INVESTIGATIONES LINGUISTICAE, VOL. XIII; POZNAŃ, DECEMBER 2006

Close Copy Speech Synthesis for Speech Perception Testing[1]

Jolanta Bachan[1] and Dafydd Gibbon[2]

[1]Adam Mickiewicz University
ul. Międzychodzka 5, 60-371 Poznań, Poland

jolabachan@op.pl

[1,2]Universität Bielefeld
Postfach 100131, 33501 Bielefeld, Germany

gibbon@uni-bielefeld.de

Abstract

The present study is concerned with developing a speech synthesis subcomponent for perception testing in the context of evaluating cochlear implants in children. We provide a detailed requirements analysis, and develop a strategy for maximally high quality speech synthesis using Close Copy Speech synthesis techniques with a diphone based speech synthesiser, MBROLA. The close copy concept used in this work defines close copy as a function from a pair of speech signal recording and a phonemic annotation aligned with the recording into the pronunciation specification interface of the speech synthesiser. The design procedure has three phases: Manual Close Copy Speech (MCCS) synthesis as a "best case gold standard", in which the function is implemented manually as a preliminary step; Automatic Close Copy Speech (ACCS) synthesis, in which the steps taken in manual transformation are emulated by software; finally, Parametric Close Copy Speech (PCCS) synthesis, in which prosodic parameters are modifiable while retaining the diphones. This contribution reports on the MCCS and ACCS synthesis phases.

1 Objectives and context for Close Copy Speech synthesis development

1.1 Objectives and procedure

The aim of this study is, first, to develop a restricted domain speech synthesis concept for automatically generating acoustic stimuli for use in evaluating cochlear implants for children and, second, to implement a prototype synthesiser. The main motivation for including a speech synthesiser in the system is to increase the flexibility of the available test stimuli.

The basis for the synthesiser is the *Close Copy Speech* (CCS) synthesis or resynthesis method, in which it is the task of the synthesiser to "repeat utterances produced by a human speaker with a synthetic voice, while keeping the original prosody" (Dutoit, 1997). In this method, "close copy" means that the synthetic speech is as similar as possible to a human utterance. In fact, in the present

[1] The present development project is part of the Cochlear Implant Testing project led by Grażyna Demenko, and the M.A. thesis of Jolanta Bachan under the supervision of Grażyna Demenko and Dafydd Gibbon. Special thanks are due to Thorsten Trippel for the initial BLF2TextGrid conversion via the TASX XML format, and Arne Hellmich for suggestions for the BLF2MBROLA conversion.

context, "copy" means that the input to the synthesis engine for a given utterance is derived directly from a corresponding utterance in the annotated corpus data. The method can be taken a step further, in parametrising the prosody, so that modifications of the original prosody (speech timing and pitch patterns) can also be systematically introduced. For the purposes of this study, MBROLA, a *de facto* standard diphone synthesis engine with a suitably modular language-to-speech interface, was selected (Dutoit 1997).

In the present study, the definition by Dutoit is interpreted to mean that the Natural Language Processing or Text-To-Speech (TTS) component of the synthesiser is replaced by an analysis of a recorded speech signal. The analysis in the present context consists of a recorded speech signal, a method for pitch extraction from the speech signal, and an aligned phonemic annotation of the speech signal. The development procedure used in this study has three phases:

1. Manual Close Copy Speech (MCCS) synthesis: manual transfer of parameters from the original signals and annotations to the synthesiser interface.
2. Automatic Close Copy Speech (ACCS) synthesis: automatic transfer of parameters from the original signals and annotations, based on specifications derived from the MCCS phase.
3. Parametric Close Copy Speech (PCCS) synthesis: interactive and automatic parametrisation at the ACCS derived synthesiser interface.

This paper reports on the background to the development, and on the MCCS and ACCS synthesis phases of the development.

1.2 Context of the TTS development

The context of the present development is a project for testing the functionality of cochlear implants in children. The project strategy involves the development of tests supported by software, administration of the tests to normal children, children with hearing aids, and to children with cochlear implants. An overview of the context is shown in Figure 1; the individual components are needed for defining the use cases and the use case based requirements for the speech synthesiser.

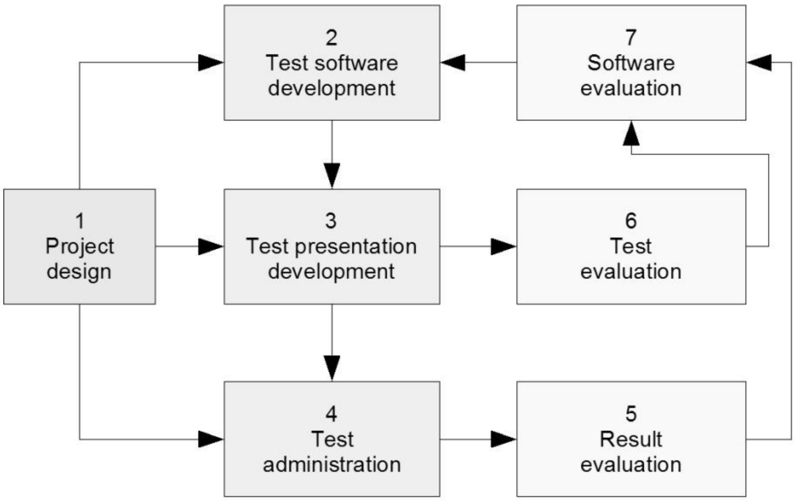

Figure 1: Project context for TTS software development

1.3 Overview of the paper

The present study is concerned with developing a Close Copy Speech synthesis subcomponent for component #2 shown in Figure 1. Evaluation feedback is expected from all other components. The components #3, #4, #5, #6 and #7 serve to define use cases for deployment of the TTS software; the main use cases considered are #3, test presentation development and #4, test administration.

The paper deals, first, with the project requirements and use cases which feed into the CCS synthesiser development; second, with system requirements; third, with the MCCS development phase; fourth, with the ACCS development phase; finally with a conclusion concerning the application and evaluation procedures.

