EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

WWWTranscribe - A MODULAR TRANSCRIPTION SYSTEM BASED ON THE WORLD WIDE WEB

Christoph Draxler

IPSK – Department of Phonetics and Speech Communication
University of Munich
Schellingstr. 3, D 80799 Munich, Germany
Tel. +49/89/2866 9968, Fax +49/89/280 0362, E-mail draxler@phonetik.uni-muenchen.de

ABSTRACT

WWWTranscribe is a transcription system based on the WWW. It is platform independent and allows network access to speech databases. Its modular structure make it flexible, and it connects easily to existing signal processing applications or database management systems. WWWTranscribe consists of static HTML documents containing forms. To these forms CGI applications are attached that perform data processing and that dynamically create subsequent HTML documents.

The system has been developed for the orthographic annotation of the German SpeechDat(II) telephone speech database. In its current implementation, it automatically creates SAM-PA annotation files according to the SpeechDat(II) database specifications [5], [4]. Variants of the system are being used for transcription by other SpeechDat(II) partners.

1. INTRODUCTION

In speech data collections, the transcription of the speech signal is one of the most time-consuming tasks. It requires substantial human expertise, be it for the transcription itself or the validation and quality control of automatic transcriptions. Both hardware and human resources are scarce and thus they need to be used efficiently. Two key concepts to achieve this goal are

- hardware platform independence, and
- work group access to speech databases via network.

Tools based on the http (hypertext transfer protocol) and HTML (Hypertext Markup Language) underlying the WWW (World Wide Web) promise both platform independence and distributed access to data using a single uniform protocol.

1.1 WWW Client-Server Architecture

A WWW client (a browser, e.g. Netscape Navigator) sends a URL (Uniform Resource Locator) to a WWW server via a TCP/IP network, e.g. the Internet (Fig. 1). It receives HTML formatted documents which it then displays, or data which is output either by the browser or external helper applications.

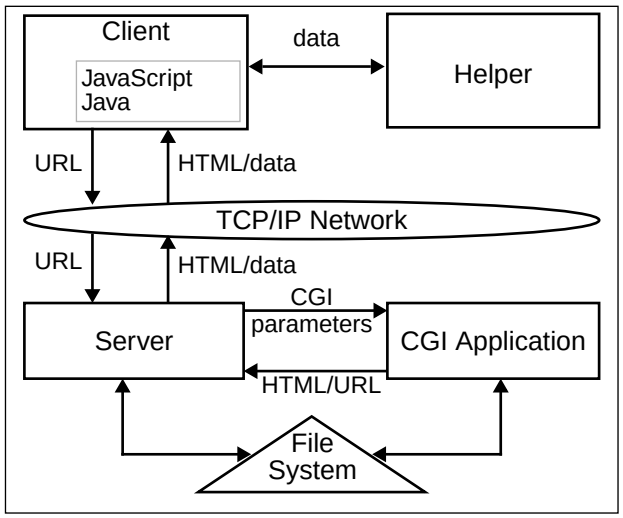

Fig. 1: WWW client-server architecture

Modern WWW browsers permit the execution of scripts or applets. Script languages, e.g. JavaScript, are restricted to accessing the objects in the currently displayed windows, e.g. form field contents, buttons, or browser properties, e.g. version number, platform, etc. Applets are usually implemented in Java, a full-blown object-oriented programming language. In general, neither scripts nor applets are allowed to access resources on the client for security reasons.

The WWW server either accesses the local file system to retrieve the requested document, or calls an external application via the CGI (Common Gateway Interface). The external CGI application may perform arbitrarily complex actions, including calls to external signal processing applications, or accessing DBMSs. It returns either a dynamically created HTML document which is passed on to the client, or a URL which again is processed by the server.

