A Graphical Parametric Language-Independent Tool for the Annotation of Speech Corpora

Kallirroi Georgila, Nikos Fakotakis, George Kokkinakis

Wire Communications Laboratory
Electrical and Computer Engineering Dept.
University of Patras
261 10 Rion, Patras, Greece
{rgeorgil, fakotaki, gkokkin}@wcl.ee.upatras.gr

Abstract

Robust speech recognizers and synthesizers require well-annotated corpora in order to be trained and tested, thus making speech annotation tools crucial in speech technology. It is very important that these tools are parametric so that they can handle various directory and file structures and deal with different waveform and transcription formats. They should also be language-independent, provide a user-friendly interface or even interact with other kinds of speech processing software. In this paper we describe an efficient tool able to cope with the above requirements. It was first developed for the annotation of the SpeechDat-II recordings, and then it was extended to incorporate the additional features of the SpeechDat-Car project. Nevertheless, it has been parameterized so that it is not restricted to the SpeechDat format and Greek, and it can handle any other formalism and language.

1. Introduction

The development of robust systems for speech analysis and synthesis depends crucially on the availability of well-annotated corpora of naturally occurring, continuous speech. However, the transcription of the speech signal is one of the most difficult and time-consuming tasks. It requires both human experts and speech annotation tools that facilitate their work by providing a user-friendly interface, preferably graphical, which instructs the annotator, minimizes the number of steps s/he has to take in order to complete the annotation procedure and prevents him/her from making mistakes. In some tools a speech recognition module is also incorporated so that the annotator can use its on-line or off-line output as a reference for the transcription.

In addition, it is very important that whenever the source and target formats change depending on the application, the tools can adapt to the new requirements easily without having to modify the software. It is well known that even a slight change in the code of a program could cause severe problems to other modules not directly related to the modified parts. Thus, future demands should be predicted during the software development and parametric features must be incorporated to avoid the cost of modifying the program and testing its new version.

Another issue that has to be considered is the language that will be used. Some features depend on the language of the speech corpora, such as an incorporated speech recognizer or utilities that transform digits, numbers, letters etc. to their equivalent orthographic transcriptions. These features require significant effort when we move to a new language. On the other hand, menus, messages etc. do not depend on the corpora, that is one could work with a program in English but annotate Greek utterances, and are modified quite easily.

Finally, an efficient tool must allow more than one annotators, to work simultaneously on the speech database avoiding conflicts, and it should also be platform independent, that is work on multiple operating systems.

Our tool was first developed for the annotation of the SpeechDat-II recordings, and then it was extended to incorporate the additional features of the SpeechDat-Car project. SpeechDat is a European telephone speech database collection project funded by the European Union (LE2-4001) (Hoege et al, 1997). It aims at providing uniform databases for the development of voice driven telephone applications in most European languages. The objective of SpeechDat-Car (LE4-8334) is also to create speech databases for most European languages but in order to support training and testing of robust multilingual speech recognition for in-car applications (van den Heuvel et al, 1999).

Other existing tools that have been used for SpeechDat-II and SpeechDat-Car projects are the WWWTranscribe (Dept. of Phonetics, University of Munich) (Draxler, 1997), SPEXTATOR (Spex), Annotator (IDIAP), Vox (CSELT) etc (Constantinescu et al, 1997). Some of them were available for all the SpeechDat partners. However, we decided to develop our own tool to fulfil the specific requirements of the Greek language (Chatzi et al, 1997).

During the development of this tool and as we moved from SpeechDat-II to SpeechDat-Car we realized the advantages of parameterization so that the resulting software is suitable for both project formalisms but not restricted to them. We tried to incorporate all the characteristics of an efficient speech annotation tool as described above. The only requirement we did not concern ourselves with was the platform portability. We developed the program only on Windows since all our systems use this operating system. Nevertheless, we may deal with this issue in the future if demand arises.

The paper is organized as follows: The description of the tool is given in Section 2. In Section 3 we discuss some implementation issues and in Section 4 our experience from its use. Finally, a short summary and conclusions are given in Section 5.