2 Requirements: use cases

2.1 Use case: Test presentation development (component #3)

The battery of speech perception tests for children with a cochlear implant was created at Adam Mickiewicz University. In the project, linguists, phoneticians, graphics designers and computer programmers were involved. The tests were designed in close cooperation with experts from the Medical Academy, and audiologists from the Marke-med centre, both in Poznan. The tools for administering these tests contain two types of speech perception test:

1. *Nonsense tests*: tests with nonsense stimuli. Some of the tests in this set make use of synthesised stimuli. The aim of these tests is to assess whether the subject is able to take the verbal tests.
2. *Verbal tests*: tests with verbal stimuli.

Both sets of tests examine children's perceptive and linguistic skills making use of acoustic signals only. There are no visual cues in the test procedure, so the subject cannot lip-read. In both kinds of test the subject answers by pointing at a picture on a computer screen. The tests were designed for young children, and touch screens were provided for children who did not know how to use a computer mouse.

The tests with verbal stimuli are designed for children who are able to comprehend speech, but who may be unable to give verbal responses. In these tests six different voices were used to test intelligibility of different voice pitches. The tests make use of the following voices: two male adults, two female adults, one male child, one female child.

The results of the first series of tests in this use case indicated that more flexibility would be provided by more extensive use of a speech synthesiser of higher quality than currently available. This result provided part of the motivation for the development of a CCS synthesis system.

2.2 Use case: Test administration by perception testers (component #4)

The perception tests are designed for use by audiologists and speech therapists. They can be used by the audiologist in programming the cochlear implant, or by the speech therapist as an achievement test. The set of speech perception tests is also useful teaching material and it can be used by parents to help their children work on their perceptive skills. The standard graphical user interface will need to be extended by manipulation options for synthesised voices. Figure 2 shows the scenario of the tests. During the testing procedure three subjects are involved: the child, the tester and the computer. A parent's presence during the tests is optional.

In the first stage of the testing procedure the tester provides the subject with instructions. If the subject understands the instructions, the tester runs the tests and the testing material appears on the computer screen. If the subject cannot understand the instructions, the test is terminated. The computer provides acoustic stimuli for the child, the tester and (if present) the parent. Then the child responds to the stimuli by pointing at a picture visualising the acoustic stimuli. If the child does not know what the stimulus is, he or she asks the tester or the parent questions. In principle, the tester is not allowed to give hints, but, for the purpose of this preliminary research (evaluating the tests), the testers may help the children with the tests if necessary. Similarly, the parents are asked questions by their children, and despite the fact that in principle they are also not allowed to give help, it is understandable that the parents help their little children with answers, and this is currently permitted. This kind of cooperation between the child, the tester and/or the parent is one of the main complicating factors in assessing the structure of the tests and the dialogue between the child and the computer. All the responses given by the child to the computer are collected and the results of the test are available on the computer screen to the tester. Finally, the tester notes down the results for future processing.

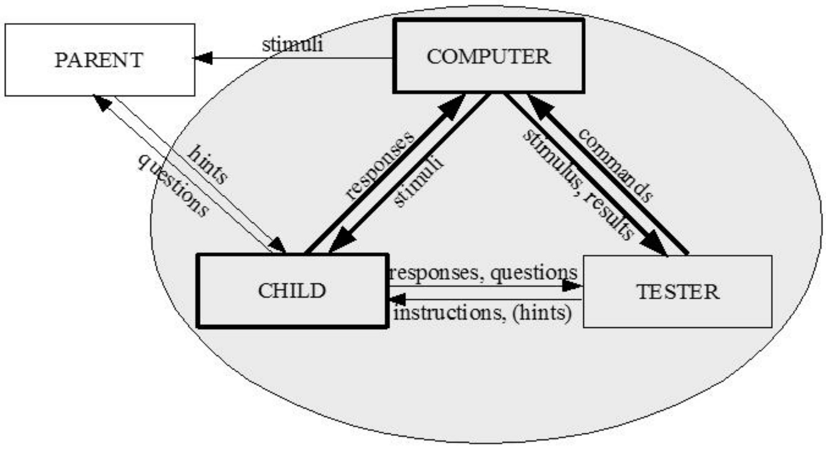

Figure 2: Test scenario showing communication relations between child, computer, tester and parent

2.3 Use case: Test evaluation (components #5, #6)

The set of speech perception tests was evaluated by students of linguistics at Adam Mickiewicz University. The evaluation of the tests started in September 2005 and went as follows:

1. In September and October 2005 the verification of the preliminary version of the set of the verbal tests and the set of tests with nonsense stimuli was carried out on Polish children with normal hearing. The two sets of tests were administered to 19 five-year-olds and 18 six-year-olds. The children's hearing was examined by audiologists. All the children had normal hearing and were normally developed.
2. In May and June 2006 the verification of the corrected and completed version of the set of verbal tests was conducted on Polish children with normal hearing. 14 four-year olds, 21 five-year-olds and 22 six-year-olds took part in the verification. The children’s hearing was examined beforehand by audiologists. All the children, except one four-year-old, had normal hearing and were normally developed.
3. In June and July 2006 the set of verbal tests was verified on children with hearing aids and children with cochlear implants:
    1. Two Polish children with hearing aids sat some of the tests. One of the children was seven years old, the other was twelve years old.
    2. A group of 15 Polish children with a cochlear implant took some of the tests. The children were at different ages. The youngest children were 2.5 years old, the oldest were 11 years old. All the children were prelingually hearing-impaired. Only one girl lost her hearing at the age of five after having acquired a good command of speech.

