Transcriber: development and use of a tool for assisting speech corpora production [1]

Claude Barras [a,2], Edouard Geoffrois [b], Zhibiao Wu [c] and Mark Liberman [c]

[a] *Spoken Language Processing Group, LIMSI-CNRS, BP 133, 91403 Orsay cedex, France*

[b] *DGA/CTA/GIP, 16 bis av. Prieur de la Côte d'Or, 94114 Arcueil cedex, France*

[c] *LDC, 3615 Market Street, Suite 200, Philadelphia, PA, 19104-2608, USA*

Abstract

We present "Transcriber", a tool for assisting in the creation of speech corpora, and describe some aspects of its development and use. Transcriber was designed for the manual segmentation and transcription of long duration broadcast news recordings, including annotation of speech turns, topics and acoustic conditions. It is highly portable, relying on the scripting language Tcl/Tk with extensions such as Snack for advanced audio functions and tcLex for lexical analysis, and has been tested on various Unix systems and Windows. The data format follows the XML standard with Unicode support for multilingual transcriptions. Distributed as free software in order to encourage the production of corpora, ease their sharing, increase user feedback and motivate software contributions, Transcriber has been in use for over a year in several countries. As a result of this collective experience, new requirements arose to support additional data formats, video control, and a better management of conversational speech. Using the annotation graphs framework recently formalized, adaptation of the tool towards new tasks and support of different data formats will become easier.

Key words: transcription tool, speech corpora, broadcast news, linguistic annotation formats

---

[1] This work was done while the first author was working with DGA.

[2] Corresponding author. Tel.: +33-1 69 85 80 61; fax: +33-1 69 85 80 88; e-mail: Claude.Barras@limsi.fr

1 Introduction

Speech research has long been conducted using small- or medium-sized databases recorded in controlled conditions. Until a few years ago, they often consisted of short duration recordings, and the speech was read by or elicited from a well-identified speaker. For read speech, orthographic transcription was not much of a problem since the content was known in advance. The need to transcribe appeared with spontaneous speech, but for short duration recordings made in a controlled environment transcription was easy and a classical text editor associated with a simple sound player was generally enough.

With the advent of work on long duration recordings of uncontrolled speech, the situation has changed. Navigation in a long duration recording becomes an issue, as well as time-alignment of the annotations with the signal. Additional information like background conditions, speaker turns or overlapping speech should be indicated along with the orthographic transcription. Further annotations can be needed by new research areas like named entities or topic detection. Therefore, new tools are required. Furthermore, for large quantities of data, productivity becomes a concern and can be increased by ergonomic tools.

In the framework of the Defense Advanced Research Project Agency (DARPA) programs, the Linguistic Data Consortium (LDC) has produced several hundreds hours of manually transcribed Broadcast News data, and developped specific tools and internal know-how for this production. There is now a growing need for producing similar data in other places. For instance, a project for transcription and indexing of multilingual Broadcast News started at the French Délégation Générale pour l'Armement (DGA) in 1997. A software environment was needed for creating the necessary corpora. After examination of existing solutions, it appeared that no available transcription software completely filled the needs, and it was decided to develop a new tool. The development of "Transcriber" started at the DGA in coordination with the LDC in late 1997, and the first release was presented in May 1998 (Barras, Geoffrois, Wu and Liberman, 1998). Since then, development went on and new features have been added according to the needs, until reaching a stable state. Besides, the experience acquired while using the tool and the desire to address new tasks have raised more scientific issues related to the format and the structure of the annotations. This article describes the current status of the tool, the experience gained and some future directions.

In the next section, we present the major requirements identified for the tool and explain why existing annotation tools could not fulfil our needs. Section 3 describes the main features of Transcriber, the format of the transcriptions, and explains the main implementation choices. Some experience of using the tool is presented in Section 4. Future directions and format evolution are discussed in Section 5.

2 Motivations

2.1 Data characteristics

A tool for the manual transcription of large amounts of radio and television soundtrack recordings was needed in order to create corpora and develop automatic speech recognition systems for indexing and retrieval of Broadcast News in several languages. The DARPA Broadcast News transcription task started in 1995 with the first formal evaluation campaign in 1996 (Stern, 1997), and a project on the same task started at DGA in 1997.

The Broadcast News task was the first wide-scale effort to address speech which has not been produced specifically for research purposes. Recordings can have durations from several minutes to several hours. Annotations have to provide the following information:

- an *orthographic transcription* along with a precise description of all audible acoustic events, including hesitations, repetitions, vocal non-speech events and external noises;
- a division into speech *turns*, with an identification of the speaker for each turn;
- a division into larger *sections*, such as "stories", including a clear separation of advertising and news sections;
- indication of variations in transmission channel or acoustic *background conditions*.

Turns, section boundaries and changes of acoustic conditions have to be temporally localized. The orthographic transcription also needs to be precisely and frequently synchronized with the speech signal (breakpoints can be located at pauses, breaths, sentences or any other convenient places), thus defining shorter segments. There are frequent portions of overlapping speech in spontaneous dialogs which need to be addressed. All these features imply some specific requirements for the annotation tool.

2.2 Requirements

The main requirement is to allow the user to manage long duration signals and input the various annotations described in the previous section as efficiently as possible. We also wanted a tool which can be easily installed and used.

2.2.1 User interface

Transcribing audio or video recordings is a very time-consuming task. It is usually done by educated native speakers of the language with no specific skill in computer science. Therefore, a transcription tool should mimic as much as possible the user interfaces of standard office software, so as to reduce training time. Its use should be intuitive, in order to lower the cognitive load and decrease error rates. In particular, it must provide an easy and intuitive association between the time course of the speech signal and the textual representation of the transcription and other annotations. Users should find it easy to navigate within either the audio stream or the textual transcription. Navigation and modification in either domain should automatically translate into appropriate changes in the other domain, and the methods for creating links between text and time must be easy and intuitive. In addition, fast response is crucial. Indeed, regardless of the interface design, a tool will not be accepted by users unless it responds quickly to user actions (McCandless, 1998).

Two features deserved special attention. First, in order to help navigation into the signal and segmentation, a cursor on the waveform should show the current position in the signal even while listening, ie. the cursor should move in synchronization during playback. This feature is not straightforward to implement in a portable way. Second, the user should not experience any delay when navigating in long duration signals, ie. displaying of such signals, including scrolling and zooming, should be very fast and reactive. This feature requires specific optimizations.

2.2.2 Multilingual transcriptions

In the framework of a multilingual indexing project, support for multiple languages is needed. Several aspects are involved: keyboard input, character display with specific issues on bi-directional scripts (for languages like Arabic), and internal data encoding with adequate file input/output. The localization of the interface is also useful, though less critical.

2.2.3 Easy deployment

We wanted a tool that would work on inexpensive computers, in order to reduce the cost per workstation. This implies that the interface should remain reactive even with limited computing power. More generally, we wanted a portable tool which could be easily installed on already existing computers and environments, and in particular which works on most Unix systems and on Windows.

To further ease deployment, the tool should not be encumbered by proprietary licence issues, both for ourselves and for potential partners. Of course, using free software also reduces the per-user cost.