2. Description

Before the annotation procedure starts, the user must decide on the language s/he is going to work with, the structure of directories and files and the waveform and transcription formats. Otherwise, the default or previously selected settings will be used. At this time s/he is ready to

start annotating the speech corpora. After part or all the data has been transcribed, statistical information about the annotated files can be extracted. In addition, a set of files could be selected according to user-defined criteria and directed to Entropic's Hidden Markov Model Toolkit (HTK) (Young et al, 1997) for training a speech recognizer. Each one of the above steps is described in detail in the following paragraphs.

2.1. Selection of language

The tool is language independent. All window labels and messages are written in text files. When the program is loaded it reads the "□□□□□□□□□" file where the name of the file that contains the labels and messages is stored. By default, the labels and messages are in Greek. If someone wants to add a new language s/he can do it in two ways. The first is to create a file containing the labels and messages in the new language following the format of the corresponding file in Greek. This is done, by using a simple text editor. However, for less experienced users a utility is provided by the tool itself. As shown in Figure 1a, the user can select the label or message s/he wants to change from a list. In the edit mode (Figure 1b) s/he is given the label or message in the current language and is expected to enter the corresponding text in the new one.

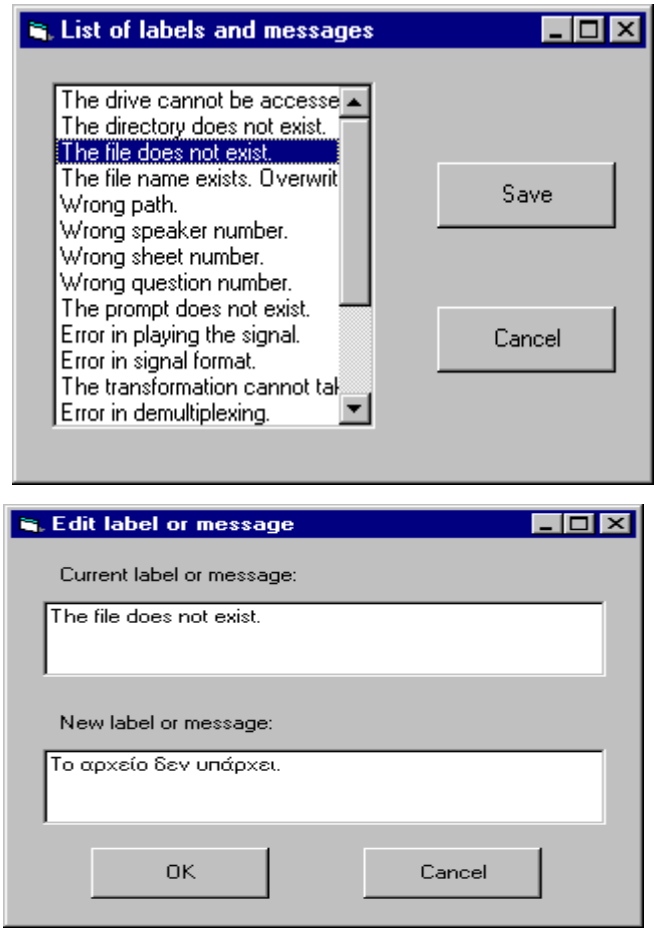

Figure 1. (a) List of labels and messages, (b) editing of a label or message.

The current language is the one that has been last selected or the default language, i.e. Greek, if the initial settings have not been modified. After the user has completed all the labels and messages, s/he must give the name of the file where this information will be stored. For the changes to take effect, one should exit the program and reload it.

