The TASX-environment: an XML-based toolset for the creation of multimodal corpora

Jan-Torsten Milde
Department of Linguistics and Literary Studies
University of Bielefeld, Germany
milde@coli.uni-bielefeld.de

Abstract

The design and implementation of an XML-based corpus environment for multilevel annotated multimodal (language) data is described. The TASX-environment (TASX: Time Aligned Signal data eXchange format) constitutes a technical basis for all aspects of the corpus setup procedure: XML-based annotation of the multimodal data, transformation of non XML-annotations, and the web-based analysis and dissemination of the data.

1 Introduction

In this paper we describe the design and implementation of an XML-based corpus environment for complex annotated multimodal data. The TASX-environment[1] presented here supports the complete corpus setup procedure: XML-based annotation of raw speech and video data, the transformation of non XML-data and the analysis and dissemination of the corpus.

The development of the corpus environment complements the LeaP[2] project, which explores the acquisition of prosody by both second language learners of German and English. From the collected data an XML-annotated multimodal corpus is created. The TASX-environment is also used in number of linguistic projects at our faculty and externally, including work on language documentation, multi-party converstional analysis, multimodal construction dialogues, doctor patient interaction and others.

The paper is organized in four sections. The underlying XML-based TASX format is explained and the components of the TASX-environment are described in more detail. Finally, a short conslusion is given.

2 The TASX format

A central aspect of our research is to explore up to which point current standard XML technology (XML, XSL-T, XSL-FO, XPATH, SVG, XQUERY) can be used to model multimodal corpora, to transform, query and distribute the content of such corpora and to perform adequate linguistic analysis. As a result, all linguistic data in our system is stored in an XML-based format called TASX: the *T*ime *A*ligned *S*ignal data e*X*change format.

A TASX-annotated corpus consists of a set of sessions, each one holding an arbitrary number of descriptive tiers, called layers. Each layer consists of a set of separated events. Each event stores some textual information (e.g. a syllable or the description of gesture) and is linked to the primary audio data by two time stamps denoting the interval of this event. Relations between events on different tiers can be encoded by defining links using the ID/IDREFS mechanism of XML. This is similar to the approach of stand-off markup as proposed by MATE (Dybkaer et al., June 1999), respectivly NITE (Carletta et al., 2002). Finally, arbitrary meta-data can be assigned to the complete corpus, each session, each layer and each event. Meta data is represented as a named set of attribute-value pairs. No further restrictions are made on the AV pairs. The design of TASX is based on the careful analysis of common annotation formats and tools. We tried to maximally reduce these formats while still allowing to represent all of the stored information and (even more important) the represented *information structure.* The TASX-format is thus powerful enough to encode most of the corpus annotation formats in use. Indeed a number of format transformation programms have been implemented. For a more detailed description of the TASX-environment

[1] http://coli.lili.uni-bielefeld.de/~milde/tasx/

[2] http://www.spectrum.uni-bielefeld.de/LeaP/

see also (Milde and Gut, 2001), (Milde and Gut, 2002a), (Milde and Gut, 2002b). Table 1 shows the formal TASX-format specification. Two levels of TASX-annotation can be distinguished:

1. TASX level 1: all events do immediately refer to the common timeline of the underlying signal. This approach has been taken by most of the currently available annotation formats and annotation tools.
2. TASX level 2: events can either refer to the common timeline or events can refer to events in other layers. This allows to construct hierarchical annotation stuctures. While conceptually integrated into the TASX format, there is currently no tool available to manipulate TASX level 2 annotated language data.