2.3 Existing annotation tools

We first considered using existing transcription tools. One of the most well-known tools for signal analysis is Entropic's product ESPS/*waves+* (formerly known as Xwaves) which efficiently manages signal and spectrogram displays and allows the user to edit a segmentation of the signal (e.g. at the phonetic level or at word level). However, it is not adapted to the transcription of broadcast news or of spontaneous conversations. For the transcription of multilingual telephone conversations and broadcast news recordings in the framework of the DARPA programs, the LDC developed a tool based on an interface between *waves+* and the Emacs text editor. The resulting tool runs on Unix workstations, and requires a significant amount of training and supervision, since users must learn basic Unix skills, basic Emacs skills, and basic *waves+* skills. The Entropic "annotator" product has similar characteristics. These solutions were unsatisfactory because of the issues of user training and supervision, and hardware and software expense [3] .

Another, independent, annotation tool (named TNG) was developed at the LDC in Java a few years ago. However, compared with *waves+*, the waveform display and its update in response to user requests were relatively sluggish, so that it required a high-end workstation to be usable. It could not display a moving cursor during playback, and the first version of Java could only support 8-bit mu-law audio. Furthermore, the status and the licensing policy of Java and of some libraries needed for the user interface or audio management remained unclear for a long period. This direction was thus not pursued.

Many speech research laboratories have developed software for their own needs and some of them have released these tools publicly (with varying licensing schemes). First versions of the OGI CSLU Toolkit (Schalkwyk, de Villiers, van Vuuren and Vermeulen, 1997) included Lyre, a signal viewer with some segmentation capabilities. SFS tools from University College London (Huckvale, 1987-1998) are a set of powerful programs for speech processing, including display, but not designed for interactive user interfaces. The Spoken Language

[3] In addition, following the acquisition of Entropic by Microsoft, its product line of speech tools has been terminated, so that future availability of software relying on *waves+* is compromised.

Systems Group from MIT has described the architecture of their speech analysis and recognition tool SAPPHIRE (Hetherington and McCandless, 1996), which includes graphical tools; the design of SAPPHIRE seems promising but the tool is not publicly available. The EMU Speech Database System from Macquarie University (Sydney) is a collection of software tools for developing and extracting data from speech databases, including the creation of hierarchical and sequential labels of speech utterances (Cassidy and Harrington, 2000). The CHILDES system developed at Carnegie Mellon University provides tools for studying conversational interactions and for linking transcripts to digitized audio and video (MacWhinney, 2000), and large databases are available in the associated CHAT coding.

Since this first overview in late 1997, new tools appeared. The problem of synchronization between ethnographic speech data and related annotations has been addressed by the LACITO Archive project (Jacobson, Michailovsky and Lowe, 2000); the tool SoundIndex, initially written for the Macintosh platform, is used for time-alignement. The Institute for Signal and Information Processing (ISIP, Mississippi State University) provides several public domain software in the field of speech recognition and signal analysis, and the same group released Segmenter, a graphical tool to aid in performing segmentation and transcription of two-channel telephone speech data (Deshmukh, Ganapathiraju, Gleeson, Hamaker and Picone, 1998); recent versions are also available for the Broadcast News task. The tool TransEdit has been developed at Carnegie Mellon University for the Windows platform (Burger, 1999). It was designed following speech annotators' requests with flexibility and multimedia support in mind, resulting in very user-friendly tool. A more complete survey of existing annotation tools is available online (Bird and Liberman, 1999-2000). Some of them have also been evaluated in the framework of the EC-funded MATE project, which started on March 1998, and aims to develop a standard for spoken dialogue corpus annotation, and a related set of tools (McKelvie, Isard, Mengel, Moller, Grosse and Klein, 2000).

To summarize, a wide range of tools exist, but no solution adapted to the needs was available at the time of our choice. In particular, no one provided a really interactive management of long durations signals synchronized with the transcription. We therefore considered adapting existing tools. Solutions relying on commercial products or on software covered by restrictive licences could not be easily modified nor redistributed. Among the freely available tools, some had interesting features, but were not designed for Broadcast News transcription. We tried to reuse components of existing tools, but it proved to be a difficult software re-engineering problem, and it soon appeared that it would be more efficient to start the development of a new tool.

2.4 Development and distribution of Transcriber

Development of Transcriber began in late 1997. In May 1998, a first release was made publicly available, presented at the LREC conference (Barras, Geoffrois, Wu and Liberman, 1998) and put into daily usage at DGA. We chose to distribute the tool as free software, under the GNU general public license (Free Software Foundation, 1991). We mainly wanted to ease the production of speech corpora and encourage their sharing. We also believe in the efficiency of open source for software development (Stallman, 1998). Having developed a new tool, the additional cost of distributing it and maintaining a Web site is modest, and we expected an increase in user feedback and contributions from external developers.

Transcriber is now used in many places (at the time of writing, more than 60 persons from 17 countries have subscribed to the announcement mailing list), and we regularly receive valuable feedback from users. Since the first release, many new features have been implemented, portability and robustness have been improved, and the data format has been enriched, while always maintaining backward compatibility. The tool has reached a stable state, which we now describe.

3 Description of Transcriber

This section describes the user interface, with emphasis on the features relevant to the structure of speech annotations and specific to Transcriber, then presents the data formats, and explains some implementation choices.

3.1 User interface

The user interface of the tool is comprised of two main parts (cf. Figure 1): a text editor in the upper half of the screen, and a signal viewer in the lower half of the screen, along with the temporal segmentation at the different levels. In between, a maskable button bar provides tape-recorder-like icons for signal playback and shows the name of the files currently being edited.

The interface appearance (fonts, colors, localization) and behaviour (keyboard shortcuts, playback mode, etc.) are user-configurable. These configuration options can be saved. The file that the user is working on and the cursor positions can also be saved so that the session configuration is automatically restored when restarting the tool. Users can thus resume their work as if they had not