The above features do not depend on the language of the speech corpora. That is, one can have labels and messages in English but annotate Greek speech files. However, there are utilities that enhance the annotation procedure and consequently depend on the language that the corpora have been uttered. In subsection 2.5 we will see that the annotator can transform digits, numbers, letters, dates and times to their orthographic transcriptions and turn characters from lower case to upper case and vice versa automatically by pressing the corresponding buttons. The necessary information for these transformations is stored in text files, the names of which are also written in "□□□□□□□□□". One can define the structures for the new language either by using a text editor following the format of the already existing corresponding files or by using a utility, which facilitates this procedure. An example of number format definition is given in Figures 2 and 3. In Figure 2 it is shown how the user can create a structure, in this case, the structure for numbers 20, 30 up to 90. Once the structures have been defined, s/he can combine them to create the format for N-digit numbers. An example for 3-digit numbers is depicted in Figure 3. In the first box we have a list of the existing structures. The user can combine these structures using the arrow buttons. Each line in the second box constitutes a different format of a 3-digit number, that is a 3-digit number could be formed by the structures "□□□□□□□□□□□□□" and "□□□□□□□", e.g. "one hundred", "seven hundred" etc. Note that the structure "□□□□□□□" is used just for one word. In the same way a 3-digit could be the combination of the structures "□□□□□□□□□□□□□", "□□□□□□□", "□□□" and "□□□□□□□", e.g "two hundred and eight" etc. The "New" button is used for creating a new format, that is a new line in the second box. Thus the user can define the formats of all N-digit numbers by substituting 3 with N in the text box.

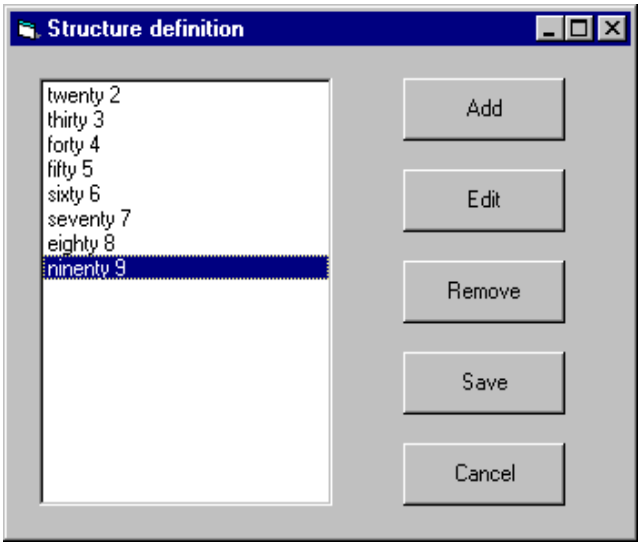

Figure 2. Structure definition

Date and time transformations are currently supported only for Greek and English due to their complexity and diversity of structure in different languages. To ensure that

