The BOSS Architecture for Unit Selection Synthesis

Stefan Breuer

Institute for Communication Research and Phonetics (IKP), Bonn

breuer@ikp.uni-bonn.de

History

- BOSS has been under development at IKP for some years
- based on work in the Verbmobil project by Karlheinz Stöber (1998/1999)
- a complete re-implementation with new data structures by Karlheinz Stöber led to BOSS II in 2000.
- extended by Jörg Bröggelwirth, Mathijs Visser (Eindhoven), Philip Groß, and Stefan Breuer between 2000 and 2003.
- moved to version 3 in July 2004 by Philip Groß and Stefan Breuer

Applications

- Adaptation to Dutch by Esther Klabbers, Raymond Veldhuis and Mathijs Visser in 2001, presented at Eurospeech 2001 and the satellite SSW4 workshop in Pitlochry.
- Adaptation to a directory enquiries front-end for klickTel GmbH by Julia Abresch and Stefan Breuer, presented at ICPhS 2003
- Adaptation to Polish in collaboration with Grazyna Demenko (Szczyrk 2003)
- Adaptation to British English in collaboration with Mark Huckvale (UCL) and YOU!

Features

- One of the first open systems to support unit selection with large corpora
- C++ (gcc / Linux)
- modular
- client / server
- Standardized methods for data storage:
  - communication via XML data structures
  - uses an SQL database for the retrieval of speech data annotations at runtime
- Open Source
- but: a platform for development, not a ready-to-use TTS system!

The structure of BOSS

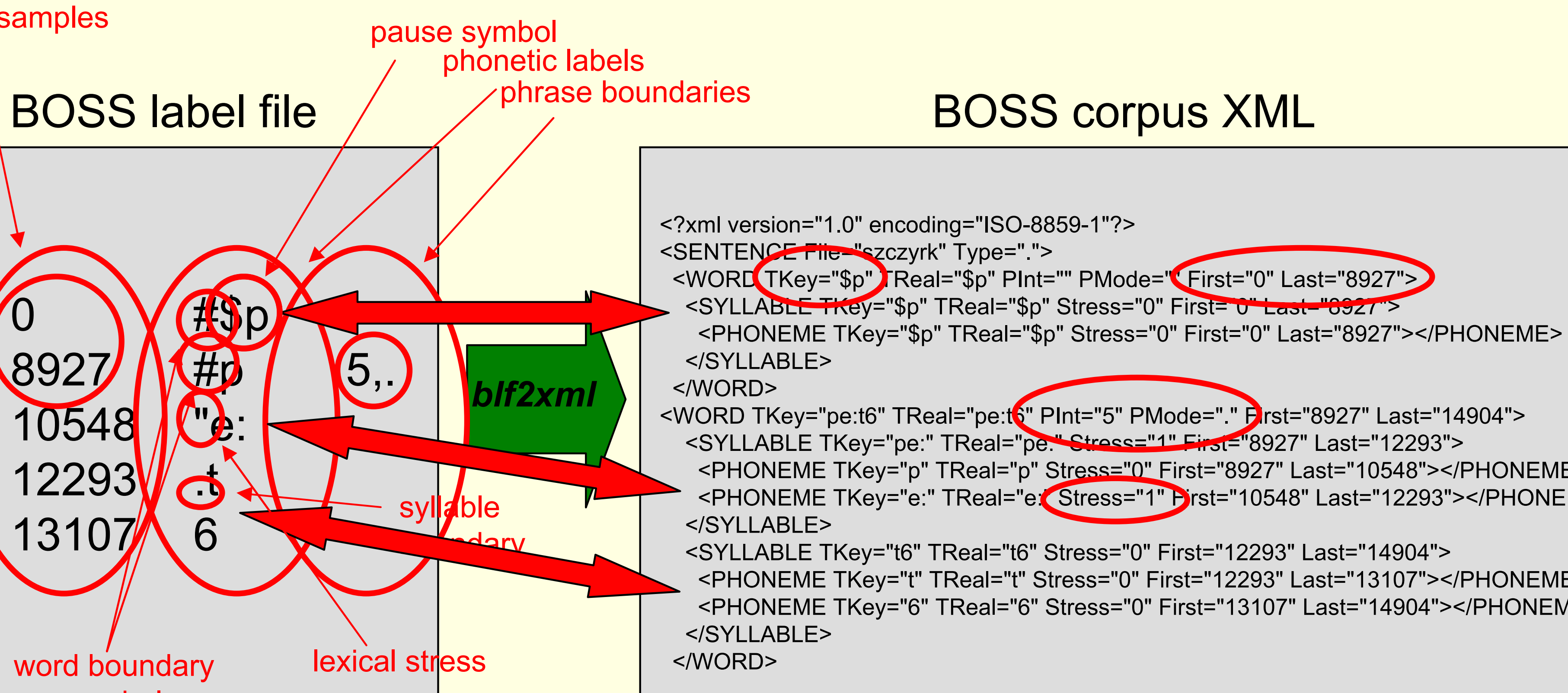
- a collection of tools for XML-based annotation of the speech data
- a collection of synthesis modules for transcription, duration prediction, network communication etc. in the form of class libraries
- a server executable, the actual synthesis, that integrates the synthesis modules and provides cost functions and unit selection capabilities
- an example client application that does some basic text-preprocessing and converts input text into the XML structure required by BOSS. The client uses the BOSS network module to communicate with the server.