Element definition	Attributes
<!ELEMENT tasx (meta*,session+)>	--
<!ELEMENT session (meta*,layer+)>	<!ATTLIST session s-id CDATA #REQUIRED day CDATA #REQUIRED ref IDREF #IMPLIED month CDATA #REQUIRED year CDATA #REQUIRED>
<!ELEMENT layer (meta*,event+)>	<!ATTLIST layer l-id CDATA #REQUIRED ref IDREF #IMPLIED>
<!ELEMENT event (#PCDATA,meta*)>	<!ATTLIST event e-id CDATA #REQUIRED start CDATA #REQUIRED end CDATA #REQUIRED ref IDREF #IMPLIED mid CDATA #IMPLIED len CDATA #IMPLIED>
<!ELEMENT meta (desc*)>	<!ATTLIST meta
<!ELEMENT desc (name,val)>	m-id CDATA #REQUIRED
<!ELEMENT name (#PCDATA)>	ref IDREF #IMPLIED
<!ELEMENT val (#PCDATA)>	access CDATA #IMPLIED
	level CDATA #IMPLIED>

Table 1: The TASX DTD.

2.1 The TASX-annotator and the corpus engine

The complete TASX-environment consists of:

- tools for the annotation of empirical language data (video and audio material),
- a simple meta-data editor
- programs for the transformation of various formats of linguistic standard software (Transcriber, Praat, ESPS/waves+, SyncWriter, Exmaralda etc.)
- a set of programs for linguistic analysis of the TASX-annotated data, and
- a corpus system for the distribution of language data via the internet, including interactive corpus query and multimodal data display in a standard web browser.

In the following sections these modules are described in more detail

2.2 The TASX-annotator

The TASX-annotator is a central component of the TASX-environment. The tool allows the multilevel annotation and transcription of video (multi-channel) and audio data (see figure 1). A separate video playback window is opened up for each video file making it possible to e.g. display multiple perspectives of the same scene. The video playback is synchronized with the transcription. For audio transcriptions an oszillogram is calculated and is displayed inside the main window. The programm is very user friendly and can be used without a high level of computer skills. It is possible to completely control the tool by either mouse *or* by keyboard shortcuts.

Different data views are programmed (time-aligned partiture, word-aligned partiture, sequential text view) to make annotation as effective as possible.

The *time aligned view* is organized as a two dimensinal grid of infinite size. A layer is presented as a horizontal tier of events. The order of the layers is arbitrary and can be changed instantly. The user is able to define time intervals by dragging the mouse. Each time interval represents an event. The event is displayed as a graphical box which can be selected and moved with the mouse. The content of an

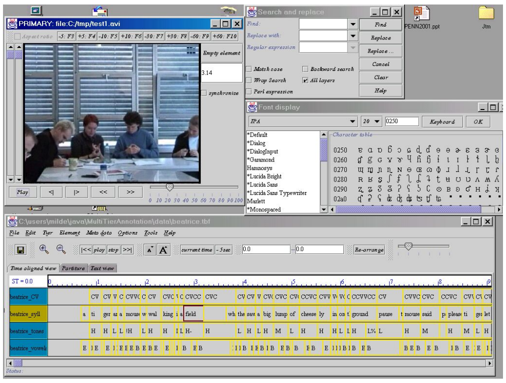

Figure 1: A screenshot of the TASX-annotator. In the bottom half, the main panel is visible, where the time aligned tier view has been selected. On top of the main window, the font selection panel is visible (showing some IPA characters). Above it, the find tool is shown. In the upper left corner the video display can been seen.

event is entered in an additional text field. Any (unicode) font (e.g. IPA fonts, HamNoSys fonts etc.) available for the operating system can be used for the transcription. The user can choose font and fontpage from a table displaying all characters of the selected font. It is also possible to define a virtual keyboard which maps the given keystrokes to arbitrary characters of the target font.

In the *text view* the data can be manipulated in a standard text editor panel. The content of the editor represents the layer and each line represents an event. A list selection box allows switching between different layers. It is possible to transfer text from standard text editors, e.g. Microsoft Word, by cut and paste operations. In order to additionally speed up the transcription process, a word completion function has been implemented for the text view. Entering the initial letter of a word and consecutively pressing CTRL+L will bring up all words starting with this letter. Once the text is tranferred into the TASX-annotator, the events still have to be aligned with the primary audio and video data. Switching back to the time aligned view and moving the events with the mouse makes this task quite simple.