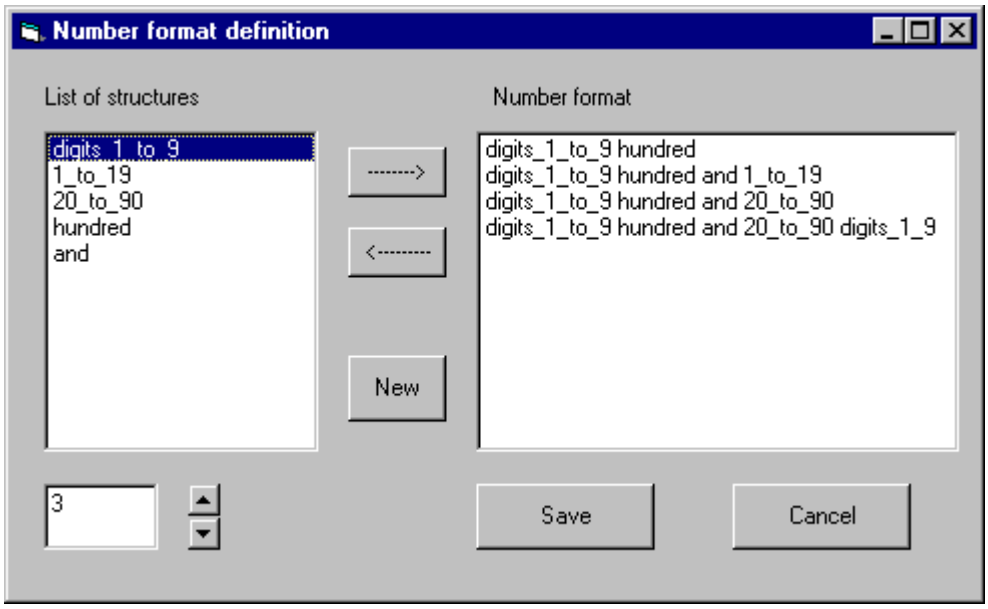

Figure 3. Definition of number formats

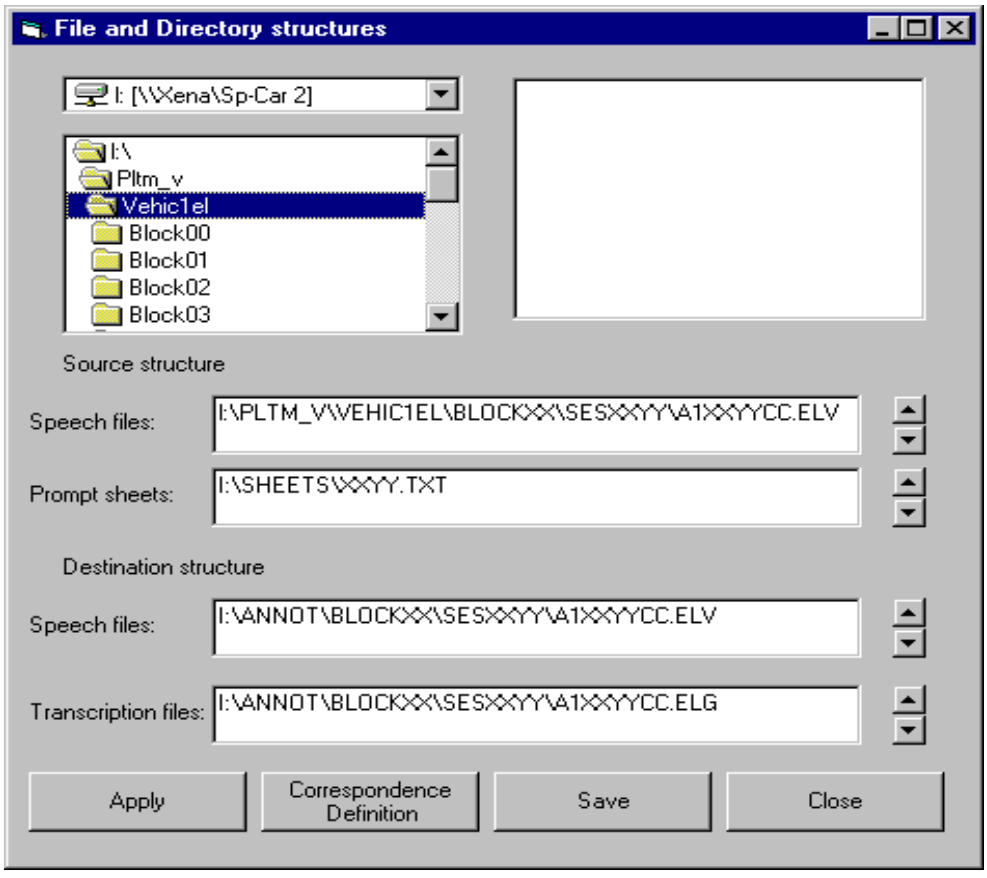

Figure 4. Definition of file and directory structures

the number, date and time transformations are correct, the annotator can test the structures s/he has defined by giving a number, date or time expression and getting its transcription. The changes take effect at once, that is the user does not have to exit and rerun the program.