Data organisation in BOSS

- Annotation data for speech corpora is represented and stored in a specialized XML format. For faster access at runtime, the data is converted into an SQL database structure.
- An analogous XML format exists for the internal communication between BOSS modules. All information about the text that is to be synthesized, e.g. its transcription and predicted prosodic parameters, are stored in this structure. The XML document is represented by a DOM (Document Object Model) in the BOSS server.
- Advantages:
  - third-party software tools and libraries for XML manipulation
  - easier exchange of data with other systems

Preparing a corpus for BOSS

- as a first step, all label files have to be converted into a corpus XML file. This is done by a tool called *blf2xml*.
- label files in BOSS (BLF) have a very simple format, which makes it easy to convert from or into BLF



BOSS label file

```
0       #$p
8927    #p      5,.
10548   "e:
12293   .t
13107   6
```

BOSS corpus XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SENTENCE File="szczyrk" Type=".">
 <WORD TKey="$p" TReal="$p" PInt="" PMode="" First="0" Last="8927">
  <SYLLABLE TKey="$p" TReal="$p" Stress="0" First="0" Last="8927">
   <PHONEME TKey="$p" TReal="$p" Stress="0" First="0" Last="8927"></PHONEME>
  </SYLLABLE>
 </WORD>
<WORD TKey="pe:t6" TReal="pe:t6" PInt="5" PMode="." First="8927" Last="14904">
  <SYLLABLE TKey="pe:" TReal="pe:" Stress="1" First="8927" Last="12293">
   <PHONEME TKey="p" TReal="p" Stress="0" First="8927" Last="10548"></PHONEME
   <PHONEME TKey="e:" TReal="e:" Stress="1" First="10548" Last="12293"></PHONE
  </SYLLABLE>
  <SYLLABLE TKey="t6" TReal="t6" Stress="0" First="12293" Last="14904">
   <PHONEME TKey="t" TReal="t" Stress="0" First="12293" Last="13107"></PHONEME
   <PHONEME TKey="6" TReal="6" Stress="0" First="13107" Last="14904"></PHONEM
  </SYLLABLE>
 </WORD>
```

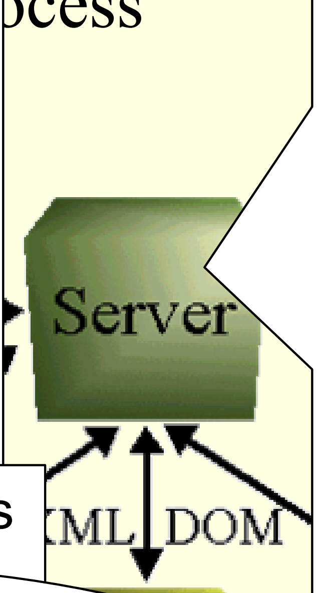
Preparing a corpus for BOSS: Adding information

- additional information about the speech files can be inserted in the form of attributes (such as *First*, *Last* etc. in the previous examples) of elements (i.e. WORDS, SYLLABLES etc.)
- this can be done with the aid of the corpus tools that are part of BOSS II.
- Examples of the attributes that can be inserted include
    - *CLeft* and *CRight*, which contain the preceding and following phonetic context of each element
    - *RM[0-10]* and *LM [0-10]* which contain Mel frequency cepstrum coefficients for the left and right boundaries of each element.
- At the end of this process, the XML documents are converted into a relational database table structure for performance reasons.

he class unitSelection loops through the elements f the XML DOM and tries to find the units that are ghest up in the hierarchy (starting with WORD) by uerying the SQL corpus database. If the WORD is ot found, it proceeds to find the SYLLABLEs ontained in the WORD, next the PHONEME ements. The results of each query (if ored in a special data structure which e DOM of the resp nown as the *pre-se* equence of unit can e columns represe nd the rows repres

The resulting s ignal is transferred to t syn

Thank you for your



handles network communication an contains the modu scheduler that integrates the vario synthesis compone and calls them in th appropriate order. scheduler, a class called *boss_synthe* is the core of the system. It takes a sentence element put (i.e. the heduler, as each dule, processes

he first module to be called is usually the transcription. This module dds the SYLLABLE and PHONEME elements and provides the with the tributes *TKey* (which contains the transcription of the element) as well the phrasing attributes *PInt* and *PMode* to the XML DOM.
he element structure is now analogous to the corpus XML structure.
he German transcription module inherits from the base class

oss_transcription: ***boss_transcription_de* : *boss_transcription***

he German module uses a three-step process to yield a transcription r each WORD element:

1. lexicon lookup
2. morpheme decomposition

ang s, s gua BOS to i an t

The class BOSS_ConMa then retrieves the names the relevant speech files from the database, loads signals, concatenates the and applies some spectra smoothing.