In the *partiture view* the data cannot be edited. In practice this means that the data is transformed into an HTML table and then displayed to the user. A number of different HTML formatted views have been designed. The views can also be saved to external files and loaded into standard web browsers.

One potential strength of the TASX-annotator is its manner of handling the export/import of XML based information. A standard way of solving this problem would be the implementation of a set of format specific XML parsers which construct the internal representation (e.g. JDom) of the XML file. In the TASX-annotator we follow a different approach. The system integrates an XSL-T processor (saxon), making it easy to perform on the fly data transformations. The import of an XML-file is split into two steps: first an XSL-T stylesheet transforms the XML file into TASX, second another XSL-T stylesheet will transform the TASX file into a simple text oriented format. This format can be loaded efficiently.

A crucial problem when setting up larger corpora are inter-anotator transcription errors. While the TASX-annotator is designed to be used by a single person, it still provides a number of routines to combine (merge), control and align annotations created by a larger team of people. We do not integrate more complex control functions. This contradicts our approach of clearly separating corpus creation from corpus analysis.

Currently a number of related annotation tools are under development (e.g. Elan (Brugman and Wittenburg, 2001), AGTK (Bird et al., 2001), (Bird and Liberman, 1999), Anvil (Kipp, 2001), Exmaralada (Schmidt, 2001)), each of them designed for a specific target audience. Most of the tools are using Java as an implementation base and encode the linguistic data in a comparable way as proposed here (XML-based, using time spans to mark events, separating meta-data and content). As a result it becomes relativly simple to convert/generate TASX-annotated corpora into/from these formats.

2.3 Transcoding tools

The development of tools for the TASX-environment is based on the concept that a re-implementation of functionalities already available in other language and speech processing software is not necessary. Established software systems such as Praat or ESPS/waves+ do not need to be duplicated.

The TASX-environment therefore focuses on the development of transcoding filters from and into various formats. These include: Praat/freq, Praat/label, ESPS/waves+, ESPS/F0-analysis, Transcriber, annotation graphs stored in XML, SyncWriter and basic text formats (see table 2). In addition, filters for data import and export of the Exmaralda system (Schmidt, 2001) are available.

TASX	import	export
Annotation graphs	XSL-T	XSL-T
Exmaralda	XSL-T/Java	XSL-T/J.
HTML-table	–	XSL-T
HTML-partiture	–	XSL-T
RTF	–	XSL-T/J.
Anvil	XSL-T	XSL-T
EAF	XSL-T	XSL-T
Praat-label	Perl/XSL-T	XSL-T
Transcriber(STM)	Perl/XSL-T	XSL-T
WaveSurfer	Perl/XSL-T	XSL-T
ESPS-label	Perl	XSL-T
ESPS-freq	Perl/XSL-T/J.	XSL-T
SyncWriter	Perl	–

Table 2: List of currently implemented transcoding tools. The table shows the programming languages used to implement the transcoders.

Most of these components are implemented in Java, transformations are defined in XSL-T and a smaller number of additional tools is written in Perl (mainly to transform non-XML data). We will gradually increase the number of transcoding programms. All programms follow a unix like processing approach: input can be read from a file or standard in, the transcoding result will be printed to standard out. This allows to combine the transcoders in arbitrary ways. TASX can thus be seen as a common interlingua for the supported linguistic formats.

2.4 Pause tracker

To speed up the annotation process a pause tracking programm has been developed. The programm separates speech from pauses and generates a TASX annotated XML document with two tiers, one holding all pause events, the other one holding all speech events.

The tracker uses Praat (Boersma, 2001) to perform the actual speech analysis. It simply calculates the pitch curve of the audio signal. If no pitch is detected, then non-speech is assumed, otherwise speech. In a second step, the results of this classification are combined to continuous stretches of pauses/speech. Finally the TASX conformant output is generated.

The pause tracker has shown to work quite reliably on a set of recording in different languages (Japanese, English, German, Saterfriesisch, French, Ega). Even if tracking is far from perfect, the human annotator gets a good pre-segmentation of the signal. This allows to move very quickly through the file, possibly performing minor adjustments to the boundaries or combining a set of separated events of one speaker.