2.2. Structure of directories and files

The user is allowed to choose whether s/he will work with single files or with sets of files uttered by the same speaker and sharing similar characteristics. If the annotator handles single files s/he must define the source directory where the speech file is located and the destination directory where it will be placed with its label file after annotation. The label file is a text file (see subsection 2.4) containing the utterance description and other available information. When utterances spoken by the same speaker are processed as a set, the procedure is somewhat more complicated. There are options for multiple source and destination directories with various

structures and file name formats. The following structure applies to SpeechDat-II:
Source Structure
□□□□□□□□□□□□□□□□□□□□□□□□□□ – speech files
□□□□□□□□□□□□□□□□□□ – prompt sheets
Destination Structure
□□
– speech files
□□
– transcription files,
where □□ is the file number, □□□□ the speaker number and □□□□ is the prompt sheet number. □□ indicates the type of file. The correspondence between □□ and □□ is user-defined, e.g. □□□□□□□□□□□□ corresponds to □□□□□□□□□□□□. One of the prompts that the speaker utters is the sheet number. Thus the correspondence between □□□□ and □□□□ is found using a checksum algorithm (see subsection 2.5). This applies to SpeechDat-II and SpeechDat-Car. The checksum algorithm is embedded in the program and cannot be modified. In general, three types of correspondence are supported: direct, e.g. □□□□□□□□ or □□□□□□□□□□□, one based on a simple equation, e.g. □□□□□□□□□□□□□ and one based on the checksum algorithm.

These directory and file structures are created in two ways as in the case of language dependent features. Either by using a text editor or the incorporated utility as depicted in Figure 4. If the annotator has to work with single files s/he must define the source and destination files by browsing the local or network paths using the drive, directory and file list boxes. The “Apply” button applies the selected path to the focused text box. In case the user has to work with sets of files as in SpeechDat, s/he selects again the appropriate path with the drive and directory list boxes, applies it to the focused text box, and complete the rest of the structure using notations such as □□□□, □□ etc. In Figure 4 we can see the file and directory structures defined for SpeechDat-Car. Note the up and down bars next to the text boxes, that are used when we have simultaneous recordings from different sources as in SpeechDat-Car, where each utterance is recorded through 4 microphones in the car and through the GSM network. The file names that contain these structures are again stored in “□□□□□□□□□”. Changes take effect at once.

2.3. Waveform formats

Various formats (pcm raw data, wav, A-law, mu-law), sample sizes (8-bit, 16-bit) and sample rates are supported. Multiplexed files are also handled, e.g. in SpeechDat-Car where the four channels corresponding to the four microphones installed in the car are multiplexed. On-line de-multiplexing is done automatically during the annotation procedure. However, it is a process that can cause delays, depending on the number of files and the number of waveforms that are multiplexed in each file. These delays can be avoided by using off-line de-multiplexing before the annotation for one or more sets of files.

2.4. Transcription format

The transcriptions follow the SAM label format. SAM label files are text files where each row can be up to 80 characters long (SAM 5.10), or more than 80 (SAM 6.0), and □□□□□□□□ ended (according to the DOS format) (Senia, 1997). Rows are produced according to the main SAM paradigm:

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Where:

- □□□ is a three letter mnemonic followed by a colon “□” no spaces are allowed between them, so we can define as SAM-mnemonic the set “□□□□”;
- after the mnemonic follow all the defined items separated by commas;
- missing items are accepted and nothing needs to be put between commas to substitute them;
- spaces are not significant but good printing rules allow a better readability.

In our tool, a utility enables the user to define the mnemonics and their formats and whether the annotator will be asked to enter information about them. For example if a “□□□□” mnemonic is defined for the telephone device of the recording and marked for question, each time the annotator completes a set of files s/he will be asked for the telephone model. On the other hand a mnemonic could not be marked for question and be updated automatically, e.g. “□□□□” and “□□□□” for the beginning and ending samples of the waveform or it could be given a default value, e.g. “□□□□” for the recording place, that is “□□□□□□□□□□□□□□□□□□□□”.

2.5. Annotation procedure

In case single files are processed, the annotator opens the speech file, transcribes it and saves the transcription and other available information in a label file following the format described in subsection 2.4.