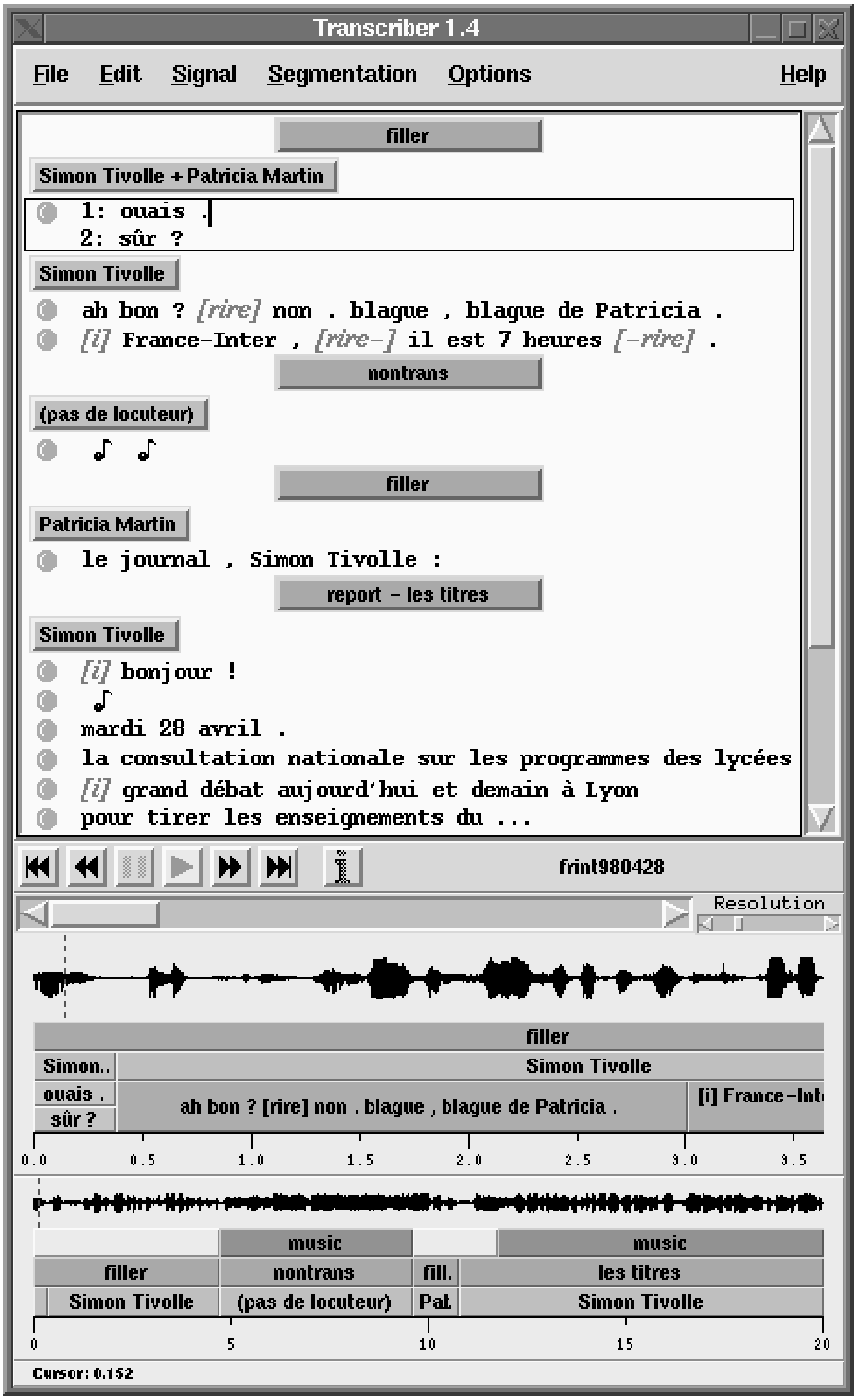

Figure 1. Screen shot of the user interface.

exited from the tool.

3.1.1 Text editor

The text editor allows for creating, displaying, and editing the transcription. A transcription consists of plain text and various markers. Standard features of a text editor are provided: cut/copy/paste of the selection, find and replace, spell checking, and a limited undo. Markers are created using the menus or keyboard shortcuts and can be edited by clicking on them to pop up a dialog window.

Two types of markers can be distinguished. Some are used for structuring: The transcription is divided into segments, which are grouped into turns which are themselves grouped into sections, and change in acoustic background conditions can appear at any point in time (cf. 2.1). These markers bear time-stamps, which correspond to the boundaries in the segmentation displayed under the signal in the lower half of the screen. They are displayed in the text editor in different ways depending on their type (cf. Figure 1):

- a new section is indicated by a button in the middle of a line with the topic name;
- a new speech turn is indicated by a button at the left of a line with the speaker name;
- the beginning of a segment in the orthographic transcription is indicated by a large dot to the left of a line; the text in the following paragraph belongs to that segment;
- a change in acoustic conditions is indicated by a music icon inside the text.

Turns and sections have attributes, which can be edited by clicking on the button. The speaker associated to the turn can be chosen from a list of all existing speakers, or a new speaker can be created. Speakers' identities can be searched for in the transcriptions, and can also be imported from other transcriptions. A specific mechanism is provided for the annotation and transcription of overlapping speech involving two speakers. Similar functions are provided for the topics associated to the sections. Background conditions (appearance or disappearance of background conversations, music, electric noise or any other kind of noise) can also be edited by clicking on the icon.

Other markers can be inserted in the text for any non-speech event, short noise, lexical annotation, language change or free comment. An open list of predefined descriptions for each kind of event is proposed to the transcriber. The event descriptions are task-specific but can be modified. These markers bear a flag indicating the extent of the marker in the text. Some events do not extend over other words, e.g. most of the speakers' vocal non-speech sounds. By default they are displayed between square brackets, e.g. `[i]` for an inspiration. Other events do, e.g. external noises which often overlap with speech or language changes. By default they are displayed in a slightly different way, e.g. `[n-] ... [-n]` for a beginning and end of a generic noise. These markers do not bear a temporal synchronization in the current implementation, but could do in the future.

3.1.2 Signal display and playback

The signal is displayed under the text editor. The signal waveform can be interactively scrolled and zoomed, even during playback. A portion of the signal can be selected for zooming or restricting playback to the selected region. Two views of the signal at different scales can be simultaneously displayed, which is useful for having a global view of the context in addition to a more precise, local view. When the audio file contains several channels, the waveforms are displayed in parallel.

Playback is controlled by tape-recorder-like buttons or by keyboard shortcuts. Various playback modes are provided, to suit the different stages of the transcription: continuous playback is useful for segmenting the signal, playback of the current segment for transcribing it, or continuous playback with a short pause at each segment boundary for verification. During playback, the cursor in the signal moves continuously in synchrony with the sound. This allows the user to associate the location on the waveform to what they hear and eases signal segmentation.

All functions remain available during playback. The user can thus annotate continuously. As playback can be controled by keyboard shortcuts, he can also almost always keep the focus in the text editor. One exception is for moving a boundary, which requires mouse dragging in the segmentation display in the lower half of the screen.

3.1.3 Signal segmentation

The temporal segmentations at the different levels (orthographic transcription, speech turn, topic change, acoustic conditions) are drawn under the signal and are synchronized with it during scrolling or zooming operations. The information associated to each segment is displayed entirely or partially according to the zoom level. Each segmentation level in each view can be independently masked at user option.

The segment boundaries can be edited by dragging them with the mouse. A new boundary can be inserted at the current cursor position using the menu or a keyboard shortcut (by default the return key, as a new line is created in the text editor). Since this is possible during playback, a rough segmentation can be quickly created by hitting a key at desired segmentation points while listening. A more precise positioning of the boundaries can be achieved in the second phase using the mouse to drag them to the correct positions.

A new speech turn or section can be inserted at any previously created boundary. Changes in acoustic background conditions can be inserted at any position, using specific commands. When a boundary is shared across levels, dragging it at one level automatically moves it at the other levels. Sequentiality of the time marks is always ensured. A boundary normally cannot be moved past its neighbors, but can be forced to move further and push its neighbors accordingly.

3.1.4 Synchronization between text and signal

The text editor and the temporal segmentation under the signal can be considered as two different views of the same transcription object. Any change in the text editor is immediately displayed in the temporal segmentation. Two cursors are simultaneously active, one in the text editor (where text can be inserted in the transcription) and one in the signal viewer (where playback will start). Both cursors are synchronized and constrained to be always consistent, i.e., they have to always stay within the same temporal segment: as soon as one cursor moves to another segment, the other cursor automatically moves to the same segment, and the windows are automatically scrolled when needed. The current segment is highlighted both in the text editor and in the signal segmentation display. During playback, the text of the segment being currently played can thus be easily followed in the text editor. If the cursor is moved to another segment while listening, playback is interrupted and restarts at the beginning of the new segment.