Results in these scenarios can be compared in order to determine which manipulations of prosodic parameters lead to the best test results. The effectiveness of the set of speech perception tests was evaluated qualitatively by fourth-year students of linguistics. In parallel to this, the tests were evaluated by audiologists. Note that the testers were concerned with evaluating the perception tests, not the actual cochlear implants. The focus of the research was on evaluation of the level of efficiency, ergonomics, motivation and suitability of the tests for the subject. The testers evaluated many parameters. The relevant parameters for CCS development are as follows:

1. the intelligibility of the instruction, picture and sound combinations used in the tests,
2. the dialogue between the child and the computer.

The problems discovered were:

1. Tests with nonsense stimuli:
    1. Synthesised stimuli in the set of tests with nonsense stimuli were of poor quality.
    2. The children had problems understanding the instructions to the tests with nonsense stimuli.
2. Tests with verbal stimuli:
    1. Some sounds were very difficult to recognise, because of the speaker's fast speech rate.
    2. The pitch of the female voice was too low.
    3. The accentuation was not prominent enough for the purpose of some tests.
    4. Some sounds were segmented incorrectly.
    5. Some sounds were missing.
    6. The dialogue between the testee and the computer needs improvement. Children sometimes did not know whether they gave a correct answer or not. They also looked at the testers or the parents for a sign of confirmation before giving the answer.
    7. If children with a cochlear implant could not understand the stimuli, they wanted to read the word from the testers' or their parents' lips.
    8. There is no test including stimuli presented in noise.

For discussion of these results, see Bachan (2006). The results provided a rather specific set of requirements for CCS development.

2.4 Use case: Software evaluation (component #7)

The task for the software evaluation use case is to coordinate evaluation results from other components in the form of recommendations to the software developers. In practice, evaluation results may go directly to the software developer, but in the ideal case the software evaluator will relate the evaluations to the original project goals before proposing software revisions and further development.

Based on the original project goals, some future directions for software development emerged:

1. Introduction of higher quality speech synthesis in order to correct the existing synthesised stimuli and make speech stimuli dynamic and flexible.
2. Addition of a calibrated test in noise, preferably using speech synthesis.
3. Design and implementation of a database.

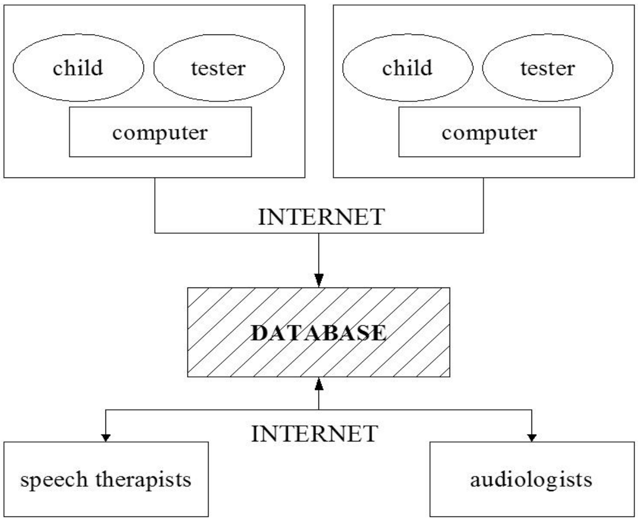

Figure 3: Networked database model with different access rights for different use cases

The workflow could be improved if the results were sent via the Internet and stored in a database on a server. The data could wait there for future processing by speech therapists and audiologists. Figure 3 presents a model of such a database and its use.

3 Requirements: system and resources

3.1 System requirements

The inputs required by the CCS synthesis system are as follows:

1. Source speech:
   1. source speech recordings,
   2. source DB: annotated speech database.
2. Speech synthesiser:
   1. diphone database,
   2. synthesis engine.
3. Parametrisations of close copy synthesis (not discussed in this paper).

The outputs to be produced by the CCS synthesis system are as follows:

1. Target pronunciation specification: specification table for input to speech synthesis engine.
2. Target acoustic output: produced by the speech synthesis engine.

An additional user interface for interacting with the system will be required for the PCCS synthesis system, but this is not discussed in further detail here. There are two sets of operations which users in one or more of the use cases will need in a user interface:

1. Duration warping: various linear or non-linear changes in the durations of phonemes in the utterance.
2. Frequency warping: various linear or non-linear changes in the frequency of whole utterances or parts of utterances such as focussed syllables or nuclear tones.
3. Database management and stimulus presentation.

3.2 Available resources: recordings

A corpus of recordings[2] for a male voice was available from a speech synthesis development scenario. The texts were spoken by a professional speaker and the recordings were made in a professional recording studio. The sampling rate of the data in the available format is 16kHz in a standard WAV format. The texts for use in the synthesiser development consisted initially of a selection of 1200 sentences from the corpus of approximately 4000 utterances.

3.3 Available resources: annotations

Annotation of the recordings at phoneme level was performed automatically using a program CreatSeg (Demenko & al. 2006) and checked by trained phoneticians. Phonemic segments which were not correctly handled by the automatic segmentator were manually edited. Additionally, the annotations also contain prosodic information, based partly on functional judgments and partly on prosodic information. The annotation uses the following information types:

1. Sample serial numbers (column 1).
2. Phonemic/allophonic label tier (column 2):
   1. Labels for 40 phonemes.
   2. Lexical stress types (Demenko, Grocholewski, Wagner, Szymanski 2006).
   3. Word and syllable boundaries.
   4. Four additional labels:
      1. [?] for a glottal stop,
      2. [#$p] for a pause,
      3. [#$j] for a segment such as a click or a sigh which is to be deleted for the purposes of speech synthesis. Moreover, if [$j] is added to the first segment of a word, the whole sentence is to be deleted for the speech synthesis purposes, e.g. [#m] means the first segment of a word, [#$jm] means the first segment of a word which is to be deleted.
      4. [/] for a syllable segment which is to be deleted for the purposes of speech synthesis.
3. Prosodic tier (column 3): Prosodic phrase boundary labels.

[2] The authors gratefully acknowledge the provision of this corpus by Grażyna Demenko (Principal Investigator of the Cochlear Implant Evaluation project).