1.2 SpeechDat

SpeechDat is a European telephone speech database collection project funded by the European Union (2nd Language Engineering Programme LE2 4002-1). The aim of

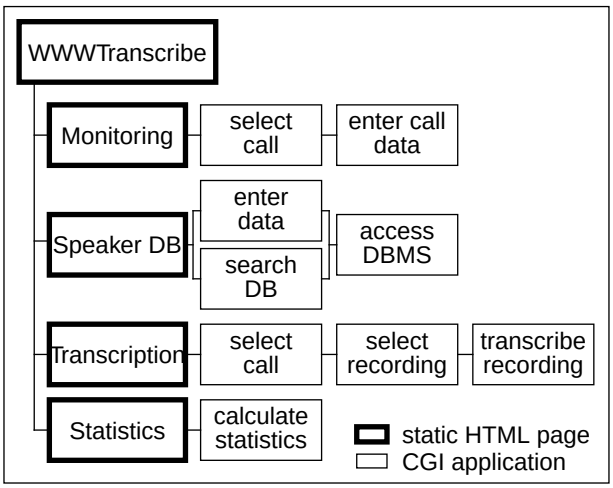

Fig. 2: WWWTranscribe architecture

the project is to provide uniform databases for the development of voice driven telephone applications in most all major and many minor European languages.

In SpeechDat(M), 1000 speakers have been recorded in 8 European languages (Danish, English, French, German, Italian, Portuguese, Spanish, Swiss French) via the fixed telephone network. In SpeechDat(II), further 10 languages are being recorded (Dutch, Finnish, Flemish, Greek, Luxemburgish French and German, Norwegian, Swedish, Swiss German, Welsh) with up to 5000 speakers and via the fixed and mobile network. For details see [6] and [7], or the project's WWW site:

`http://www.phonetik.uni-muenchen.de/SpeechDat.html`

2. WWWTranscribe

WWWTranscribe was developed for the orthographic transcription of the German SpeechDat recordings. It consists of four modules for the following tasks:

- monitoring recordings
- speaker database access
- transcription of recordings, and
- call and speaker statistics

Basically, a static HTML root page serves to initiate a task, and sequences of dynamically generated HTML pages guide the user through the task. The HTML pages contain forms which are processed by CGI applications. These applications either generate further HTML pages, or save the user input to the local file system or a DBMS.

The architecture of WWWTranscribe is shown in (Fig. 2). There is a module for each of the four tasks and a top-level root page.

2.1 Recording monitoring

In the recording monitoring, incoming calls are registered: call ID and date, sheet number used in the call, the speaker's gender, age, and dialect region are taken from the file system and selected recordings. This data is stored in a global log file.

2.2 Speaker database access

In the speaker DB module, the data sheets returned by the speakers to the Phonetics Department are entered into the global speaker database containing speaker information (name, address, speaker characteristics, type of handset used, place from which call was made, etc.). This database resides in a DBMS where it is protected from unauthorized access.

2.3 Transcription

In the transcription module, the transcriber logs in and enters the ID of the call to be transcribed. A call consists of approx. 50 recordings, each containing a single utterance corresponding to a prompt in the interview. All recordings for the current call are listed in a popup menu, and the transcriber selects one. Once a recording is selected, the transcription page is displayed (Fig. 3).

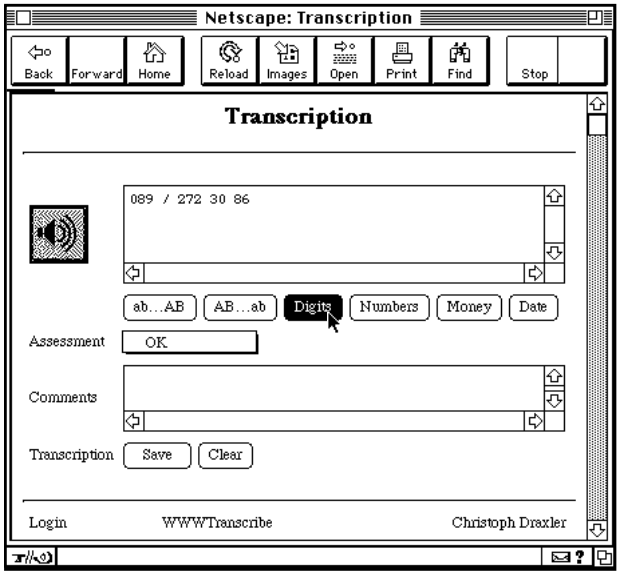

Fig. 3: Transcription window

It contains a signal output button with a speaker icon, transcription and comment text fields, an assessment menu, and save and clear buttons. A click on the speaker button outputs the speech signal auditively.

