

# Das Sprachsynthesystem Espeak

Janine Wolf

20.05.2008

## Ein paar Grundlagen: Sprachsynthese

### Generelles

Kleiner Exkurs: Concept-to-Speech

### Geschichte

### Einsatzorte

### Vorverarbeitung

Normalisierung des Textes

Phonetische Analyse

Prosodische Analyse

### Synthese

Synthese mittels "Audioschnipsel"

Signalerzeugung

## eSpeak

### Generelles

### Funktionsweise

eSpeakEdit - Der Editor zu eSpeak

Generelles

## Quellen

## Sprachsynthese

- ▶ Erzeugen von Sprache (akustisches Signal) aus...

## Sprachsynthese

- ▶ Erzeugen von Sprache (akustisches Signal) aus...
  - ▶ Textinput (Text-to-Speech)

## Sprachsynthese

- ▶ Erzeugen von Sprache (akustisches Signal) aus...
  - ▶ Textinput (Text-to-Speech)
  - ▶ Konzepten (Concept-to-Speech)

## Kleiner Exkurs: Concept-to-Speech

- ▶ verwendet in manchen Dialogsystemen
- ▶ Input: maschinengenerierte Nachricht
- ▶ Beispiel: ILEX (virtueller Museumsführer, NLG-System)

# TTS und CTS

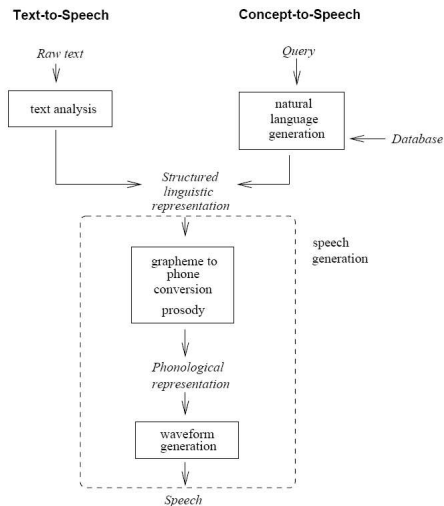


Figure 1. Text to speech and Concept to speech

# Ein sehr kurzer geschichtlicher Abriss

## Ein sehr kurzer geschichtlicher Abriss

- ▶ 1779: Christian Kratzenstein (dän. Wissenschaftler) baut ein Modell des menschlichen Vokaltrakts, der die Vokale produzieren kann

## Ein sehr kurzer geschichtlicher Abriss

- ▶ 1779: Christian Kratzenstein (dän. Wissenschaftler) baut ein Modell des menschlichen Vokaltrakts, der die Vokale produzieren kann
- ▶ 1791: Wolfgang von Kempelen entwickelt "Sprechmaschine" - Nachbildung der Sprachproduktion mittels Hebeln unterschiedlich spannbaren Stimmbändern und Mund (Gummi), Lunge (Blasebalg), Ledertubus (Geometrie und Resonanzverhalten des Vokaltrakts), Pfeifen (Frikative); umschnallbar und bedienbar mit den Händen

## Ein sehr kurzer geschichtlicher Abriss

- ▶ 1779: Christian Kratzenstein (dän. Wissenschaftler) baut ein Modell des menschlichen Vokaltrakts, der die Vokale produzieren kann
- ▶ 1791: Wolfgang von Kempelen entwickelt "Sprechmaschine" - Nachbildung der Sprachproduktion mittels Hebeln unterschiedlich spannbaren Stimmbändern und Mund (Gummi), Lunge (Blasebalg), Ledertubus (Geometrie und Resonanzverhalten des Vokaltrakts), Pfeifen (Frikative); umschnallbar und bedienbar mit den Händen
- ▶ 1873: Charles Wheatstone baut eine "Speaking Machine" beruhend auf Kempelen (1791)

## Ein sehr kurzer geschichtlicher Abriss

- ▶ 1779: Christian Kratzenstein (dän. Wissenschaftler) baut ein Modell des menschlichen Vokaltrakts, der die Vokale produzieren kann
- ▶ 1791: Wolfgang von Kempelen entwickelt "Sprechmaschine" - Nachbildung der Sprachproduktion mittels Hebeln unterschiedlich spannbaren Stimmbändern und Mund (Gummi), Lunge (Blasebalg), Ledertubus (Geometrie und Resonanzverhalten des Vokaltrakts), Pfeifen (Frikative); umschnallbar und bedienbar mit den Händen
- ▶ 1873: Charles Wheatstone baut eine "Speaking Machine" beruhend auf Kempelen (1791)
- ▶ 1930: BellLabs baut "Vocoder" - tastaturgesteuerter, elektronischer Sprachsynthesizer