While the pause tracker gives good results when doing conversational analysis it is not of much help for fine grained phonetic research. Here a tracking system for vowels and consonants would be very useful. Garcia et.al. ((Garcia et al., 2002)) are working on such a system.

2.5 Statistical analysis

In the inital design phase of the TASX system we planned to implement the statistical analysis in XSL-T and Java. Indeed, a number of smaller programs have been realized in this technique. Unfortuneatly it quickly became evident, that XSL-T is not suited to perform such calculations on larger sets of data. It lacks high precision arithmetic functions and consumes to much memory. When using external Java functions, a large number of data conversions have to take place. Also the resulting code is very hard to read and debug.

Instead we have chosen to use the R system, an open source implementation of the `S-Plus` statistics language (Ihaka and Gentleman, 1996), (Venables and Ripley, 1999). R implements all major statistical tests and calculations and is equipped with a large number of high level graphic routines to generate visually informative presentations of the results. Even more important it includes efficient input/output routines to load and save semistructured data (either XML-annotated or plain ascii text).

3 The corpus system

The main function of the corpus system constitutes the internet-based dissemination of the corpus data. With the currently implemented interface it is also possible to inspect and query the speech corpus, to listen to the audio material and to display the graphic representation of the waveforms in a standard web browser. We make use of the built-in features of the web browser here. Furthermore, the PAX-tools (Gibbon and Trippel, 2001) for displaying the intonation contour, the intensity and the spectrogram of the selected regions in the audio file can be integrated.

When playing back the sound file, both the audio parts and the waveform images are generated automatically by a small Java servlet program. The servlet parses the XML-annotated corpus, extracts the time stamps of the relevant events and then cuts out the corresponding parts of the original sound file.

The corpus system is split into two larger subcomponents: the *information pool* and the *corpus engine* (see figure 2). The information pool stores the primary data (raw audio data) as well as the XML-annotated transcriptions of the audio files. The corpus engine consists of five subsystems:

1. Web-client: the interactive user interface is completly defined to run in a standard web browser. We are using HTML-query forms which activate services on the server side to generate XSL-T-filters processing the data. Waveforms are displayed using SVG. This will allow the user to select parts of the sound signal and to perfom more complex phonetic analyses.

2. Web-server: the web server distributes the corpus information in several standard formats (XML, HTML, PDF, SVG, WAV).

3. Servlet-engine: the servlet engine activates the suitable services on the server side (transformation of XML-annotated data, on-the-fly phonetic analysis, generation of graphics).

4. Servlets: a set of TASX/XML-aware servlets are used to transform the data in numerous ways: generating HTML to be displayed in the browser, generating PDF to be printed out, generating wavefiles and images of the waveforms. XSL-T and XSL-FO are used to perfom the transformations. The servlets have access to the information pool and the relational database.

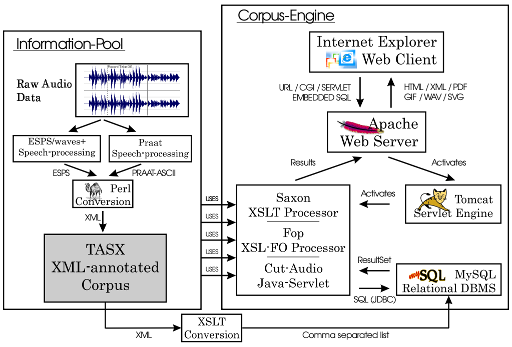

Figure 2: The system architecture of the corpus system. The corpus system is split into two subsystems: the information pool (left) storing the TASX-annotated data and the corpus engine (right) distributing the data over the internet.

5. Relational database: in order to improve the system performance, the XML-annotated corpus data is stored in a relational database. The database basically replaces a standard file system. An XSL-T-program translates the XML-annotated corpus data into a suitable format for the DBMS.