3.2 Data format

The set of annotations includes not only the orthographic transcription, but also all the information about turns, speakers, sections, acoustic condition changes, and other events. These data need to be stored in a file, processed in various ways, and exchanged easily. The data format thus needs to be chosen carefully. It should as far as possible follow existing standards, or at least be easily converted with some of them.

3.2.1 File format

Obviously, Unicode which is the most standard multilingual character encoding (The Unicode Consortium, 2000) should be supported. Unicode provides a unique encoding for every character in almost all existing languages and thus allows texts in several languages to appear within a single document.

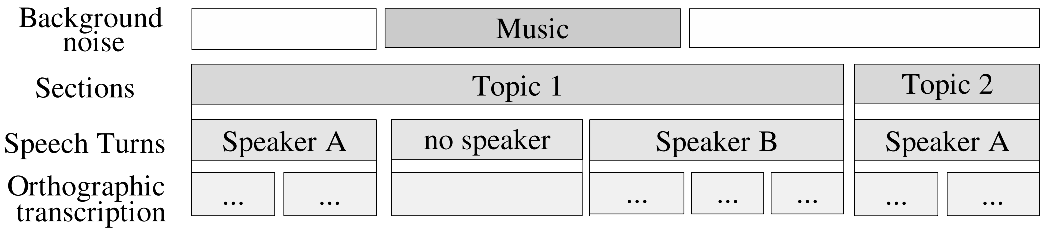

Figure 2. The 4 segmentation levels of a transcription.

Besides, transcriptions are complex objects, and a structured machine-readable format is needed. We considered SGML (Standard Generalized Markup Language) and its more recent subset XML (Extensible Markup Language) (Bray, Paoli and Sperberg-McQueen, 1998). Both allow a document to be structured as a tree. Each node of the tree contains a set of attributes with a value. The syntax used in the document can be specified in a Document Type Declaration (DTD). Tools exist for ensuring automatically the well-formedness and validity of a document, that is, that it correctly follows the SGML or XML syntax as well as its specific DTD. More importantly, SGML and XML are widespread standards, which helps sharing documents. In addition, they support Unicode character codes. Automatic processing of XML documents is much easier than SGML, and thus XML was adopted.

3.2.2 DTD design

The format was designed as being backward compatible with a previous format used at the LDC for the DARPA Broadcast News evaluations. The transcriptions have three hierarchically embedded layers of segmentation (orthographic transcription, speaker turns, sections), plus a fourth level of segmentation (acoustic background conditions) which is independent of the other three (cf. Figure 2). A global list of speakers along with their attributes is also managed inside a transcription, as is a list of topics. Figure 3 shows a manually indented sample of a transcription file corresponding to the screen shot of Figure 1.

In our case, the validation of a document is not enough to ensure its logical consistency; indeed, some properties — e.g. the fact that the “startTime” and “endTime” attributes must bear numerical values which are in increasing order, or that each of the four types of segmentation is constrained to be a partition of the whole signal — exceeds the capabilities of a DTD and have to be verified afterwards in the application. Some of these issues could be addressed using CSS (Cascading Style Sheets) and XSL (Extensible Stylesheet Language) which aim to provide more complex manipulations of XML files (Clark, 1999).

The default event description provided with the tool is currently specific to the task and to the transcriber’s language. Agreement could be reached on an

Header

```
<?xml version="1.0" encoding="ISO−8859−1"?>
<!DOCTYPE Trans SYSTEM "trans−13.dtd">
<Trans version="1" version_date="981211"
   audio_filename="frint980428" scribe="YM" xml:lang="fr">
```

List of topics

```
 <Topics>
  <Topic id="to1" desc="les titres"/>
 </Topics>
```

List of speakers

```
 <Speakers>
  <Speaker id="sp1" name="Simon Tivolle" type="male"/>
  <Speaker id="sp2" name="Patricia Martin" type="female"/>
 </Speakers>
```

Transcription

```
 <Episode program="France Inter" air_date="980428:0700">
 (...)
  <Section type="filler" startTime="9.609" endTime="10.790">
   <Turn speaker="sp2" startTime="9.609" endTime="10.790">
    <Sync time="9.609"/>
    le journal ,Simon Tivolle :
   </Turn>
  </Section>
  <Section type="report" topic="to1"
          startTime="10.790" endTime="20.000">
   <Turn speaker="sp1" startTime="10.790" endTime="20.000">
    <Sync time="10.790"/>
    <Event desc="i"/>
    bonjour !
    <Sync time="11.781"/>
    <Background time="11.781" type="music" level="high"/>
    <Sync time="12.237"/>
    mardi 28 avril .
    <Sync time="13.344"/>
    la consultation nationale sur les programmes des lycées :
    <Sync time="16.236"/>
    <Event desc="i"/>
    grand débat aujourd'hui et demain à Lyon ...
   </Turn>
  </Section>
(...)
 </Episode>
</Trans>
```

Figure 3. Sample of a transcription file.

international set of non-speech events or other annotations. This would ease the international exchange of produced corpora. However, deciding which annotations are language-independent is not straightforward, and the transcriber should remain able to add his or her own annotations.

In 1998, NIST designed an Universal Transcription Format or UTF based on previous LDC formats for the production of Hub-4 Broadcast News and Hub-5 Conversational speech corpora (NIST, 1998). Conversions between our format and UTF are partially lossy in both directions because of slightly different orientations (our format supports improved speaker characteristics but not yet the named entities optionally present in UTF). A version of Transcriber has been produced that can read, edit and write transcripts in the CHILDES format (MacWhinney, 2000). This involves a very different DTD, expressing a different (and considerably more elaborate) set of annotation categories. We aim to address the problem of making it easy to adapt Transcriber for use with a nearly unlimited variety of different annotation frameworks.

3.3 Implementation issues

This section presents the main development choices which were made, in line with the requirements.

3.3.1 Programming language and development mode

We were confronted with the choice of a language for the development. Over the last few years, there has been a growing interest in various scripting languages (Ousterhout, 1998). One of the most open and successful ones is Tcl/Tk. It is a multi-platform script language available for several Unix systems, Macintosh and Windows (Ousterhout, 1994). The syntax of the Tcl language is rather simple, but a complex user interface can be written in a few lines using the Tk graphical library. The absence of compilation significantly speeds up the development process, and computers have become powerful enough nowadays to provide rapid reactions even with interpreted applications. The need for a C or C++ development is reduced to the critical or system-dependent parts which can easily be interfaced with the Tcl script. Tcl/Tk was therefore chosen for the development of Transcriber. At the time of writing, Transcriber runs under several Unix systems (Linux, Solaris, SGI) and Windows, and a port to the Macintosh is under way.

Combined with the free distribution, the use of a scripting language allowed rapid prototyping development with quick user feedback on the tool. Numerous functions were modified or added according to user requests. For example, management of overlapping speech was changed several times in order to provide a more intuitive user interface. This development mode lasted over a year with monthly updates.