Deleted items will be taken into consideration at a later stage. Table 1 shows a list of phoneme labels used in the annotation (Demenko, Wypych, Baranowska, 2003).

Table 1: SAMPA phoneme labels used in the corpus annotation

BLF Polish modified SAMPA	*Orthography*	*Phonemic transcription*	*BLF Polish modified SAMPA*	*Orthography*	*Phonemic transcription*
p	pik	pik	i	kit	kit
b	byt	byt	y	typ	typ
t	test	test	e	test	test
d	dym	dym	a	pat	pat
k	kat	kat	o	pot	pot
g	gen	gen	u	puk	puk
c	kiedy	cjedy	@ - *English schwa*		
J	giełda	Jjewda	m	mysz	myS
f	fan	fan	n	nasz	naS
v	wilk	vilk	n'	koń	kon'
s	syk	syk	N	pęk	peNk
z	zbir	zbir	l	luk	luk
S	szyk	Syk	r	ryk	ryk
Z	żyto	Zyto	w	łyk	wyk
s'	świt	s'fit	j	jak	jak
z'	źle	z'le	w~	ciąża	t^s'ow~Za
x	hymn	xymn	j~	więź	vjej~s'
t^s	cyk	t^syk			
d^z	dzwon	d^zvon			
t^S	czyn	t^Syn			
d^Z	dżem	d^Zem			
t^s'	ćma	t^s'ma			
d^z'	dźwig	d^z'vik			

The annotations are in the BOSS Label File (BLF) format, designed for the BOSS "Bonn Open Speech Synthesis" system. Table 2 shows the structure of the BLF annotation. The file represents a three column matrix, with sample numbers in the first column, an allophonic representation including word and syllable boundary allophones and lexical stress types in the second column, and a prosodic boundary representation in the third column. The use of sample numbers and not time stamps makes additional knowledge of sampling rate metadata necessary. The table represents the first part of the Polish sentence *Na szczęście myśl o przeprowadzce była tylko chwilowa i Gosia będzie nadal z nami mieszkać*. from the corpus.

Table 2: Fragment of BLF file input resource

Sample number (16 kHz rate)	*Segmental labels*	*Prosodic labels*
0	#$p	
5798	#n	-5,.
6863	a	
8008	#S	
9312	t^S	
10047	"e	
10880	j~	

Sample number (16 kHz rate)	*Segmental labels*	*Prosodic labels*
11351	.s'	
12640	t^s'	
13634	e	
14481	#$jm	
15613	|y	
16235	z'	
17214	l	
18843	#o	

In the phonemic/allophonic annotation label column, the following conventions are used:

1. [#] encodes the beginning of a word,
2. [.] encodes the beginning of a syllable (does not occur in this extract),
3. [#n] stands for a word-initial allophone of the phoneme /n/,
4. ["] denotes falling accent realised by F0 fall on postaccented syllable/syllables or F0 interval between accented and postaccented vowels,
5. ["e] stands for the accented allophone of the phoneme /e/ with falling accent,
6. [#$p] stands for a pause ([#$p] is always inserted at the beginning and at the end of a sentence and can also appear in the middle of a sentence),
7. label [#$jm] is read as
    1. [#m] - word-initial allophone of the phoneme /m/,
    2. [$j] - a segment not to be used for the speech synthesis; the whole word is ignored for the purposes of speech synthesis.

In the prosody label column, information about the type of utterance is represented:

1. [-5,.] indicates the beginning of a sentence with falling intonation,
2. [5,.] indicates the end of a sentence with falling intonation, e.g. a statement.

For further information cf. Demenko et al. (2006).

3.4 Available resources: diphone database

The diphone database used in the study is the PL1 MBROLA Polish female diphone database[3] created under the free database access terms of the MBROLA project. The diphone database consists of 1443 diphones and contains 37 phonemes in standard Polish SAMPA notation. All the phonemes are listed in Table 3.[4]

Table 3: Polish SAMPA transcription used in the PL1 Polish female MBROLA voice

PL1 Polish SAMPA	*Orthography*	*Phonemic transcription*	*PL1 Polish SAMPA*	*Orthography*	*Phonemic transcription*
p	pik	pik	i	kit	kit
b	bit	bit	I	typ	tIp
t	test	test	e	test	test
d	dym	dIm	a	pat	pat
k	kat	kat	o	pot	pot
g	gen	gen	u	puk	puk
f	fan	fan	e~	gęś	ge~s'
v	wilk	vilk	o~	wąs	vo~s
s	syk	sIk	m	mysz	mIS

[3] Created by Krzysztof Szklanny and Krzysztof Marasek, whose work we gratefully acknowledge.

[4] One of the differences between the PL1 MBROLA Polish female database phoneme set and the SAMPA Polish phoneme set used in the corpus annotation is that the former does not have /c/ and /J/ phonemes. However, these phonemes are not frequent in Polish, so there is no great data loss trying to replace them with phonemes available in the Polish diphone database. Table 6, later in the article, shows all the differences between both sets.

PL1 Polish SAMPA	*Orthography*	*Phonemic transcription*
z	zbir	zbir
S	szyk	SIk
Z	żyto	ZIto
s'	świt	s'fit
z'	źle	z'le
x	hymn	xImn
ts	cyk	tsIk
dz	dzwon	dzvon
tS	czyn	tSIn
dZ	dżem	dZem
ts'	ćma	ts'ma
dz'	dźwig	dz'vik
n	nasz	naS
n'	koń	kon'
N	pęk	peNk
l	luk	luk
r	ryk	rIk
w	łyk	wIk
j	jak	jak

4 Design: CCS architecture comparison

4.1 Text-To-Speech (TTS) synthesis

The standard components of regular text-to-speech synthesis are:

1. Natural Language Processing (NLP) module, which preprocesses and normalises an input text, produces phonetic transcription (phonetisation), together with a specification of prosodic features (pitch pattern, intensity and timing).
2. Digital Signal Processing (DSP) module, which transforms this data into speech, which may use uniform units such as diphones or corpus based weighted non-uniform unit selection.
3. Database of speech units such as diphones for the language in question.