## Ein sehr kurzer geschichtlicher Abriss

- ▶ 1779: Christian Kratzenstein (dän. Wissenschaftler) baut ein Modell des menschlichen Vokaltrakts, der die Vokale produzieren kann
- ▶ 1791: Wolfgang von Kempelen entwickelt "Sprechmaschine" - Nachbildung der Sprachproduktion mittels Hebeln unterschiedlich spannbaren Stimmbändern und Mund (Gummi), Lunge (Blasebalg), Ledertubus (Geometrie und Resonanzverhalten des Vokaltrakts), Pfeifen (Frikative); umschnallbar und bedienbar mit den Händen
- ▶ 1873: Charles Wheatstone baut eine "Speaking Machine" beruhend auf Kempelen (1791)
- ▶ 1930: BellLabs baut "Vocoder" - tastaturgesteuerter, elektronischer Sprachsynthesizer
- ▶ 1968: erstes komplettes TTS-System wird fertiggestellt

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte
- ▶ Sprache in Videospiele/Spielzeug

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte
- ▶ Sprache in Videospielen/Spielzeug
- ▶ Aufzeichnung von schriftlichen Vorträgen in Sprachform

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte
- ▶ Sprache in Videospielen/Spielzeug
- ▶ Aufzeichnung von schriftlichen Vorträgen in Sprachform
- ▶ Erzeugen von Podcasts/Audioblogs

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte
- ▶ Sprache in Videospielen/Spielzeug
- ▶ Aufzeichnung von schriftlichen Vorträgen in Sprachform
- ▶ Erzeugen von Podcasts/Audioblogs
- ▶ Erzeugen von Tonbandansagen (z.B. am Telefon)

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte
- ▶ Sprache in Videospielen/Spielzeug
- ▶ Aufzeichnung von schriftlichen Vorträgen in Sprachform
- ▶ Erzeugen von Podcasts/Audioblogs
- ▶ Erzeugen von Tonbandansagen (z.B. am Telefon)
- ▶ Dialogsysteme (!)

# Einsatzorte für Sprachsynthese

- ▶ Vorlesen von Texten für Sehbehinderte
- ▶ Sprache in Videospielen/Spielzeug
- ▶ Aufzeichnung von schriftlichen Vorträgen in Sprachform
- ▶ Erzeugen von Podcasts/Audioblogs
- ▶ Erzeugen von Tonbandansagen (z.B. am Telefon)
- ▶ Dialogsysteme (!)
- ▶ ...

# Beispielprogramme

- ▶ Festival/FestVox
- ▶ FreeTTS
- ▶ MBROLA
- ▶ Logox

# Vorverarbeitung

- ▶ Überführen des Textes in eine interne, phonemische Repräsentation

# Vorverarbeitung

- ▶ Überführen des Textes in eine interne, phonemische Repräsentation
- ▶ 1. Normalisierung des Textes (Text für Schritt 2 vorbereiten)
- ▶ 2. Phonetische Analyse (Text in Phoneme überführen)
- ▶ 3. Prosodische Analyse (Prosodie bestimmen)

# Vorverarbeitung

- ▶ Überführen des Textes in eine interne, phonemische Repräsentation
- ▶ 1. Normalisierung des Textes (Text für Schritt 2 vorbereiten)
- ▶ 2. Phonetische Analyse (Text in Phoneme überführen)
- ▶ 3. Prosodische Analyse (Prosodie bestimmen)

# Normalisierung des Textes

## ► Satztokenisierung

# Normalisierung des Textes

- ▶ Satztokenisierung
- ▶ Non-standard words

# Normalisierung des Textes

- ▶ Satztokenisierung
- ▶ Non-standard words
  - ▶ Abkürzungen (z.B. Jan 1)

# Normalisierung des Textes

- ▶ Satztokenisierung
- ▶ Non-standard words
  - ▶ Abkürzungen (z.B. Jan 1)
  - ▶ Buchstabenfolgen (z.B. IBM, DVD)