3.3.2 Multilingual text editor

The standard Tk text widget was chosen for editing the transcription. Multilingual transcriptions are possible, since recent Tk versions manage the display of Unicode characters. We also considered the Emacs text editor, which is a free, powerful text editor and supports multi-linguality; however it would have become harder to provide an integrated tool with a consistent user interface.

Unicode characters are managed internally in Tcl, and can be easily re-mapped to various alternative encodings. However, we have not experimented widely with non-roman scripts. The main limitation on script choice at present is the Tk text widget, which cannot yet handle bi-directional text or general rendering of composite Unicode characters (e.g. with diacritics). However, we hope that these capabilities will be added, since multilinguality and Unicode support are high on the list of priorities for the developers of Tcl/Tk.

No generic architecture for input methods is now available in Tcl/Tk. Keyboard configuration can often be handled at the operating system level; but if needed, it is easy to configure the tool to bind any keyboard combination to a given Unicode character.

3.3.3 Interactive display of long duration waveforms

Since providing interactive display and playback of long duration signals was a high priority, scrolling and zooming of the waveform had to be achieved without freezing the interface, even on a low-cost computer.

A specific waveform display module has been developed for Transcriber. This time-critical part is written in C, and is optimized for interactive zooming and scrolling the sound files without interrupting real-time output. The sound file is never loaded in memory, since a single hour of signal could easily exceed the available memory. The first time a long sound file is accessed, a low resolution temporal envelope of the waveform (minimal and maximal sample values for each 10 ms segment) can optionally be computed and stored on disk in order to speed up later display. In this case the display is computed using only the pre-computed envelope instead of the much bigger sound file. If the pre-computation of the envelope is disabled, the low-resolution display is disabled as well to avoid any sluggish display. During scrolling, only the required part of the waveform is computed, not the whole display. Signal segmentation display has also been designed for efficiency. All these optimizations dramatically increase the interactivity of zooming and scrolling.

As an option, remote sound file access is provided through a server controlled with sockets and specifically optimized for the tool, thus being more efficient than a standard network file access. For signal display, the waveform is computed on the server and is transmitted over the network instead of accessing the whole signal through the network. This feature makes it possible to centralize all recordings on a server, allowing interactive remote access without duplication of resources. This feature is mainly intended for the consultation of remote archives.

3.3.4 Audio management with Snack

Synchronization of the cursor during playback usually requires low-level access to the audio driver, which can limit portability. Much time was spent during early development for a reliable sound control, especially because of hardware or of low-level operating system problems. The Snack audio extension provided a good solution to these multi-platform audio difficulties.

Snack is an extension for the Tcl/Tk scripting language which provides multi-platform audio management. It was developed by K. Sjölander at KTH speech laboratory (Sjölander, 1997-2000; Sjölander, Beskow, Gustafson, Lewin, Carlson and Granström, 1998). Most commonly used sound file formats are supported, playback is efficiently supported for Windows and several Unix systems including Linux, and it runs in the background while staying under the control of the application. These excellent technical characteristics and the fact that it is distributed as free software made Snack obviously the best choice for multi-platform audio management. It was thus chosen for use within Transcriber.

3.3.5 Implementation of the parser

An XML parser was needed to make the interface between the application and the data, ensuring that any well-formed XML file will be correctly read or written. Furthermore, production of valid documents according to their DTD is important for their automatic exploitation, and we therefore needed a validating parser. At the time of development, no free validating XML parser was available for Tcl/Tk. A parser was therefore designed using tcLex, a lexical analyzer generator extension to Tcl and distributed as free software (Bonnet, 1998-1999). Unicode encoding is supported and automatically detected upon reading. The internal representation of the transcription was chosen to consist mainly in the XML data structure, which as a result is always kept in memory and dynamically updated according to transcription modifications. Saving the transcription only requires a dump of the existing data. When a DTD is active, each modification of the XML data structure in memory is immediately validated, which ensures that saving the current XML image to a file will produce a valid XML file.

4 Experience

Transcriber has been used for the DGA project on Broadcast News for over a year. It has also been used by the French company VECSYS for several months in the framework of the European Language Engineering project OLIVE (de Jong, Gauvain, Hiemstra and Netter, 2000). In this section, we report on the practical use in these two places, and on some of the experience gained. We describe the material which was transcribed, the working conditions and the productivity, and the transcription guidelines which were provided to the transcribers.

4.1 Material transcribed

The reference material for the DGA project consists of 20 hours of morning news program (7h-9h) recorded in December 1998 (10 weekdays from 2 consecutive weeks) from the national French radio station "France-Inter". This choice was motivated by the fact that the distribution rights for this data could be obtained, and by the news-oriented but varied content. The typical 2-hour program contains 3 news bulletins (for a total of about 50 minutes), specialized news (20 min.), various chronicles (10 min.), review of the French press and of the European press (15 min.), interviews and live questions from listeners (20 min.), and weather reports (5 min.). The review of the European press was done by a non-native speaker, and contained, of course, a lot of foreign names and expressions.

The material transcribed by VECSYS included 15 hours of radio recordings from French programs "France-Inter" and "France-Info", and 65 hours of television soundtracks from various channels in French and German (23 hours of "Arte" programs in French, 30 hours of "Arte" programs in German, and 12 hours of French channels "France 3", "France 2" or "TF1"). "Arte" programs consisted mostly of news bulletins and documentaries on social or political issues.

4.2 Working conditions

Two half-time transcribers were hired for the DGA project. They were educated, native French speakers. Both were given a PC (Pentium Pro 200 MHz) under Linux with headphones and loud-speakers. Each one had to transcribe a set of 10 one-hour sound files copied to their hard disks. They worked in the same room and could share their experiences. They had dictionaries and lists of journalists' names at their disposal. They went to great lengths to find the correct spelling of proper names, despite the fact that a specific marking was available for uncertain orthography. They were informed in advance of the recording sessions that they would have to transcribe, and decided to get newspapers from the corresponding days. The European press review proved to be a difficult challenge, , since foreign newspapers were more difficult to get. When they had completed a one-hour sound file, an additional verification was done in the presence of a speech researcher in order to discuss the specific problems which arose. Further checking and normalizations were performed on the whole set of transcriptions, and the transcribers had feedback about the errors.

Eight half-time native speakers of French and German produced the transcriptions for VECSYS. They started with a 15 day training period in the company, and then they were provided with a PC running Linux, a modem and the sound files on a CD-ROM and worked at home. They were also given lists of journalist names, and paper drafts when available for the Arte programs; otherwise they relied on their own resources — for example, some did name spell checking via the internet. The produced transcriptions were sent to the company by e-mail. They were verified and corrected by a person specializing in this task, and who had use of all the necessary dictionaries.

4.3 Productivity

A monitoring function was added to the tool in order to be able to analyze the production of transcriptions and estimate the amount of work needed for the transcription of one hour of material. This was also a user's request, since they were interested in monitoring their own daily progress. Time spent using the tool was measured and recorded, along with various measures of the transcription task (number of temporal breakpoints, of speech turns, of words...).

