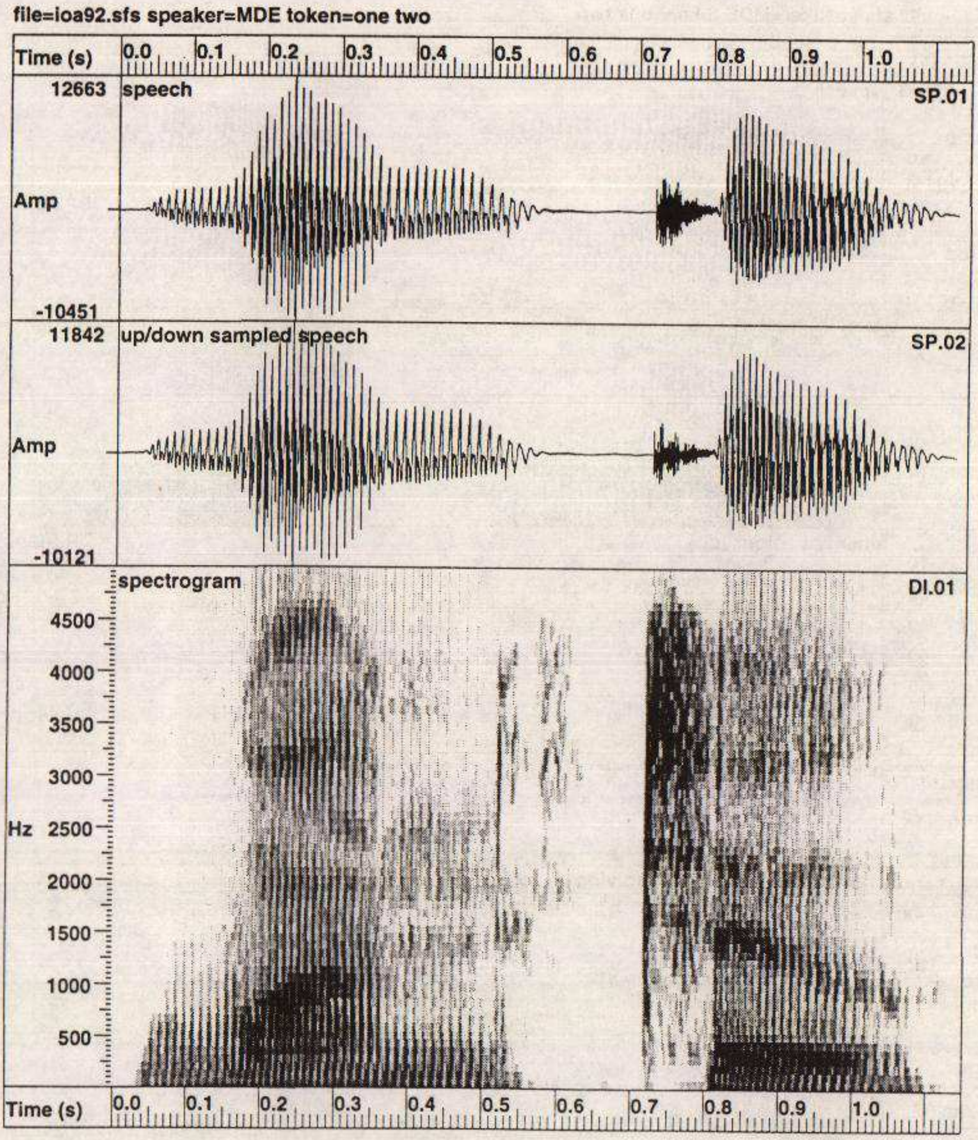

Figure 2: The example file showing speech and spectrogram items.

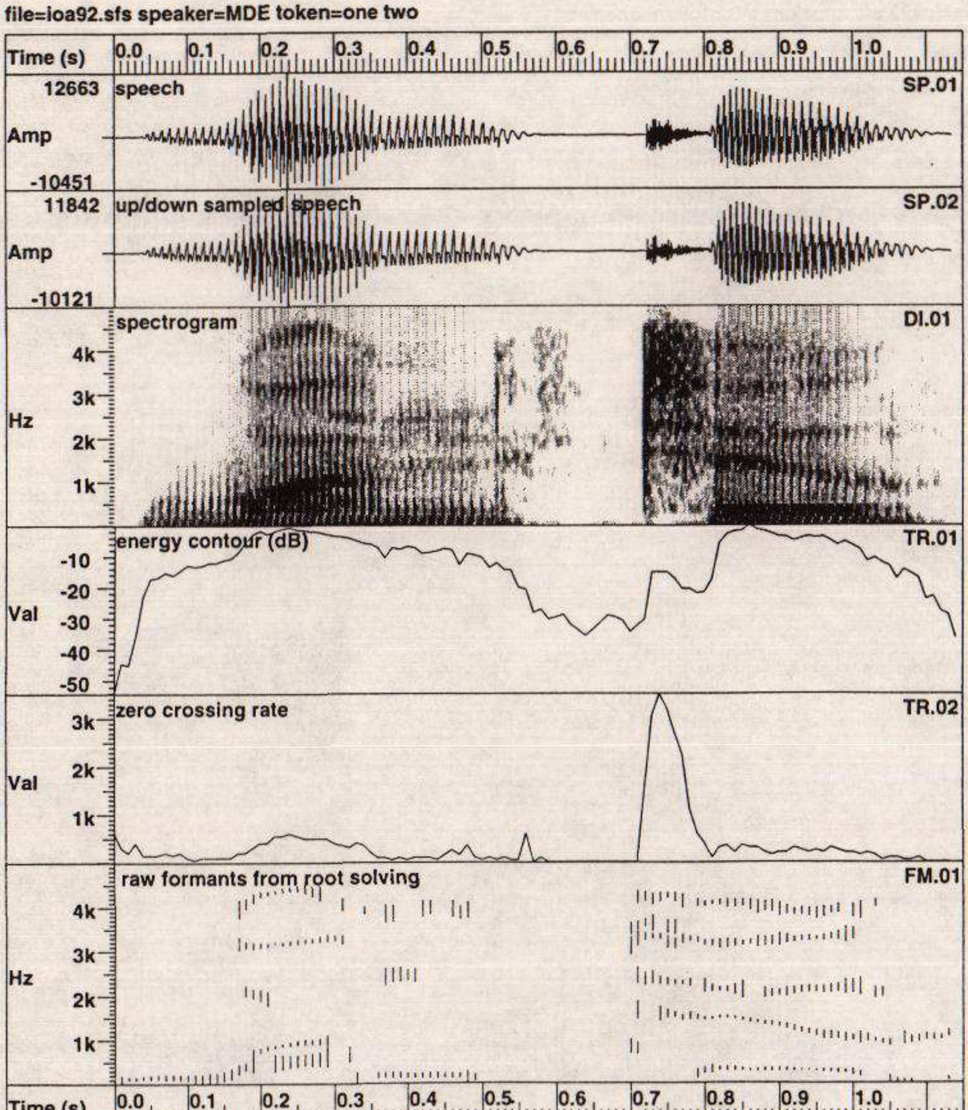

Figure 3: The example file after further processing.