# Normalisierung des Textes

- ▶ Satztokenisierung
- ▶ Non-standard words
  - ▶ Abkürzungen (z.B. Jan 1)
  - ▶ Buchstabenfolgen (z.B. IBM, DVD)
  - ▶ Zahlen und Daten (z.B. 1689, 01.06.08)

# Normalisierung des Textes

- ▶ Satztokenisierung
- ▶ Non-standard words
  - ▶ Abkürzungen (z.B. Jan 1)
  - ▶ Buchstabenfolgen (z.B. IBM, DVD)
  - ▶ Zahlen und Daten (z.B. 1689, 01.06.08)
  - ▶ Akronyme (z.B. IKEA, MoMa, NASA)

# Normalisierung des Textes

- ▶ Satztokenisierung
- ▶ Non-standard words
  - ▶ Abkürzungen (z.B. Jan 1)
  - ▶ Buchstabenfolgen (z.B. IBM, DVD)
  - ▶ Zahlen und Daten (z.B. 1689, 01.06.08)
  - ▶ Akronyme (z.B. IKEA, MoMa, NASA)
- ▶ Disambiguierung von Homographen (z.B. to live near a zoo with live animals)

# Normalisierung des Textes

- ▶ Satztokenisierung:
  - ▶ Beachtung von Interpunktion
  - ▶ Sprachspezifika, z.B. Großschreibung am Satzanfang
  - ▶ Worttokenisierung und Einbeziehung von Kontext
  - ▶ wenn das Korpus groß genug ist, auch Wahrscheinlichkeiten
  - ▶ Maschinelles Lernen

# Normalisierung des Textes

- ▶ Satztokenisierung:
  - ▶ Beachtung von Interpunktion
  - ▶ Sprachspezifika, z.B. Großschreibung am Satzanfang
  - ▶ Worttokenisierung und Einbeziehung von Kontext
  - ▶ wenn das Korpus groß genug ist, auch Wahrscheinlichkeiten
  - ▶ Maschinelles Lernen
- ▶ Non-standard words:
  - ▶ Maschinelles Lernen mittels Regeln
  - ▶ Reguläre Ausdrücke
  - ▶ direkte Graphem-Phonem-Übersetzung mittels Regeln

# Normalisierung des Textes

- ▶ Satztokenisierung:
  - ▶ Beachtung von Interpunktion
  - ▶ Sprachspezifika, z.B. Großschreibung am Satzanfang
  - ▶ Worttokenisierung und Einbeziehung von Kontext
  - ▶ wenn das Korpus groß genug ist, auch Wahrscheinlichkeiten
  - ▶ Maschinelles Lernen
- ▶ Non-standard words:
  - ▶ Maschinelles Lernen mittels Regeln
  - ▶ Reguläre Ausdrücke
  - ▶ direkte Graphem-Phonem-Übersetzung mittels Regeln
- ▶ Disambiguierung von Homographen:
  - ▶ Einsatz von POS-Taggern

# Phonetische Analyse

Festlegen der Aussprache im normalisierten Text mittels  
Aussprachewörterbuch

# Phonetische Analyse

Festlegen der Aussprache im normalisierten Text mittels  
Aussprachewörterbuch

- ▶ Beachte aber:
  - ▶ Eigennamen (morph. Dekomposition, analoge Formation, unbekannte Namen auf bereits bekannte Aussprachen mappen...)

# Phonetische Analyse

Festlegen der Aussprache im normalisierten Text mittels  
Aussprachewörterbuch

- ▶ Beachte aber:
  - ▶ Eigennamen (morph. Dekomposition, analoge Formation, unbekannte Namen auf bereits bekannte Aussprachen mappen...)
  - ▶ Andere, nicht verzeichnete Wörter (Graphem-Phonem-Übersetzung nach phonotaktischen Regeln der Sprache, Maschinelles Lernen)

# Prosodische Analyse

- ▶ Intonation und Rhythmus
- ▶ Mischung aus Satzanalyse und verschiedenen Heuristiken (z.B. Absenkung/Anhebung der Satzmelodie bei Aussagen/Fragen)

# Prosodische Analyse

- ▶ Intonation und Rhythmus
- ▶ Mischung aus Satzanalyse und verschiedenen Heuristiken (z.B. Absenkung/Anhebung der Satzmelodie bei Aussagen/Fragen)
- ▶ Akzentuierung (mit Wahrscheinlichkeiten, die N-Gramme benutzen)