The total time needed for the production of one hour of transcribed material, including careful verification of the transcription, amounted to around 50 hours for both DGA transcribers. Of interest is that they did not follow the same strategy: the first one chose to segment and annotate the whole signal first, performing the orthographic transcription in a second pass; the second one did segmentation, annotation and transcription in parallel. The superiority of one strategy over the other one could not be demonstrated. However, getting accurate segmentations took a lot of time. This was an indication that a good automatic segmentation of the signal into short segments might speed up the overall transcription work. We have therefore given the transcribers an automatically computed pre-segmentation into breath groups produced by the LIMSI speech partitioning system (Gauvain, Lamel and Adda, 1998), which they could modify as necessary, and they found it useful. Indication of speaker changes were also provided, but the transcribers found them more confusing than helpful. These are subjective appreciations from the transcribers, and further investigation is necessary before drawing conclusions.

Mean transcription time for the VECSYS experience also amounted to around 50 times real time, with a large disparity depending on the program. Radio news programs were easier, and television debates were much harder due to frequent overlapping speech and the difficulty of speaker identification from the soundtrack only.

4.4 Transcription guidelines

Transcribers were provided with a written document describing the transcription guidelines, i.e. explanations about what should be annotated and how to annotate it. Initial guidelines were written by LIMSI. They were intentionally kept simple (and thus predictably incomplete) in their first version, and were augmented as necessary when specific questions arose.

The transcription guidelines covered the following topics:

- What should be annotated : orthographic transcription of the foreground; non-speech events and background noise conditions; speech turns with a precise identification of the speaker (name, gender, accent in the case of foreign speakers) and topics.
- What should not be annotated, such as transcription of commercials.
- How to add punctuation to increase readability without interfering with automatic processing.
- How to deal with numbers, spelled letters, unknown words, etc.
- How to mark pronunciation errors, truncated words, overlapping speech, noises, etc.
- How to mark utterances in foreign languages, or isolated foreign word or expressions.

Designing good guidelines proved to be far from straightforward. They have to meet several, sometime conflicting, requirements: they must ensure usability for several types of automatic processing, and take into account readability of the transcriptions by humans; they must help the transcribers in ambiguous situations and standardize the expected annotations, without bothering them with too many conventions which might be difficult to remember or causing lost time on fine details; they must cover most cases without becoming inconsistent. To summarize, they have to keep a good balance between completeness and simplicity.

In practice, the initial transcription guidelines have evolved to deal with the problems encountered during the sessions and the transcribers' questions. They were concerned with the use of capitalization, spelling of acronyms, marking of foreign words, etc. The tool itself also evolved accordingly, a good example being the management of overlapping speech.

4.5 Management of overlapping speech

Our priority was the transcription of single-channel broadcast news recordings for speech recognition systems training, and within this framework overlapping speech segments are currently discarded from further automatic exploitation. However, future tasks may use them. They make the transcription more complete, and it was judged less frustrating for the transcriber to be able to transcribe overlapping speech, whether this data will be used or not. Different situations were identified in the broadcast news task:

(1) clear foreground speech with background speech - e.g. translation with the original foreign voice in background: in this case, only the foreground voice had to be transcribed with an acoustic condition marker indicating background speech.
(2) limited interjections from other speakers (e.g. hum, yes...): they were indicated as instantaneous noises inside the main speaker transcription.
(3) a dialog between two speakers with frequent overlapping at the boundaries: when feasible, it could be transcribed using the specific mechanism for simultaneous speech described later.
(4) more than two overlapping speakers: the transcribers were requested not to annotate these.

It proved to be difficult to provide an ergonomic user interface for overlapping speech. In a first implementation, the constraint that the segmentations should be a strict partition of the signal was relaxed, and the last speech segment of one turn could overlap with the first speech segment of the next turn (solution 1 in Figure 4). The overlapping segments could be drawn in the temporal segmentation under the signal, but the resulting display in the text editor was confusing, because the two overlapping speech segments belonged to two separate speech turns and their simultaneity did not appear clearly enough. Several interfaces were tried and changed at the user's request before eventually choosing another representation (solution 2 in Figure 4). The overlapping part is clearly marked as a speech turn with two speakers. Despite the creation of this artificial speech turn, this led to a more acceptable solution in the interface. In the text editor, the parallelism between the two utterances appears clearly (Figure 1).

In conversational speech, overlapping is often so common that this approach becomes problematic both for the transcriber and for the eventual user. In the case of telephone speech recordings, two simultaneous speakers are often well enough separated on the separate channels for automated processing to go forward without special source-separation algorithms. In this case, it is much easier for the transcriber to segment and transcribe each channel as an independent stream, and the result is also more easily assimilated by training or testing programs as well as by human users. This approach to the transcription of heavily overlapped speech with a separate audio channel for each speaker (which is essentially the one that the LDC has been using) requires a different user interface as well as a different transcription specification. Providing such a solution in Transcriber is one of our goals for the future. Meanwhile, we

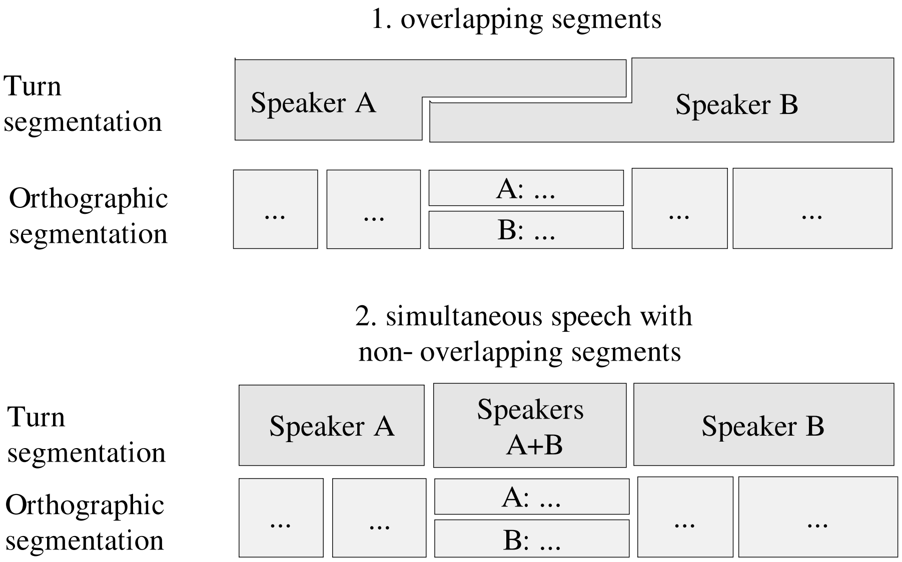

Figure 4. Two solutions tested for the representation of overlapping speech

understand that one user has solved the problem temporarily by running two simultaneous invocations of Transcriber, one for each channel! The resulting files are then merged (or split) automatically later on. A better solution will be to integrate the parallel streams of transcription under simultaneous program control.

4.6 Relevance of implementation choices

When looking back at the choices performed, we feel that the use of a scripting language considerably speeded up the development. The choice of Tcl is not mandatory, and the Tk widget has also been interfaced with the Perl scripting language. For a development restricted to the Windows platform, Visual Basic would bring similar advantages. In a multi-platform framework, the availability of the Snack extension for audio management in Tcl would be currently a decisive argument for still choosing Tcl.

