Karaoke for linguists

An audio-enabled interlinear viewer

Stuart McGill (stuart.mcgill@tiscali.co.uk), Endangered Languages Academic Programme, SOAS

Endangered *data* and XML

Almost all language data created nowadays is *digital* data, which can very quickly become obsolete. This is a Word file from 1992 viewed in Word 2007:

```
@˜€Ð[1]Êx
m�¡  m£ d¡a!   nãm�  b�   d�  a! m£
I am going homeI am not going home
```

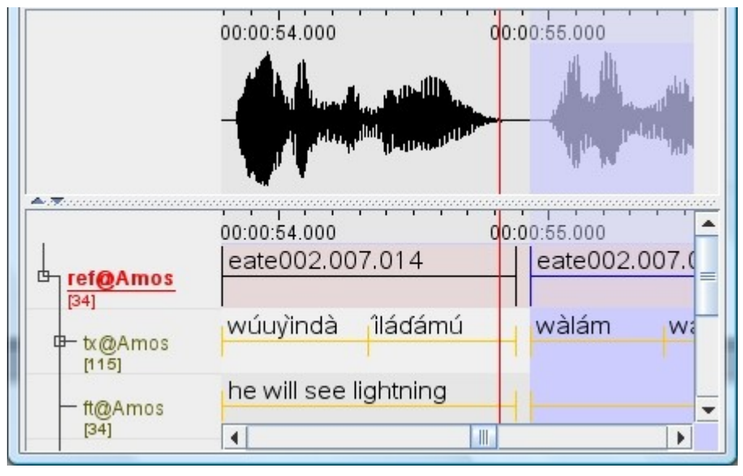
This 'data' from the Paasaal language (Ghana) is now useless after only 15 years. To help solve such problems, a format called **XML** (e**X**tensible **M**arkup **L**anguage) has become the standard for text archival. XML documents contain data 'marked up' with descriptive <tags>, and can be stored as easily-readable plain text:

```
<sentence>
   <text>ŋáá mú díya rɛ</text>
   <translation>I'm going home</translation>
</sentence>
```

While converting Word documents to XML might be another 'burden' for linguists, it also an *opportunity*. The viewer described here 'taps in' to the XML created by two common programs.

Time-alignment

Ideally audio and video recordings should be TIME-ALIGNED to their transcriptions – this allows instant playback of any particular sentence. The MPI program **ELAN** helps to set up time-alignment:



The data is stored as XML, including the start/end times of each sentence.

Interlinear annotations

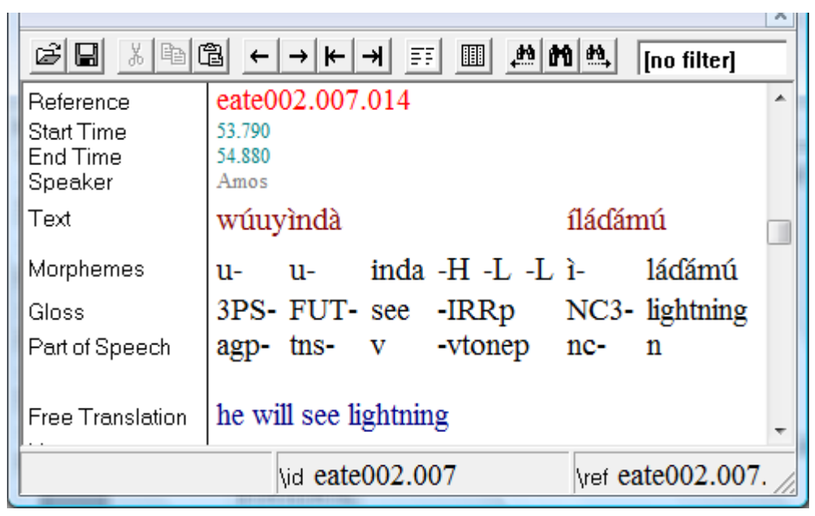
Audio recordings are often accompanied by INTERLINEAR glosses. When a language is not well-known, interlinear glosses help the listener to work out the meaning of each word or morpheme:

wú-u-yìndà í-ládámú

3S-FUT-see NC3-lightning (NC = 'noun class')

he will see lightning [Cicipu language, Nigeria]

These annotations can be created by exporting the XML from ELAN into SIL's **Toolbox** program, which can in turn save them as XML.