If the user wants to process a speaker’s set of files, s/he enters the speaker number □□□□ (see subsection 2.2) in the speaker selection text box. In case a prompt sheet exists, the user listens to the file where the sheet number is uttered and if this number is correct according to a checksum algorithm, the procedure is continued with the annotator’s task facilitated by the prompts. Otherwise, the speaker session is considered invalid and the annotator proceeds with another speaker. After all the speaker utterances are transcribed, the user gives the available speaker and recording information for the mnemonics marked for question (see subsection 2.4) and the final label files are created.

The annotator’s task is also enhanced by utilities that transform digits, numbers, letters, dates and times to their orthographic transcriptions, and that transform lower case to upper case characters and vice versa. In all these cases, the annotator can select the part of the text where the transformation will be applied. Otherwise, the whole text is transformed. If a transformation is not applicable, e.g. if the selected text does not contain digits and the “Digits” button is pressed, nothing happens. Different kinds of noises or other acoustic phenomena are indicated using user-defined symbols. The “Discard” button discards characters that are irrelevant to the transcription format. The “Back”, and “Clear” buttons help to cancel current changes or clear the transcription. The annotator can also mark the files with a user-defined notation, e.g. “GARBAGE” for problematic files, “OK” for correct ones etc.

The speech signal is displayed graphically to help locate speech segments and non-speech acoustic events.