The implementation of the corpus system is based on open source software. The TASX-annotator is a pure Java application; all other tools are smaller XSL-T and perl scripts. As a result, the complete TASX-environment runs on Windows and Unix platforms. The software is distributed under GPL and can be downloaded from our website[3].

4 Conclusions

The TASX-based approach has proved to be highly efficient and enabled us to develop a powerful linguistic environment in a very short time. Due to the highly structured format of the TASX-annotated data more complex research questions can be investigated in a systematic way. A TASX corpus can be transformed

[3] http://coli.lili.uni-bielefeld.de/∼milde/tasx/

into large number of different formats, making it possible to integrate TASX-aware components into standard linguistic working procedures. This will hopefully lead to the creation of linguistic resources which can be used over a long period of time by different researchers with a wide range of scientific goals.

References

S. Bird and M. Liberman. 1999. A Formal Framework for Linguistic Annotation. Technical Report MS-CIS-99-01, Department of Computer and Information Science, University of Pennsylvania.

Steven Bird, Kazuaki Maeda, and Xiaoyi Ma. 2001. Agtk: the annotation graph toolkit. In Peter Buneman Steven Bird and Mark Liberman, editors, *IRCS Workshop on Linguistic Databases, University of Pennsylvania, Philadelphia, USA*.

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345.

Hennie Brugman and Peter Wittenburg. 2001. Mpi tools for linguistic annotation. In Peter Buneman Steven Bird and Mark Liberman, editors, *IRCS Workshop on Linguistic Databases, University of Pennsylvania, Philadelphia, USA*.

J. Carletta, D. McKelvie, and Isard A. 2002. Supporting linguistic annotation using xml and stylesheets. In G. Sampson and D. McCarthy, editors, *Readings in Corpus Linguistic, Continuum International.*

L. Dybkaer, M. B. Moeller, N. O. Bernsen, J. Carletta, A. Isard, M. Klein, D. McKelvie, and A. Mengel. June 1999. The mate workbench. In David Traum, editor, *Proceedings of ACL'99, Demonstration Abstracts. University of Maryland*, pages 12 – 13.

J.E. Garcia, U. B. Gut, and A. Galves. 2002. Vocale - a semi-automatic annotation tool for prosodic research. In B. Bel and I. Marlien, editors, *Proceedings of the Speech Prosody 2002 conference, 11-13 April 2002. Aix-en-Provence: Laboratoire Parole et Langage*, pages 327 – 330.

D. Gibbon and T. Trippel. 2001. Pax - an annotation based concordancing toolkit. In Peter Buneman Steven Bird and Mark Liberman, editors, *IRCS Workshop on Linguistic Databases, University of Pennsylvania, Philadelphia, USA*.

R. Ihaka and R. Gentleman. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314.

Michael Kipp. 2001. Anvil - a generic annotation tool for multimodal dialogue. In *Proceedings of the Eurospeech 2001, Aalborg*, pages 1367 – 1370.

J.-T. Milde and U. B. Gut. 2001. The TASX-engine: an XML-based corpus database for time aligned language data. In Peter Buneman Steven Bird and Mark Liberman, editors, *IRCS Workshop on Linguistic Databases, University of Pennsylvania, Philadelphia, USA*.

J.-T. Milde and U. B. Gut. 2002a. A prosodic corpus of non-native speech. In B. Bel and I. Marlien, editors, *Proceedings of the Speech Prosody 2002 conference, 11-13 April 2002. Aix-en-Provence: Laboratoire Parole et Langage*, pages 503 – 506.

J.-T. Milde and U. B. Gut. 2002b. The tasx-environment: an xml-based toolset for time aligned speech corpora. In *Proceedings of the third international conference on language resources and evaluation (LREC 2002, Gran Canaria*.

T. Schmidt. 2001. Gesprächstranskription auf dem Computer - das System EXMARaLDA. *Gesprächsforschung, http://www.gespraechsforschung-ozs.de*, 2.

W. N. Venables and B. D. Ripley. 1999. *Modern Applied Statistics with S-Plus. Third Edition.* Springer. ISBN 0-387-98825-4.