A validating XML parser has been developed for the tool in Tcl using the tcLex library. However, XML parsing and validating in an interpreted language proved to be rather slow, especially with Unicode support. The current version of the parser would not be adapted for reading a long annotation file with word-level synchronizations or even phonetic annotations. We consider using another XML parser in the future, especially with the development of standard programming interfaces for the manipulation of XML documents (e.g. with the Document Object Model or DOM, Wood et al., 1998). This would also reduce the maintenance workload for this part in Transcriber.

Other limitations remain. The “undo” function should be improved to allow an unlimited number of undoes. Right-to-left writing and bi-directional support, which is needed for some languages, seems difficult to implement correctly with the current version of the Tk text widget. Display of transcription files for material exceeding one hour becomes slow in our configurations, mostly because of the numerous embedded buttons and images inside the text widget. Added to the parsing duration, this can make the launching of the tool last several tens of seconds, and scrolling in the text editor is also a bit less reactive. On the other hand, signal display remains perfectly reactive for signals up to several hours. This second feature, combined with the permanent and fluid synchronization with the text editor, seems to be currently the most appreciated feature of the tool.

5 Future directions

Though Transcriber has reached a stable state, its dissemination has prompted new needs. Users would benefit from further help such as automatic consistency checking, automatic alignment of transcription with signal, video display, or variable-speed playback. New application domains call for an increased flexibility in sound files management and annotation formats. This section presents these possible extensions.

5.1 Consistency checking

More tools are clearly needed for ensuring consistency of the transcriptions. Help should be provided for checking the consistency of proper names throughout the various transcriptions. A user-defined glossary and editable shortcuts have been introduced in the tool at the request of users; however, this is not yet completely satisfactory. A mechanism of automatic completion using previously written names in all existing transcriptions (compiled by hand or even automatically) seems to be an interesting solution and remains to be implemented. Online dictionaries, encyclopedias, or even maps for place names, should be made easily available to the transcriber. The LDC uses external databases of names, accessed via client-server connections, and it will be useful to some applications to provide support for this feature.

5.2 Automatic speech processing

Creating a pre-segmentation (cf. section 4.3) or checking the transcription by aligning it automatically with the signal is currently done by researchers using independent, elaborate tools (a speech recognition engine, acoustics models, and, for the alignment, a lexicon). It might be interesting to integrate such tools with Transcriber, for example to display the segments where poor alignment was detected. This might be useful for researchers, but could also, if the interface is user-friendly enough, be used directly by transcribers to check their transcription.

5.3 Multimedia

Speaker identification on television soundtracks is very difficult, because speakers are not introduced by the presenter in the same way as on the radio, their visual appearance being generally sufficient. In the short term, watching the video during the verification phase is an alternative (as has been the practice at the LDC). But the best solution for this problem would be to provide the complete video recording, not only the audio track. This would also ease the whole transcription process in the case of background noise. With the current development of video capabilities on standard computers, it can be hoped that easy technical solutions for interfacing the tool with a video player will be available in the near future. Such an interface will also be useful for other applications in which video recordings are to be transcribed or annotated, such as the study of gesture in communicative interaction.

5.4 Sound files management

Multiple sound files could be managed in a single transcription file. Specific functions should be available for multi-channel sound files (e.g. telephonic conversations as in the Switchboard task), for instance for playback of one channel at a time. It might then become useful to extend the interface to manage multiple windows. Additionally, variable-speed playback (as is commonly available in analog tape-based transcription systems, and in some software systems) will help productivity by permitting faster "proof-listening."

5.5 Format evolution

In the project, most effort was initially devoted to the user interface. The format choice was rather conservative and derived from existing LDC formats which proved already adapted to the broadcast news task. We also kept the single tree structure, which brought serious limitations to further extensions. Also, the tool is very sensitive to the modifications of the DTD. This limitation is not due to the XML paradigm which can be used for virtually any kind of data structure, but to the current implementation.

However, most user interface concepts in the tool which proved attractive for the users are not specific to the broadcast news task, and it quickly appeared useful to open the tool to other formats. A large number of other formats is currently used in the field of speech research. As an attempt to better coordinate existing efforts, the Text Encoding Initiative (TEI) provided in 1994 recommendations for the transcription of written and also spoken materials in SGML (Sperberg-McQueen and Burnard, 1994); current efforts aim at adapting TEI to XML and expanding its coverage. The MATE project is also trying to provide a standard format for spoken dialogue annotation (McKelvie, Isard, Mengel, Moller, Grosse and Klein, 2000). Various existing annotation formats are referenced online (Bird and Liberman, 1999-2000).

As a first step, the tool was adapted to the CHAT coding used in the CHILDES system (MacWhinney, 2000). Large amount of transcribed conversational speech is available in this format, and some researchers studying language acquisition would be interested in a version of Transcriber devoted to their needs, as an alternative to already existing tools. The DTD was extended with new tags and the source code had to be slightly modified for this task. But a more generic solution would be preferable, e.g. by simply reading the DTD or any adapted formal description of the format and having the interface of the tool automatically adapted to the chosen format.

Current developments are based upon Bird and Liberman's reflections about annotation graphs (Bird and Liberman, 2000). They show that virtually any existing annotation can be viewed as a labelled acyclic graph, in which some nodes bear ordered time values, and they develop a complete formalism for annotation graphs. Within this framework, all segments of the transcriptions are stored as an unordered set of typed arcs between identified nodes.

Switching to this framework for internal data management and for the reference transcription file format will lead to a much more generic tool, and conversion to other formats will become easier (Geoffrois, Barras, Bird and Wu, 2000). This does not preclude alternative formats, with time-ordered segments or in a human-readable format. For example, the internal format will no longer constrain new sections to impose a new turn, though such constraints can remain in the interface of the tool for a specific task.

6 Conclusions

We have presented Transcriber, a tool for assisting in the creation of speech corpora. It provides an intuitive and interactive interface for transcribing and annotating long duration signals.

Interface prototyping in a scripting language was shown to be an effective development approach, when robust libraries are available. Being distributed as free software, our project has been followed by numerous speech scientists and engineers who gave valuable hints for further developments that made the tool much more portable and usable. A web site has been designed for the distribution of the tool, and an announcement and a developer mailing list are in use. Our aim is to develop the future versions with the potential co-developers in a modular fashion with an interactive dialog, taking full advantage of the open source development framework.

After more than one year of testing the system, we feel that Transcriber is suitable for large-scale production of speech resources. It is now used by several research or development teams in various countries. Our initial target was very focused towards broadcast news transcription. But the interest in the tool showed that other areas need interactive tools that are easy to use. Future developments will use a generic data structure based on annotation graphs and provide multimedia extensions. This will lead to a much more user-configurable and task-configurable tool.

Acknowledgements

The design of the initial transcription conventions for French at LIMSI was coordinated by Martine Adda-Decker. Martine Garnier-Rizet coordinated the use of Transcriber within VECSYS and gave us valuable feedback on this. Snack's developer Kåre Sjölander was very helpful in always taking into account the changes which were needed for Transcriber. We are also glad to thank here all users and testers who gave us reports about their experience and their problems, since this helped us very much in improving the tool. We give many thanks to the transcribers for their patience using a program under development. Finally, the reviewers and the editors gave numerous and always judicious comments on the structure and on the style of this paper.