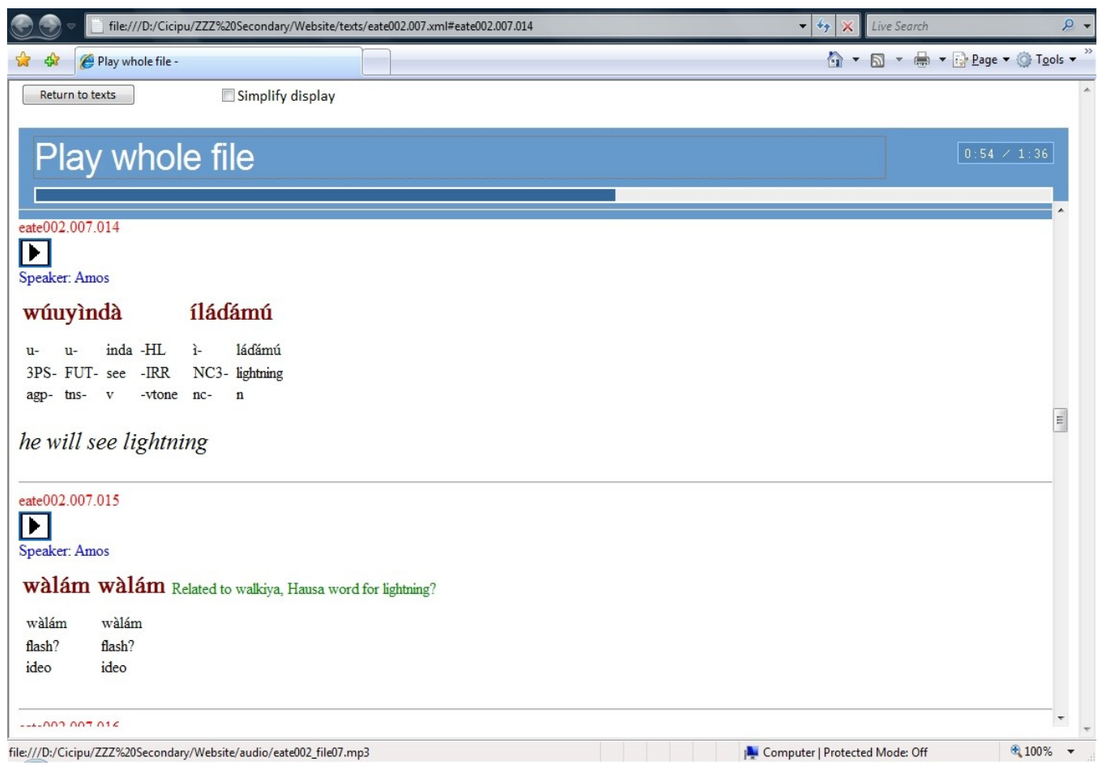


Presenting the texts

Neither ELAN nor Toolbox are good at *presenting* data. ELAN has a complex interface making it hard to learn to use, and Toolbox doesn't integrate with audio at all.

There *are* polished language documentation websites/CD-ROMs with text linked to sound, music, or video, but these require **time**, **money**, computing **expertise**, and a specialised **development environment** – OWLs (Ordinary Working Linguists) may have none of these!

The solution demonstrated below avoids these problems because it is simple and generic. The user just copies their time-aligned texts and MP3 files into the right folders. It works because XML is easy to change into **HTML** – which of course is what web pages are made of.



Adjusting fonts, size, colour, etc... can be done by anyone with a basic knowledge of HTML, but this is *not* required – what you see at the computer below is what you get 'out-of-the-box'.

No specialist software is needed to create or view the texts online – just a web browser with Flash, which almost all computers have. Alternatively the texts and MP3 files can be distributed on CD or DVD.

In a way, the aim is to extend the principle of karaoke – not only can amateurs sing along to the MP3 files, they can even create their own karaoke machine.

Possible applications

- **Website** for the use of **native speakers** and others
- Making texts easily-accessible to **linguists**, **anthropologists**, or others interested in the language
- Distribution of texts to **institutions** such as local archives
- Presenting examples in academic **talks**

Try it yourself on the computer below!

Technical details

The display of the texts is handled by an XSL transform acting on the Toolbox XML export. The transform retrieves the MP3 filename and loads a free MP3 player called Sound Manager (schillmania.com), then reads in each utterance and converts it to interlinear text. JavaScript is used to start and stop the player based on the timestamps, as well as to update the text in response to notification events.

Flash 8+ is required. A general limitation of Flash is that it can only play 44.1 kHz MP3, not 48kHz. MP3 files can easily be resampled using freely available software (e.g. BeSweet).

The XSL template used here is based on one originally developed by H. Moa. Thanks are also due to Peter Austin for helpful suggestions and comments.