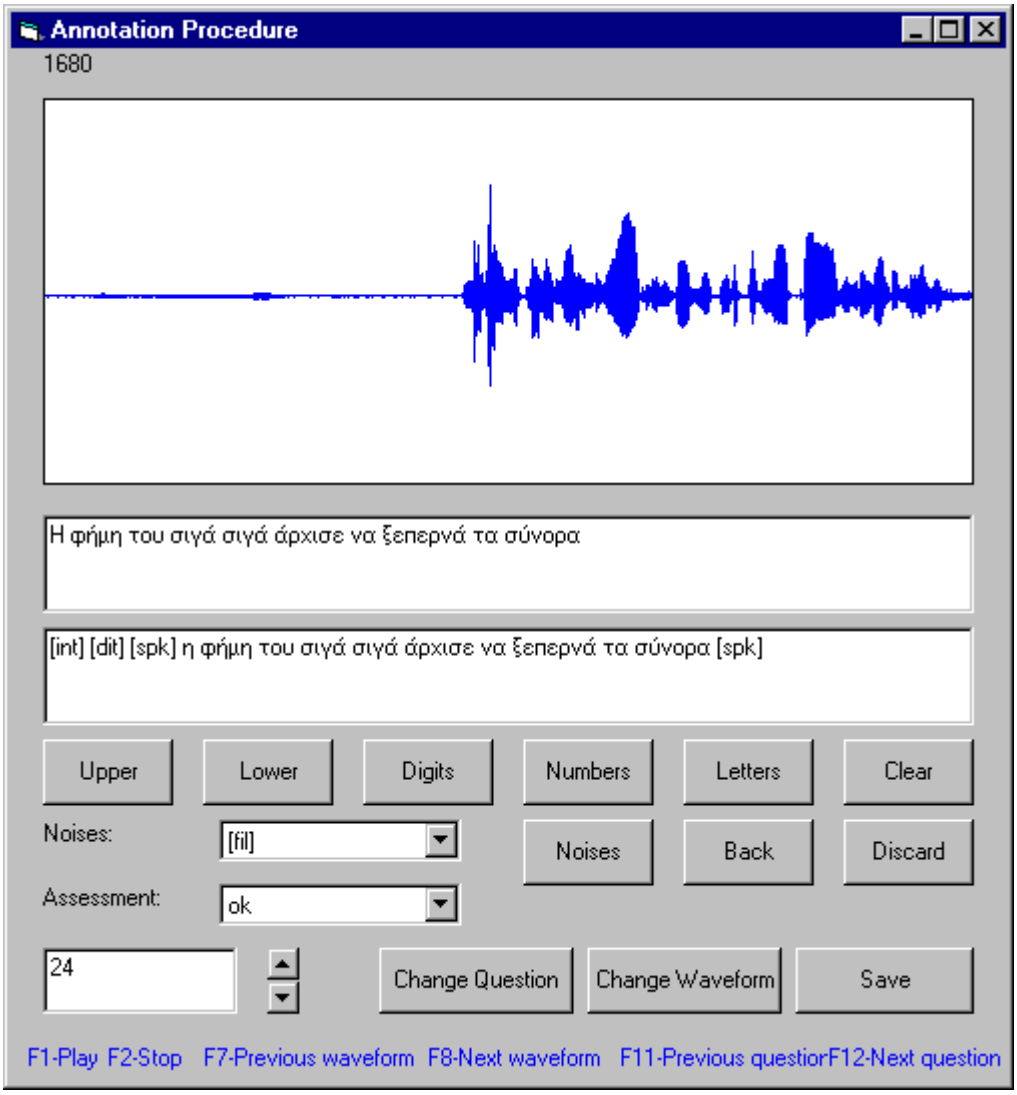

Figure 5. Central screen of the annotating procedure

In Figure 5 the central annotation screen is shown with the transcription of a SpeechDat-Car recording.

2.6. Temporary files

During de-multiplexing, the new de-multiplexed files are created. In single files' processing no temporary files are created. However, in speakers' sets of files temporary files are created so that the annotator is able to abandon the annotation procedure and continue later without having to do anything from scratch.

To save space the temporary and de-multiplexed files can be deleted when the annotation of the corresponding speech files is completed. If the annotator wants to check again a completed session, the program does not read the temporary files but extracts all the necessary information from the final ones. In addition, if it does not find the de-multiplexed files, it creates them on-line. This also applies to single files.

2.7. Utilities and statistics

Apart from the utilities that have been mentioned so far concerning the definition of linguistic features, directory and file structures etc., two additional utilities are included. One that transforms blocks of files from one format to another and one that de-multiplexes sets of multiplexed files. This is very important and saves time since the on-line de-multiplexing could be a time consuming task (see subsection 2.3). Statistical information e.g. tables, index files and distributions, formatted according to the user's needs, can be extracted from the label files. Two lexicons containing the phonetic transcriptions of words, one following SAMPA (SAMPA) and the other a format defined in our lab using a different phonetic alphabet, are created.

2.8. Interaction with other tools

The tool can get input from a speech recognizer to help the annotator, especially in cases where there is no prompt sheet. The recognition is off-line and the output of the recognizer is displayed in a label box. The user can select some files according to a criterion, e.g. discard "GARBAGE" files. A text file with two columns is created. The first is the path and the second the transcription. The speech files, the text file and the corresponding lexicon can be used by the HTK for

training a speech recognizer. It has been taken into account that the file types processed by the tool may not be supported by the HTK. In this case an automatic transformation to a valid format, e.g. wav is performed.

3. Implementation issues

The tool runs on Windows-based systems. It was developed using Microsoft's Visual Basic except of the routines, which handle speech (view, play, etc). The C++ module is loaded as a DLL in a Visual Basic project. An installation program sets up the program automatically with all the necessary files.

More than one annotator can access the speech database simultaneously. However, it is ensured that a single file or a set of files, are accessed only by one annotator to avoid conflicts. This is accomplished by using log files that indicate when a file is already in use.

The program can work with files installed on a single computer or through a local network. In the second case, network paths must be mapped to drive letters thus making the handling of local and network paths uniform.

4. Experiences

This tool has been used for the orthographic transcription of a part of the Greek SpeechDat-II database. Its predecessor, providing the same facilities but without fulfilling the demands for SpeechDat-Car and not having been language-independent and parameterized, has been used for most of the SpeechDat-II sessions. This new version is currently being used for the annotation of the SpeechDat-Car recordings. It has also been used for annotating some other speech files in order to be used for training a speech recognizer.