References

[1] Barras, C., Geoffrois, E., Wu, Z., Liberman, M., 1998. Transcriber: a Free Tool for Segmenting, Labeling and Transcribing Speech. In: *Proc. First Int. Conf. on Language Resources and Evaluation* (LREC 98), Granada, Spain, 28–30 May 1998, pp. 1373–1376.[`http://www.etca.fr/CTA/gip/Projets/Transcriber/`].

[2] Bird, S., Liberman, M., 2000. A Formal Framework for Linguistic Annotation. *Speech Communication*, this volume.

[3] Bird, S., Liberman, M., 1999–2000. *Linguistic Annotation.* [`http://www.ldc.upenn.edu/annotation/`].

[4] Bonnet, F, 1998–1999. *tcLex: a lexical analyzer generator for Tcl.* [`http://www.multimania.com/fbonnet/Tcl/tcLex/index.en.htm`].

[5] Bray, T., Paoli, J., Sperberg-McQueen, C. M. (Eds.), 1998. *Extensible Markup Language (XML) 1.0.* W3C Recommandation. See also [`http://www.w3.org/XML/`].

[6] Burger, S., 1999. TransEdit — a transcription tool from transcribers for transcribers. Presentation at the 9th International COCOSDA Workshop, Budapest, Hungary, 10 September 1999. (for informations contact `sburger@cs.cmu.edu`)

[7] Cassidy, S., Harrington, J., 2000. Multi-level Annotation of Speech: An Overview of The Emu Speech Database Management System. *Speech Communication*, this volume. See also [`http://www.shlrc.mq.edu.au/emu/`].

[8] Clark, J. (Ed.), 1999. *XSL Transformations (XSLT) Version 1.0.* W3C Recommendation. See also [`http://www.w3.org/Style/`].

[9] de Jong, F., Gauvain, J.-L., Hiemstra, D., Netter, K., 2000. Language-Based Multimedia Information Retrieval. In: *Proc. of the 6th RIAO Conference*, Paris, France, 12–14 April 2000. See also [`http://twentyone.tpd.tno.nl/olive/`].

[10] Deshmukh, N., Ganapathiraju, A., Gleeson, A., Hamaker, J., Picone, J., 1998. Resegmentation of Switchboard. In: *Proc. 5th Int. Conf. on Spoken Language Processing* (ICSLP'98), Sydney, Australia, 30 November–4 December 1998, pp. 1543–1546. See also [`http://www.isip.msstate.edu/projects/speech/software/`].

[11] Free Software Foundation, 1991. *GNU General Public License.* [`http://www.gnu.org/copyleft/gpl.html`].

[12] Gauvain, J.-L., Lamel, L., Adda, G., 1998. Partitioning and Transcription of Broadcast News Data. In: *Proc. 5th Int. Conf. on Spoken Language Processing* (ICSLP'98), Sydney, Australia, 30 November–4 December 1998, pp. 1335–1338.

[13] Geoffrois, E., Barras, C., Bird, S., Wu, Z., 2000. Transcribing with Annotation Graphs. In: *Proc. 2nd Int. Conf. on Language Resources and Evaluation* (LREC 2000), Athens, Greece, 31 May–2 June 2000, pp. 1517–1521.

[14] Hetherington, L., McCandless, M., 1996. SAPPHIRE: an extensible speech analysis and recognition tool based on Tcl/Tk. In: *Proc. 4th Int. Conf. on Spoken Language Processing* (ICSLP'96), Philadelphia, USA, 3-6 October 1996, pp. 1942–1945.

[15] Huckvale, M., 1987–1998. *SFS Speech Filing System.* [http://www.phon.ucl.ac.uk/resource/sfs.html].

[16] Jacobson, M., Michailovsky, B., Lowe, J. B., 2000. Linguistic documents synchronizing sound and text. *Speech Communication*, this volume. See also [http://lacito.vjf.cnrs.fr/ARCHIVAG/ENGLISH.htm].

[17] McCandless, M.K., 1998. *A Model for Interactive Computation: Applications to Speech Research.* Ph. D. Thesis, Massachusetts Institute of Technology.

[18] McKelvie, D., Isard, A., Mengel, A., Moller, M., Grosse, M., Klein, M., 2000. The MATE Workbench - an annotation tool for XML coded speech corpora. *Speech Communication*, this volume. See also [http://mate.nis.sdu.dk/].

[19] MacWhinney, B., 2000. *The CHILDES project: Tools for analyzing talk.* Third Edition. Mahwah, NJ: Lawrence Erlbaum Associates. See also [http://childes.psy.cmu.edu/].

[20] NIST, 1998. *A Universal Transcription Format (UTF) annotation specification for evaluation of spoken language technology corpora.* [http://www.nist.gov/speech/hub4_98/hub4_98.htm].

[21] Ousterhout, J. K., 1994. *Tcl and the Tk Toolkit.* Addison Wesley, ISBN: 3-89319-793-1. See also [http://www.scriptics.com/].

[22] Ousterhout, J. K., 1998. Scripting: Higher-Level Programming for the 21st Century. *IEEE Computer Magazine*, 31(3), March 1998.

[23] Schalkwyk, J., de Villiers, J., van Vuuren, S., Vermeulen, P., 1997. CSLUsh: an extensible research environment. In: *Proc. 5th European Conf. on Speech Communication and Technology* (Eurospeech 97), Rhodes (Greece), 22–25 September 1997, pp. 689–692. See also [http://cslu.cse.ogi.edu/toolkit/].

[24] Sjölander, K., 1997–2000. *The Snack Sound Visualization Module.* [http://www.speech.kth.se/snack/].

[25] Sjölander, K., Beskow, J., Gustafson, J., Lewin, E., Carlson, R., Granström, B., 1998. Web-based Educational Tools for Speech Technology. In: *Proc. 5th Int. Conf. on Spoken Language Processing* (ICSLP'98), Sydney, Australia, 30 November–4 December 1998, pp. 3217–3220.

[26] Sperberg-McQueen, C. M., Burnard, L. (Eds.), 1994. *TEI Guidelines for Electronic Text Encoding and Interchange (P3).* [http://etext.lib.virginia.edu/TEI.html]. See also [http://www.tei-c.org/].

[27] Stallman, R., 1998. *The GNU Project.* [http://www.gnu.org/philosophy/]. In: DiBona, C., Ockman, S., Stone, M. (Eds.), 1999. *Open Sources - Voices from the Open Source Revolution.* O'Reilly, ISBN: 1-56592-582-3.

[28] Stern, R., 1997. Specification of the 1996 Hub 4 Broadcast News Evaluation. In: *Proc. of the DARPA Speech Recognition Workshop*, Chantilly, USA, 2–5 February 1997, pp. 7–14.

[29] The Unicode Consortium, 2000. *The Unicode Standard, Version 3.0.* Addison-Wesley Longman Publisher, ISBN 0-201-61633-5. See also [`http://www.unicode.org/`].

[30] Wood, L. et al. (Eds.), 1998. *Document Object Model (DOM) Level 1 Specification.* W3C Recommendation. Available from [`http://www.w3.org/DOM/`].