The selected component MBROLA is a standard diphone synthesis engine: “MBROLA is a speech synthesiser based on the concatenation of diphones. It takes a list of phonemes as input, together with prosodic information (duration of phonemes and a piecewise linear description of pitch), and produces speech samples of 16 bits resolution (linear), at the sampling frequency of the diphone database used (it is therefore NOT a Text-To-Speech (TTS) synthesizer, since it does not accept raw text as input).”[5]

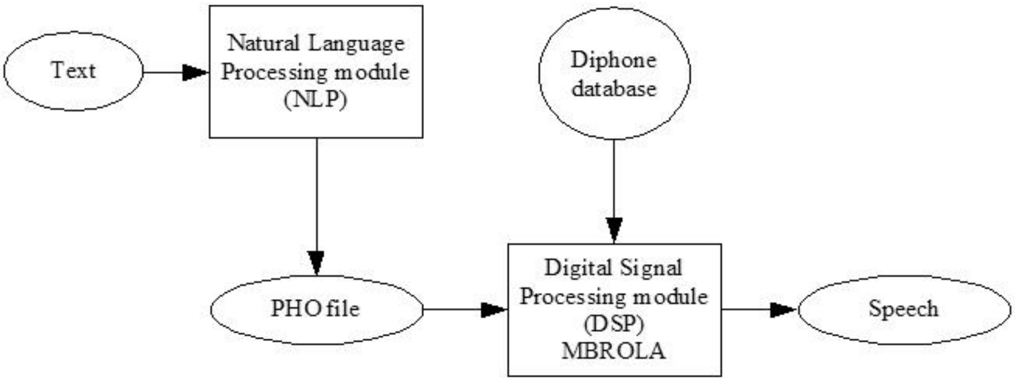
Text
Natural Language Processing module (NLP)
Diphone database
PHO file
Digital Signal Processing module (DSP) MBROLA
Speech

Figure 4: Text-To-Speech Synthesis with the MBROLA architecture

The MBROLA DSP component requires an input matrix containing phonemes, as well as specifications of duration and pitch modulation for each phoneme, but does not handle intensity

[5] MBROLA website, consulted 2006-11-30. We are grateful to the MBROLA team for this freeware application.

modulation of the output. Figure 4 shows the architecture of a standard TTS synthesis system with an MBROLA type synthesis engine. In Close Copy Speech synthesis, the NLP component is replaced with information from the speech corpus.

The architectures of manual and automatic close copy synthesis procedures are identical but for the conversion component. In the MCCS synthesis procedure, the information from the original speech signal is transferred to a spreadsheet, and the mapping operations from the recording (pitch extraction) and the annotations are performed manually. In the ACCS procedure, each of the manual operations is emulated by a software sub-component. The similar MCCS and ACCS synthesis architectures are shown in Figure 5. In the following sections, the design and implementation of the MCCS and ACCS systems are described.

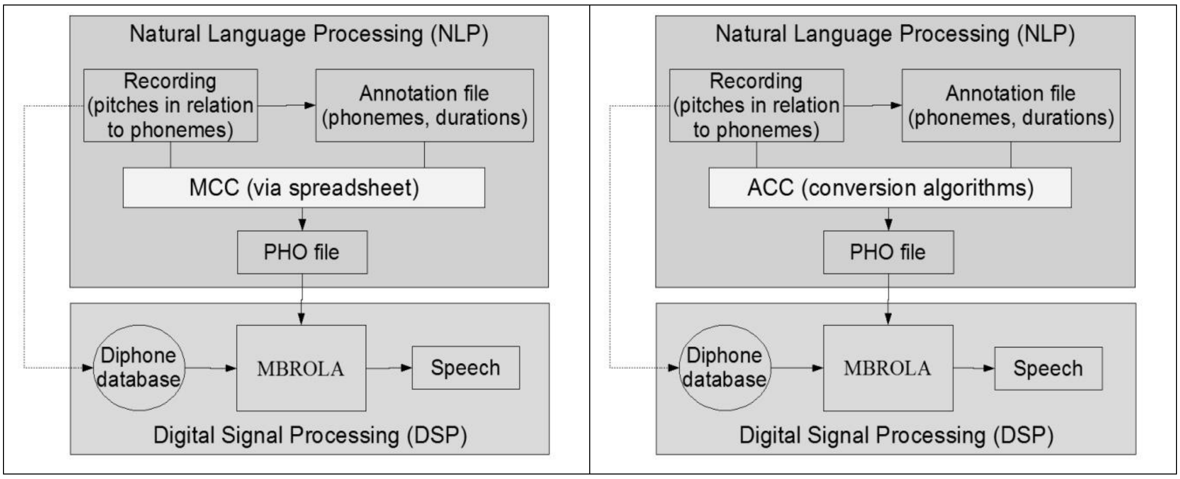
Natural Language Processing (NLP)
Recording (pitches in relation to phonemes)
Annotation file (phonemes, durations)
MCC (via spreadsheet)
PHO file
Diphone database
MBROLA
Speech
Digital Signal Processing (DSP)
Natural Language Processing (NLP)
Recording (pitches in relation to phonemes)
Annotation file (phonemes, durations)
ACC (conversion algorithms)
PHO file
Diphone database
MBROLA
Speech
Digital Signal Processing (DSP)

Figure 5: Schemata for similar architectures for Manual and Automatic Close Copy Synthesis

5 Design and implementation: MCCS synthesis

5.1 MCCS synthesis system design

Close Copy Speech (CCS) synthesis is produced by the speech synthesis engine which has to “repeat utterances produced by a human speaker with a synthetic voice, while keeping the original prosody” (Dutoit, 1997). For CCS, the standard MBROLA diphone synthesis architecture (Figure 4) is modified. The NLP component is replaced by an annotation file in which a transcription and a time stamp are aligned with the speech signal recording. The annotation and the recording together in principle include all the information which is needed for generating the specification table interface to the synthesis engine, which is normally produced by the NLP component. Consequently, in Close Copy Speech synthesis no input text is used. CCS synthesis makes use of recordings of real utterances and annotations derived from these recordings. In the annotation files, phonemes and their durations are stored. In the recordings, information about pitch in relation to the phonemes in the annotation files is found.