# Prosodische Analyse

- ▶ Intonation und Rhythmus
- ▶ Mischung aus Satzanalyse und verschiedenen Heuristiken (z.B. Absenkung/Anhebung der Satzmelodie bei Aussagen/Fragen)
- ▶ Akzentuierung (mit Wahrscheinlichkeiten, die N-Gramme benutzen)
- ▶ aber: Fokus, Absenken/Erhöhen der Stimmfrequenz je nach Kontext ist nicht zuverlässig bestimmbar
  - ▶ daher: Nutzen von SSML (Speech Synthesis Markup Language)

# Synthese

- ▶ Überführen der Repräsentation ins akustische Signal
- ▶ Verwendung von Datenbanken mit charakteristischen Informationen über Laute/Sprachsegmente oder aufgenommenen Sprachsequenzen/Lauten

# Synthese

- ▶ Überführen der Repräsentation ins akustische Signal
- ▶ Verwendung von Datenbanken mit charakteristischen Informationen über Laute/Sprachsegmente oder aufgenommenen Sprachsequenzen/Lauten

## Arten der Synthese

- ▶ Synthese mittels “Audioschnipsel”
- ▶ Signalerzeugung

# Synthese mittels “Audioschnipsel”

- ▶ im voraus aufgenommene Audioschnipsel werden aneinandergesetzt

## Unterformen dieser Syntheseart:

- ▶ Diphonsynthese
- ▶ Unit Selection

# Diphonsynthese

- ▶ Äußerungen in der Datenbank werden in Laut-Zweiergruppen geteilt

# Diphonsynthese

- ▶ Äußerungen in der Datenbank werden in Laut-Zweiergruppen geteilt
- ▶ Konkatenation von Diphonen (unter Beachtung der Grenzübergänge)

# Diphonsynthese

- ▶ Äußerungen in der Datenbank werden in Laut-Zweiergruppen geteilt
- ▶ Konkatenation von Diphonen (unter Beachtung der Grenzübergänge)
- ▶ Anpassen der Prosodie (F0, Dauer)

# Diphonsynthese

- ▶ Äußerungen in der Datenbank werden in Laut-Zweiergruppen geteilt
- ▶ Konkatenation von Diphonen (unter Beachtung der Grenzübergänge)
- ▶ Anpassen der Prosodie (F0, Dauer)

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Stellen wir uns vor...
  - ▶ die Textanalyse ist abgeschlossen.

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Stellen wir uns vor...
  - ▶ die Textanalyse ist abgeschlossen.
  - ▶ wir kennen die nötigen Diphone und prosodischen Anforderungen.

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Stellen wir uns vor...
  - ▶ die Textanalyse ist abgeschlossen.
  - ▶ wir kennen die nötigen Diphone und prosodischen Anforderungen.
  - ▶ passende Diphonsequenzen sind bereits gefunden und konkateniert.

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Stellen wir uns vor...
  - ▶ die Textanalyse ist abgeschlossen.
  - ▶ wir kennen die nötigen Diphone und prosodischen Anforderungen.
  - ▶ passende Diphonsequenzen sind bereits gefunden und konkateniert.
- ▶ nächster Schritt:
  - ▶ Anpassen der Prosodie (Pitch, Energie, Dauer), um den Anforderungen der internen Repräsentation gerecht zu werden
  - ▶ Erfüllen der prosodischen Anforderungen

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Algorithmus zur Anpassung von Grundfrequenz und Dauer

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Algorithmus zur Anpassung von Grundfrequenz und Dauer
- ▶ Diphonsequenz hat verschiedene (markierte) Perioden mit einer bestimmten Grundfrequenz (Pitch)

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Algorithmus zur Anpassung von Grundfrequenz und Dauer
- ▶ Diphonsequenz hat verschiedene (markierte) Perioden mit einer bestimmten Grundfrequenz (Pitch)
- ▶ Ausschneiden der Perioden und bestimmtem Umfeld (mittels Fensterfunktion)
- ▶ Hinzuaddieren der Perioden an passender Stelle im zu synthetisierenden Material

## Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Algorithmus zur Anpassung von Grundfrequenz und Dauer
- ▶ Diphonsequenz hat verschiedene (markierte) Perioden mit einer bestimmten Grundfrequenz (Pitch)
- ▶ Ausschneiden der Perioden und bestimmtem Umfeld (mittels Fensterfunktion)
- ▶ Hinzuaddieren der Perioden an passender Stelle im zu synthetisierenden Material
- ▶ Anpassung der Grundfrequenz: dichter/weniger dicht zusammenfügen als im Original

# Pitch-Synchronous Overlap-and-Add (PSOLA)

- ▶ Algorithmus zur Anpassung von Grundfrequenz und Dauer
- ▶ Diphonsequenz hat verschiedene (markierte) Perioden mit einer bestimmten Grundfrequenz (Pitch)
- ▶ Ausschneiden der Perioden und bestimmtem Umfeld (mittels Fensterfunktion)
- ▶ Hinzuaddieren der Perioden an passender Stelle im zu synthetisierenden Material
- ▶ Anpassung der Grundfrequenz: dichter/weniger dicht zusammenfügen als im Original
- ▶ Anpassung der Lautdauer: Perioden auslassen oder doppelt ausgeben

## Aber zurück zur allgemeinen Diphonsynthese

- ▶ gut, denn: fängt das Phänomen der Koartikulation ab

## Aber zurück zur allgemeinen Diphonsynthese

- ▶ gut, denn: fängt das Phänomen der Koartikulation ab
- ▶ nicht so gut:
  - ▶ Anpassung der Grenzen zwischen konkatenierten Diphonen kann ein unnatürliches Ergebnis produzieren
  - ▶ Phänomene wie Koartikulation können auch globaler auftreten, als nur zwischen benachbarten Lauten (entferntere Laute, Silbenstruktur...)

# Unit Selection

- ▶ Generalisierung der Diphonsynthese

# Unit Selection

- ▶ Generalisierung der Diphonsynthese
- ▶ verwenden Sprachdatenbank, in der jede Äußerung in kleinere Einheiten segmentiert wird (z.B. Silben, Morpheme, Phoneme, Sätze...)

# Unit Selection

- ▶ Generalisierung der Diphonsynthese
- ▶ verwenden Sprachdatenbank, in der jede Äußerung in kleinere Einheiten segmentiert wird (z.B. Silben, Morpheme, Phoneme, Sätze...)

## Unterschiede zur Diphonsynthese

- ▶ Diphone kommen in der Datenbank nicht genau einmal, sondern mehrmals vor
- ▶ minimale prosodische Bearbeitung der konkatenierten Einheiten nötig

# Unit Selection

- ▶ ganze Wörter oder Phrasen könnten in der Datenbank schon vorhanden sein und bräuchten keine Bearbeitung (natürliches Ergebnis!)

# Unit Selection

- ▶ ganze Wörter oder Phrasen könnten in der Datenbank schon vorhanden sein und bräuchten keine Bearbeitung (natürliches Ergebnis!)
- ▶ viele Kopien der gleichen Diphone in verschiedenen Kontexten machen es leichter, ein natürlich klingendes zu finden, wenn es gebraucht wird

# Unit Selection

- ▶ ganze Wörter oder Phrasen könnten in der Datenbank schon vorhanden sein und bräuchten keine Bearbeitung (natürliches Ergebnis!)
- ▶ viele Kopien der gleichen Diphone in verschiedenen Kontexten machen es leichter, ein natürlich klingendes zu finden, wenn es gebraucht wird
- ▶ realisiert mittels Kostenfunktion

# Unit Selection

- ▶ ganze Wörter oder Phrasen könnten in der Datenbank schon vorhanden sein und bräuchten keine Bearbeitung (natürliches Ergebnis!)
- ▶ viele Kopien der gleichen Diphone in verschiedenen Kontexten machen es leichter, ein natürlich klingendes zu finden, wenn es gebraucht wird
- ▶ realisiert mittels Kostenfunktion
- ▶ (detailreichere Darstellung in Jurafsky/Martin)

# Signalerzeugung

- ▶ Signalquelle erzeugt Anregungssignal bestimmter Grundfrequenz

# Signalerzeugung

- ▶ Signalquelle erzeugt Anregungssignal bestimmter Grundfrequenz
- ▶ Anregungssignal wird gefiltert (Aufprägung der lautcharakteristischen Spektren)