For read items, the original text of the prompt sheet is displayed in the transcription field, for spontaneous speech this field is initially empty. Any text in the transcription field can be edited.

The buttons below the transcription field perform some basic conversion tasks on the text in the transcription field:

- text to lower or upper case
- digit sequences to orthographic digit or number strings
- money amounts and date expressions to orthographic strings

The Assessment popup-menu allows the transcriber to select general noise markers.

Comments on the recording, e.g. on the quality of the speech or the signal, may be entered into the comment field.

The Save button saves the transcription to the file system at the server site in the SpeechDat SAM database exchange format [5].

2.4 Statistics

In the statistics procedure, the global log file is analyzed and key data, such as gender, age, and regional distribution are computed and displayed.

3. EXPERIENCES

WWWTranscribe is currently used for the orthographic transcription of the German SpeechDat database on Macintosh, SUN workstations, and LINUX PCs.

3.1 Access

At the Phonetics Department all computers are connected to the Internet via an Ethernet local area network. Access via modem from home is also possible. With fast modems or ISDN, even transcriptions can be performed at home; transferring a typical 16 KB recording takes 8 seconds on a 2 KB/s connection (28.8 KBd), and 2 seconds via ISDN.

Stand-alone configuration

WWWTranscribe depends on the client-server communication via http and TCP/IP. However, it can be used on computers without physical network access if both the server and the client reside on the same machine and can communicate (usually over a virtual network connection). On most platforms this will entail a performance penalty.

3.2 Interface

The interface of WWWTranscribe is intuitive and easy to use. Most transcribers had only little experience with WWW browsers or the transcription procedure as a whole, but within minutes they were familiar with the system. Some functionality not implemented by WWWTranscribe itself is made available by the browser. For example, WWWTranscribe does not allow to review the transcription of a recording; however, through the history mechanism of the browser previous transcriptions are accessible.

3.3 Transcription

The duration of an interview in the German SpeechDat recordings is approx. 9 minutes, yielding approx. 3.5 minutes of read speech, and up to 2 minutes of spontaneous speech, including one long item up to 1 minute of speech.

The transcription of read speech is straightforward because in most cases it only requires minor modifications of the original prompt text in the transcription field. These modifications include setting signal truncation and noise markers, and identifying mispronunciations or word fragments.

The short spontaneous speech items, e.g. city of call, yes/no responses, spellings of one's own first name, also do not pose any problems. In general, a recording has to be listened to twice: once to get the contents of the utterance, and then again to place the markers.

The very long spontaneous item is more difficult to transcribe because of the limited size of the transcription field where the transcription is only partially visible. Furthermore, the current audio output helper applications on the Macintosh and LINUX can only play the signal as a whole. Typically, a transcriber can memorize approx. 5 to 10 seconds of speech. As a consequence, the 1 minute item has to be repeated 10 times or more. Clearly, playing only selected portions of a signal would be useful.

The conversion buttons have proven to be a major help to the transcriber; for spontaneous items they are often used as short cuts. The transcriber enters a date in numerical format and then converts it automatically to the appropriate orthographic representation.

Transaction times

Because the annotation file is saved in its final format and location, no further handling of the files is required. The transaction time of a call thus consists only of the time required by the transcription, plus some small delay for the initial login and call selection. Typically, a complete call is transcribed in approx. 35 minutes. The transcription time depends heavily on the amount of noise in the signal, and on the duration of the long spontaneous speech item.

Transaction times were best under UNIX systems because of the efficient interprocess communication between the WWW client and the helper applications. On the Macintosh, each signal file retrieved from the WWW server is saved to disk prior to being opened by the helper application, causing a delay of a few seconds for each recording.

4. IMPLEMENTATION ISSUES

WWWTranscribe is implemented as modular and machine independent as possible. However, there are limits to the modularity and portability.

4.1 Software requirements

In a standard TCP/IP network configuration, WWWTranscribe requires a modern WWW browser, a helper application to output a-law signal files, the scripting environment perl, and TCP/IP access software. Most of the software can be obtained for free or a small shareware fee from Internet software archives, or it is part of the operating system.

4.2 Languages

In WWWTranscribe, three different languages interact:

- HTML is the document structure description language,
- perl processes the input from the forms, accesses the file system on the WWW server, and generates HTML documents
- JavaScript is used for the implementation of the conversion routines of the buttons in the transcription and for consistency checks in forms.