Manual Close Copy Speech (MCCS) synthesis with MBROLA is a process of manually creating pronunciation specification tables (PHO files), making use of recorded and annotated real utterances, and synthesising the pronunciation specification tables using an appropriate voice (diphone database). The voice may be created from the annotated utterances, in the ideal case, or may be an independently created voice, as in the case of the present study. The human copier therefore emulates the Natural Language Processing front end to a speech synthesis engine. The speech and annotation information is input by manual operations into the Manual Close Copy Speech (MCCS) synthesis procedure. The output of the MCCS synthesis procedure is a pronunciation specification table which, together with a diphone database, constitutes the input to the MBROLA synthesis engine. MBROLA is the engine which converts the specifcation table into speech using the diphone database (which may be created from the annotated recordings or taken from an external source). The acoustic output of MBROLA is a speech file in WAV format.

The modules required for the kind of resynthesis selected for the development project are based on the MBROLA diphone synthesis model, which has the following structure:

1. Natural Language Processing (in TTS; in CCS generation from annotated recordings):
    1. Phonetisation: grapheme-to-phoneme conversion.
    2. Prosody generation: text parser for duration lookup and pitch assignment.
2. Specification table ("PHO file") as interface.
3. Speech synthesis component:
    1. Diphone database.
    2. MBROLA engine.
4. Audio ("WAV file") output.

The central component for present purposes is the PHO file, i.e. the interface file which contains the pronunciation specification table produced by a TTS or CCS component, and used as input by the MBROLA engine to synthesise speech. The format specifies a table with three columns:

1. phonemes that are present in the sound to be produced,
2. duration of these phonemes,
3. pitch values represented by one or more pairs of numbers - the first number stands for the place of the pitch value in the phoneme, the second number is the pitch value itself.

The syntax of the specification table ST is defined as a sequence of one or more vectors SV, each with three components: the phoneme PH, the phoneme duration PD and the sequence of zero (for voiceless stretches) or more pitch pairs PP (in the prototype maximally one), consisting of pitch location PL and the pitch value PV:

```
<ST>   ::=  <SV>+
<SV>   ::=  <PH><PD><PP>*
<PP>   ::=  <PL><PV>
<PH>   ::=  sampa_phoneme1 | ... | sampa_phonemen
<PD>   ::=  millisecond_integer
<PL>   ::=  pitch_location_percent
<PV>   ::=  pitch_value_hertz
```

An illustration of the first five rows of the pronunciation specification table interface between the NLP and the DSP components is shown in Table 4; this example was derived from the corpus.

Table 4: Fragment of Specification Table (ST) for MBROLA PHO file

PH phoneme (PL1 SAMPA)	*PD phoneme duration (msec)*	*PP pitch pair*	
		PL pitch location (%)	*PV pitch value (hertz)*
n	66	50	200
a	72	50	210
S	82	50	240
tS	45	50	310
e~	29	50	306

5.2 Mismatches and format preprocessing

The specification table required by the MBROLA speech synthesis engine when used with the available Polish diphone database resource differs from the table provided by the resource. This incompatibility has several components, for which format conversion tools need to be specified. The incompatibilities are listed in Table 5.

Table 5: Polish annotation, diphone database and PHO file conventions

BLF annotation	*Diphone database*	*PHO file*
sample numbers	-	durations (msec)
positional allophones	phonemes	phonemes
BLF phoneme set	PL1 phoneme set	PL1 phoneme set
syllable boundaries	-	-
word boundary types	-	-
pauses	pauses	pauses
prosodic annotation	-	-

The missing boundaries and the stress markings are not usable in the current MBROLA configuration and are deleted, but will be considered at a later stage for prosody parametrisation. The SAMPA phoneme set and notation was given by the available diphone database in the MBROLA pre-processed input format, and differs from the phoneme set used in the corpus annotation. The correspondences are shown in Table 6.

Table 6: Mismatches between BLF and PL1 SAMPA

BLF SAMPA annotation labels	*PL1 SAMPA symbols*
p	p
b	b
t	t
d	d
k	k
g	g
c	-
J	-
f	f
v	v
s	s
z	z
S	S
Z	Z
s'	s'
z'	z'
x	x
t^s	ts
d^z	dz
t^S	tS
d^Z	dZ
t^s'	ts'
d^z'	dz'

BLF SAMPA annotation labels	*PL1 SAMPA symbols*
i	i
y	I
e	e
a	a
o	o
u	u
@ - *English schwa*	-
-	e~
-	o~
m	m
n	n
n'	n'
N	N
l	l
r	r
w	w
j	j
w~	-
j~	-

A further mismatch occurs between the BLF and MBROLA PHO formats for time specification. The BLF format includes sample numbers, while the MBROLA PHO format requires durations. In order to calculate durations, sampling rate metadata information (16 kHz) is required. The formula for bridging the gap is $(samplenumber_i - samplenumber_{i-1}) / samplingrate$.

Perhaps the most crucial mismatch lies in the discrepancy between the corpus, which is recorded using a male voice, and the diphone database, which is derived from a female voice.

This requires an *ad hoc* pitch re-adjustment. Currently the trivial formula $pitch_{female} = 2 * pitch_{male}$ is used, but parametrisations with more complex formulae incorporating a baseline are being developed. In the long term, an annotated corpus based on a female voice is required, as well as diphone databases based on male voices.