It is a user-friendly tool that enhances the annotator's task by providing many utilities. It does not require any special experience from the annotator. The definition of some structures and formats could be a little complicated for novice users in some cases but this is done only once. The annotators learned to use the tool very fast. They found most helpful the fact that they could see the prompts, the transformation of digits, letters, numbers, dates and times to their orthographic transcription, and the transformation of lower case to upper case characters and vice versa. The graphical representation of the signal helped them to locate speech segments and non-speech acoustic events. When they had to deal with multiplexed files, the off-line de-multiplexing utility saved significant time. Finally, the existence of temporary files allowed them to stop the transcription at any time without having to finish the current set of files. This has also proved to be very efficient in cases of network and power failures and no work was lost.

The required time for transcription depends on the quality of the speech signal, the length of the utterance and whether we handle a single file or a set of files. In SpeechDat-II each speaker session contained 56 utterances having duration of approximately 12 minutes. The transaction duration consists of the time required for the speaker selection, the time for transcription process and the time required to complete information such as age, sex, environment etc. and create the final label files. It varied from 25 to 35 minutes depending on the existence or not of noise in the signal, on whether the speaker was co-operative or not and on the experience of the annotator.

A SpeechDat-Car session contains 121 utterances. The annotators transcribe 2 versions of the same utterance, that is the recording through GSM and the one through a close-talk microphone in the car. At first the user needed about 95 minutes to complete the session. That was due to the fact s/he had to transcribe two signals for the same question and s/he had the same initial prompt as a reference for both the close-talk microphone and the GSM recordings. When we used the transcription of the close-talk microphone as a reference for the GSM transcription, this duration was significantly decreased to approximately 70 minutes since now the annotator only had to change the noises. The rest was ready from the annotation of the microphone channel recording.

5. Summary and conclusions

In this paper we described a speech annotation tool that was first developed for the transcription of the SpeechDat-II recordings and then it was extended to incorporate the additional features of the SpeechDat-Car database. However, it is not restricted to these formalisms. It has been parameterized so that it can handle various file and directory structures, waveform and transcription formats. It is also language independent since the user can define the labels and messages or the structures of numbers, dates and times etc. in the new language. The interface is graphical, user-friendly and its incorporated utilities enhance the annotator's task as it turns out from the experience of its use. In addition, it can get the output of a speech recognizer off-line as a reference and direct files to Entropic's HTK for training a speech recognizer. It has been implemented for Windows-based systems.

6. References

Hoege H., Tropf H., Winski R., van den Heuvel H., Haeb-Umbach R. and Choukri K., *European speech databases for telephone applications*. Proceedings ICASSP '97, Munich, Germany, Vol. III, pp. 1771-1774, 1997.

Van den Heuvel H., Boudy J., Comeyne R., Euler S., Moreno A. and Richard G., *The SpeechDat-Car multilingual speech databases for in-car applications: some first validation results*. Proceedings EUROSPEECH '99, Budapest, Hungary, Vol. 5, pp. 2279-2282, 1999.

Draxler C., *WWWTranscribe – A modular transcription system based on the World Wide Web*. Proceedings EUROSPEECH '97, Rhodes, Greece, Vol. 4, pp. 1691-1694, 1997.

Constantinescu A., Caloz G., Draxler C., Sanders E., Bournakas I. and Senia F., *Recommendations and Specifications of Annotation Tools.* SpeechDat Report SD3.1.1, 1997.

Chatzi I., Fakotakis N. and Kokkinakis G., *Greek speech database for creation of voice driven teleservice*. Proceedings EUROSPEECH '97, Rhodes, Greece, Vol. 4, pp. 1755-1758, 1997.

Young S., Odell J., Ollason D., Valtchev V. and Woodland P., *The HTK Book*. Entropic Cambridge Research Laboratory, March 1997.

Senia F., *Specification of speech database interchange format*. SpeechDat Report SD1.3.1, 1997.

SAM-PA, *Standards, Assessment, and Methods: Phonetic Aplhabets*. http://phon.acl.ac.uk/home/sampa/home.htm