HTML and perl are essential for WWWTranscribe, whereas JavaScript is not. Without JavaScript the system still works, but loses some of its interface features that make it an efficient and easy to use tool.

HTML is well standardized and understood by all WWW browsers. Tags of new HTML versions which cannot be handled by older browsers are simply ignored. Note that this causes problems for tag pairs, such as `<SCRIPT>`... `</SCRIPT>`, where the tags are ignored, but the text between the tags is displayed.

perl also is well standardized. It comes with most UNIX systems and is available for Macintosh and Windows. CGI applications run on the WWW server site - hence they are under the control of the service provider who can adapt them to the requirements of the server site.

JavaScript is run within the WWW browser, and the service provider does not a priori know which version of which WWW browser is being used. Because JavaScript is only now becoming mature, older browsers do not support the latest versions.

JavaScript is also platform dependent in that its character code table is that of the machine it runs on. Windows and the Macintosh use proprietary code tables, UNIX uses ISO 8859-1, and hence any strings generated by JavaScript must be converted to a format appropriate for the current platform.

Furthermore, for security reasons JavaScript is not allowed to read or write files on the client system. As a consequence, it cannot be used for tasks that are best implemented with access to local files, e.g. dictionary lookup to check the spelling of the orthographic transcription.

Finally JavaScript versions 1.0 and 1.1 differ considerably in important data structures such as arrays and in the allowed values of the properties of the objects in a browser window. WWWTranscribe uses JavaScript version 1.1.

Language alternatives

Script languages with less restrictions, e.g. Microsoft's ActiveX, may be considered as an alternative to JavaScript. However, only JavaScript runs on any platform for which there is a Netscape Navigator or Microsoft Internet Explorer.

Java is another alternative to JavaScript. It is much more powerful than JavaScript and available for most common platforms. Furthermore, applets in Java are secure. Dictionary lookup or a graphic display of the speech signal can be implemented in Java; however, this entails increased interaction with the WWW server because only on the server can system resources be accessed.

4.3 Operating system dependencies

File system naming conventions (maximum file name length, file name delimiters, etc.) and machine dependencies of some perl functions (e.g. `stat <file_name>`) require a careful adaptation to each platform.

To facilitate this adaptation, all vital configuration data in WWWTranscribe is held in variables at the beginning of the scripts so that these platform dependencies can be handled locally.

Some operating systems have a limit for the number or the size of arguments that can be passed to an application. The GET method of passing arguments to a CGI application is subject to such restrictions (the PUT method passes the arguments via standard input and thus is not affected). Also, some WWW servers do not support all types of parameter passing for CGI applications. Although the CGI scripts in WWWTranscribe are implemented to support both the GET and the PUT method for passing arguments they might not function properly due to the restrictions of the WWW server or the underlying operating system.

5. CONCLUSION

WWWTranscribe has successfully been used in the validation of the German SpeechDat telephone speech collection. It runs, with minor modifications only, under LINUX, MacOS, and Windows - either on stand-alone machines or in a network. Extensions of the current system will include stronger ties to DBMSs, and Java applets for signal display and consistency checking.

6. REFERENCES

[1] Constantinescu, A. et al.; SpeechDat Annotation and Validation Tools, SpeechDat Report SD 3.1.1, 1997

[2] Draxler, Chr.; WWWTranscribe Installation and User Manual, http://www2.phonetik.uni-muenchen.de/speechdat/WWWTranscribe.html

[3] Höge, H. et al.; European Speech Databases for Telephone Applications, ICASSP 97, Munich

[4] SAM-PA; Standards, Assessment, and Methods: Phonetic Alphabets, http://phon.ucl.ac.uk/home/sampa/home.htm

[5] Senia, F.; Specification of speech database interchange format, SpeechDat Report SD 1.3.1, 1997

[6] Senia, F. et al.; Definition of corpus, scripts and standards for Fixed Networks, SpeechDat Report SD 1.1.1, 1997

[7] van Velden, J., Langmann, D.; Pawlewski, M.; Specification of Speech Data Collection over Mobile Networks, SpeechDat Report SD 1.1.2/1.2.2, 1997