5.3 MCCS synthesis system implementation

The BLF and PHO specifications do not match, as already indicated in the design specification. Nevertheless, the required information is implicit in the annotation file, and the annotation file may be mined for this information. For the purpose of the MCCS procedure, a spreadsheet was designed in order to convert BLF files into PHO files. In order to map the BLF format into the spreadsheet table, a pre-processing step is necessary: the three different columns (the sample numbers, annotation labels and phrase intonation labels are placed in a CSV-formatted file which is opened as a spreadsheet table. The spreadsheet software used is OpenOffice Calc. The further steps required for conversion into the PHO format are detailed in Table 7 and described below.

Table 7: Spreadsheet for BLF to PHO format conversion

	Samples	BOSS 2006	Prosody	msec = Samples / 16	Duration msec	Polish Voice SAMPA	Duration rounded	Pitch Location: 33%	Pitch Value: 33%	Pitch Location: 66%	Pitch Value: 66%	Original Pitch Value – male: 33%	Original Pitch Value – male: 66%
1	0	#$p		0	70	_	70						
2	1120	#v	-5,.	70	95.44	v	95	33	174	66	226	87	113
3	2647	y		165.44	64.56	I	65	33	218	66	194	109	97
4	3680	.g		230	50	g	50	33	188	66	228	94	114
5	4480	l		280	36.63	l	37	33	234	66	252	117	126
6	5066	`o		316.63	83.38	o	83	33	262	66	284	131	142
7	6400	n		400	46.81	n	47	33	286	66	268	143	134
8	7149	.d		446.81	31.88	d	32	33	250	66	270	125	135
9	7659	a		478.69	51.31	a	51	33	256	66	236	128	118
10	8480	Z		530	122.94	Z	123	33	196	66	186	98	93
11	10447	#d^z'		652.94	67.06	dz'	67	33	182	66	220	91	110
12	11520	i		720	30	I	30	33	226	66	228	113	114
13	12000	.s'		750	120	s'	120	33	230	66	260	115	130
14	13920	a		870	60	a	60	33	238	66	216	119	108
15	14880	j		930	30	j	30	33	210	66	212	105	106
16	15360	#j		960	30	j	30	33	212	66	210	106	105
17	15840	a		990	80	a	80	33	208	66	186	104	93
18	17120	g		1070	62.94	g	63	33	210	66	204	105	102
19	18127	#Z	5,.	1132.94	93.56	Z	94	33	200	66	200	100	100
20	19624	&a		1226.5	123.5	a	124	33	180	66	138	90	69
21	21600	./b		1350	62.94	b	63	33	136	66	184	68	92
22	22607	a		1412.94	167.06	a	167	33	148	66	146	74	73
23	25280	#$p		1580									

The columns in Table 7 contain the following information:

1. Running BLF label indices (not in original BLF format).
2. BLF format: sample number.
3. BLF format: phonemic/allophonic annotation labels.
4. BLF format: prosodic labels.
5. Conversion of sample numbers to time-stamps (msec): division of sample numbers by sampling rate (16kHz), i.e. *sample-number*/16.
6. Phoneme durations: $duration(cell_i)$ - $duration(cell_{i-1})$, i.e. the value of the preceding cell is subtracted from the value in each cell.
7. PHO format: Polish Voice SAMPA phoneme notation, with BLF characters replaced by Polish Voice SAMPA characters.
8. PHO format: rounded (integer) phoneme duration values.
9. PHO format: Pitch Location: at 33% of phoneme length.
10. PHO format: Pitch Value in Hertz (*ad hoc* adaptation to female Polish Voice: multiplication by 2).

11. PHO format: Pitch Location: at 66% of phoneme length.
12. PHO format: Pitch Value in Hertz (*ad hoc* adaptation to female Polish Voice: multiplication by 2).
13. Original male voice pitch values at 33% (extracted manually using WaveSurfer software).
14. Original male voice pitch values at 66% (extracted manually using WaveSurfer software).

6 Design and implementation: ACCS synthesis

6.1 ACCS synthesis system design

The Automatic Close Copy Speech (ACCS) synthesis version of the synthesiser is the real-time capable speech synthesis component which is intended to be integrated into the project architecture for the test dialogue which was discussed in the requirements section. An MCCS synthesiser is clearly unsuitable for this purpose: it is only suitable for offline work in requirements development, not for the use cases outlined on the basis of the project design. The design considerations for Automatic Close Copy Speech synthesis have already been detailed in the discussion of the MCCS system, and the task of the ACCS system in the current context is to emulate the MCCS system.

Automatic Close Copy Speech synthesis is conceptually similar to Manual Close Copy Speech synthesis, except that the three main conversion steps are emulated by format conversion algorithms. Informal evaluation procedures (same/different and ABX judgments) are used to judge the success of the emulation.

6.2 ACCS synthesis system implementation

Most of the extraction and conversion tasks performed by the MCCS system are essentially the kind of scalar processing operation for which scripting languages were designed. For this reason, Perl, a ubiquitous scripting language, was selected as the main programming language for the ACCS system. The steps implemented in Perl are:

1. phoneme notation conversion for the Polish Voice, implemented as a substitution table;
2. conversion of sample numbers into the durations in milliseconds required by MBROLA.

However, the extraction of pitch patterns is delegated to a Praat script which takes a Praat annotation file (TextGrid file) derived from the original annotation format, and its matching recording as a WAV file, and extracts a pitch function for each segment on a specified tier in the recording. The integration of this information into the PHO file is then performed by a Perl script.