# Signalerzeugung

- ▶ Signalquelle erzeugt Anregungssignal bestimmter Grundfrequenz
- ▶ Anregungssignal wird gefiltert (Aufprägung der lautcharakteristischen Spektren)
- ▶ je nach Filterparametern/Sichtweise des Sprechens unterschiedliche Ansätze

# Signalerzeugung

- ▶ Signalquelle erzeugt Anregungssignal bestimmter Grundfrequenz
- ▶ Anregungssignal wird gefiltert (Aufprägung der lautcharakteristischen Spektren)
- ▶ je nach Filterparametern/Sichtweise des Sprechens unterschiedliche Ansätze

## Filterarten bei der Signalerzeugung

- ▶ Formantsynthese
- ▶ Akustisches Modell
- ▶ Artikulatorische Synthese

# Formantsynthese

- ▶ Beobachtung: zur Unterscheidung von Vokalen ist es genug, die ersten zwei Formanten treffend zu reproduzieren
- ▶ realisiert durch einen Filter

# Akustisches Modell

- ▶ bildet gesamte Resonanzeigenschaften des Vokaltrakts durch Filter nach
- ▶ Betrachtung des Vokaltrakts als Rohr mit variablem Querschnitt ("Breite" ist vernachlässigbar)

# Artikulatorische Synthese

- ▶ stellt eine Beziehung zwischen der Stellung der Artikulatoren und dem (daraus resultierenden) Querschnittsverlauf des Vokaltrakts her

# eSpeak

- ▶ kompaktes Open Source Sprachsyntheseprogramm in C++
- ▶ entwickelt von Jonathan Duddington

# eSpeak

- ▶ kompaktes Open Source Sprachsyntheseprogramm in C++
- ▶ entwickelt von Jonathan Duddington
- ▶ auf Kommandozeile oder mit GUI
- ▶ unterstützt viele Sprachen (z.B. Afrikaans, Esperanto)
- ▶ erweiterbar um Phoneme, Sprachen, Stimmen
- ▶ kann mit MBROLA-Stimmen genutzt werden

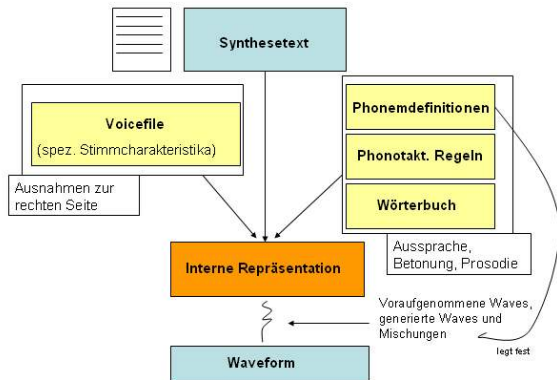
# Funktionsweise

- ▶ Stimmlose Konsonanten: Abspielen von WAV-Dateien [p][t][s]
- ▶ Vokale/Sonoranten: Generieren von WAV-Dateien aus Sequenzen von Formantparametern [a][l][j][n]
- ▶ Stimmlose Plosive/Frikative: WAV-Dateien werden mit generierten WAVs gemischt [b] [g] [v] [z]

## Jonathan Duddington über eSpeak:

eSpeak produces good quality English speech. It uses a different synthesis method from other open source TTS engines, and sounds quite different. It's perhaps not as natural or "smooth", but I find the articulation clearer and easier to listen to for long periods.

# Funktionsweise



# Generelles

## Wofür wird EspeakEdit verwendet?

- ▶ Vorbereiten der Phonemdaten für Espeak:
  - ▶ Erstellen/Verändern der keyframe-Dateien für Vokale und stimmhafte Konsonanten
  - ▶ Kompilieren der Datenpakete (Zusammensetzen der keyframes und der Audiodateien dazu)
- ▶ Herumexperimentieren mit Phonemlängen und Intonation

## Anhang: Kommandozeilenbefehle für eSpeak

Die Kommandozeilenbefehle für eSpeak befinden sich in  
“eSpeakInstallationDir” /docs/command.html

# Quellen

- ▶ <http://de.wikipedia.org/wiki/Sprachsynthese>
- ▶ Jurafsky, Martin: Speech and Language Processing (noch nicht erschienen; Draft-Version vom 23.05.07)
- ▶ P.A. Taylor: Concept-to-Speech Synthesis by Phonological Structure Matching, Centre for Speech Technology Research, University of Edinburgh