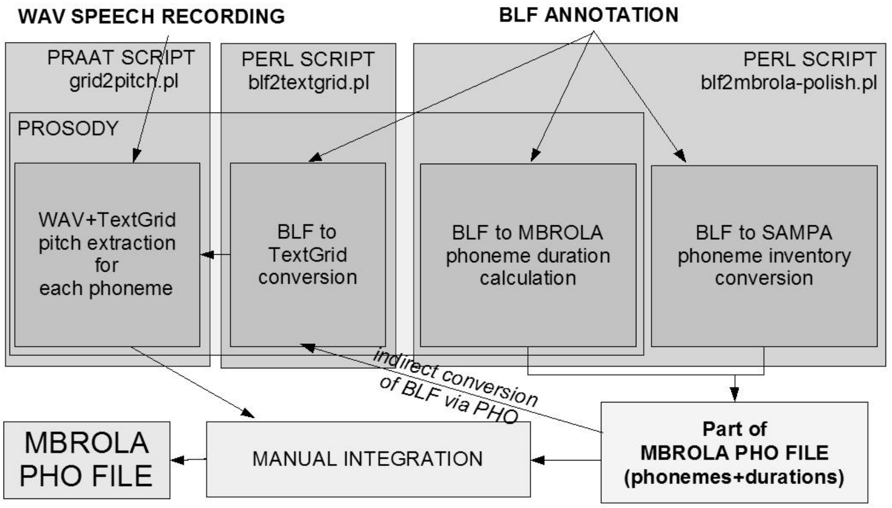
WAV SPEECH RECORDING
BLF ANNOTATION
PRAAT SCRIPT
grid2pitch.pl
PERL SCRIPT
blf2textgrid.pl
PERL SCRIPT
blf2mbrola-polish.pl
PROSODY
WAV+TextGrid pitch extraction for each phoneme
BLF to TextGrid conversion
BLF to MBROLA phoneme duration calculation
BLF to SAMPA phoneme inventory conversion
indirect conversion of BLF via PHO
MBROLA PHO FILE
MANUAL INTEGRATION
Part of MBROLA PHO FILE (phonemes+durations)

Figure 6: Detailed schema of Automatic Close Conversion Speech Resynthesis.

The overall implementation architecture is shown in Figure 6. Since the Praat script requires a different format (TextGrid), the BLF sample number and phoneme notation were converted into both MBROLA PHO and Praat TextGrid formats. Three options were considered for this:

1. Conversion of BLF directly into TextGrid notation.
2. Conversion of MBROLA notation into TextGrid notation, i.e. indirect conversion of BLF into TextGrid format, because BLF already had to be converted into MBROLA format.
3. Conversion of BLF notation into a generic XML notation (TASX, cf. Gut & Milde 2003), for which a library of functions, including TASX to Praat, already exist.

Both option 2 and option 3 were implemented. The implementation of option 2 is straightforward. The implementation of option 3 is more re-usable, but depends on an additional Java environment and the Saxon XML engine, and is therefore more complicated than necessary for the prototype.

7 Summary, evaluation and outlook

In this paper, the development of a speech synthesis component for use in speech perception tests for cochlear implants in children was described and a prototype implementation was developed. Use cases were outlined, and requirements derived from these use cases which, together with an overview of available resources, were employed in specifying the system design and in outlining future developments. The development procedure described here covered the first two of the three planned development stages, omitting the third Parametric CCS stage:

1. MCCS synthesis: manual format conversion from empirical data (speech recordings and time-aligned annotations) into the synthesis engine (MBROLA) interface format.
2. ACCS synthesis: automatic format conversion which emulates the manual format conversion procedure, using additional interface formats.

The MCCS procedure was developed as a best case gold standard for speech synthesis with MBROLA, against which future developments would be measured. The ACCS procedure was evaluated against this benchmark, with results which were, while not numerically identical with the MCCS procedure (due to differences in the pitch extraction procedure), indistinguishable from it in informal perception tests. Detailed evaluation procedures (Gibbon 1997; Gibbon 2000) were not used at this stage. However, informal pilot evaluation of ACCS intelligibility and naturalness was performed with Polish adults and children, with the result that very long utterances were less well understood by children (perhaps due to cognitive development factors).

Future work based on the project reported here includes the development of the third stage, a Parametric Close Copy Speech (PCCS) synthesis procedure, with the ACCS procedure as the platform for parametrising the prosodic features. Work on the PCCS procedure is in progress.

Bibliography

Bachan, J. 2006. Verification of a Set of Speech Perception Tests for Children with a Cochlear Implant. Speech signal annotation, processing and synthesis, [in:] *Proceedings of Speech Signal Annotation, Processing and Synthesis Symposium*, Poznań, September 2006.

Boersma, P. and D. Weenink. 2001. PRAAT, a system for doing phonetics by computer. Glot International 5(9/10): 341-345.

Demenko, G. & Wypych, M. & Baranowska, E. 2003. Implementation of Grapheme-to-Phoneme Rules and Extended SAMPA Alphabet in Polish Text-to-Speech Synthesis. *Speech and Language Technology* Vol. 7. Poznań: Zakład Graficzny UAM.

Demenko, G. & Grocholewski, S. & Wagner, A. & Szymanski M. 2006. Prosody annotation for corpus based speech synthesis. [in:] *Proceedings of the Eleventh Australasian International Conference on Speech Science and Technology*. Auckland, New Zealand.

Dutoit, T. 1997. *An Introduction To Text-To-Speech Synthesis*. Dordrecht: Kluwer Academic Publishers.

Dutoit, T. 2005. The MBROLA project. <http://www.tcts.fpms.ac.be/synthesis/mbrola.html>, accessed 2006-11-30.

Gibbon, D. & Moore, R. & Winski, R. 1997. *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter.

Gibbon, D. & Mertins, I. & Moore, R. 2000. *Handbook of Multimodal and Spoken Dialogue Systems: Terminology, Resources and Product Evaluation*. New York: Kluwer Academic Publishers.

Gut, U. & Milde, J-T. 2003. Annotation and Analysis of Conversational Gestures in the TASX environment. *Künstliche Intelligenz* 17:4.

Szklanny, K. & Masarek, K. 2002. PL1 - A Polish female voice for the MBROLA synthesizer. Copying the MBROLA Bin and Databases. <http://tcts.fpms.ac.be/synthesis/mbrola/mbrcopybin.html>, accessed 2006-11-25.

Sjölader, Kåre & Jonas Beskow. 2005. WaveSurfer 1.8.5/0511011429 © 2005.