

Text-to-Speech synthesis using OpenMARY

An introduction and practical tutorial

Marc Schröder, DFKI
marc.schroeder@dfki.de

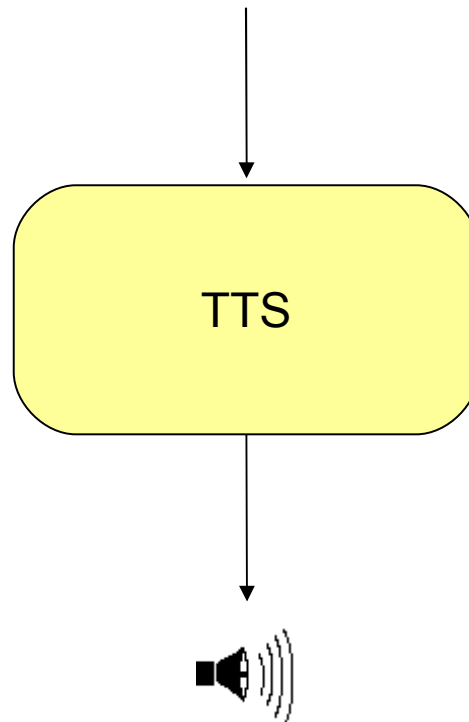
eINTERFACE Amsterdam, 14 July 2010

Overview

- ◆ Some Text-to-Speech (TTS) basics
 - ➔ Natural Language Processing
 - ➔ Generating the sound
 - diphone synthesis
 - unit selection synthesis
 - HMM-based synthesis
- ◆ OpenMARY
 - ➔ existing system MARY 4.0
 - ➔ toolkit for adding new languages and voices
- ◆ Tutorial overview
 - ➔ what you will learn to do in the tutorial

What is text-to-speech synthesis?

“You have one message from Dr Johnson.”

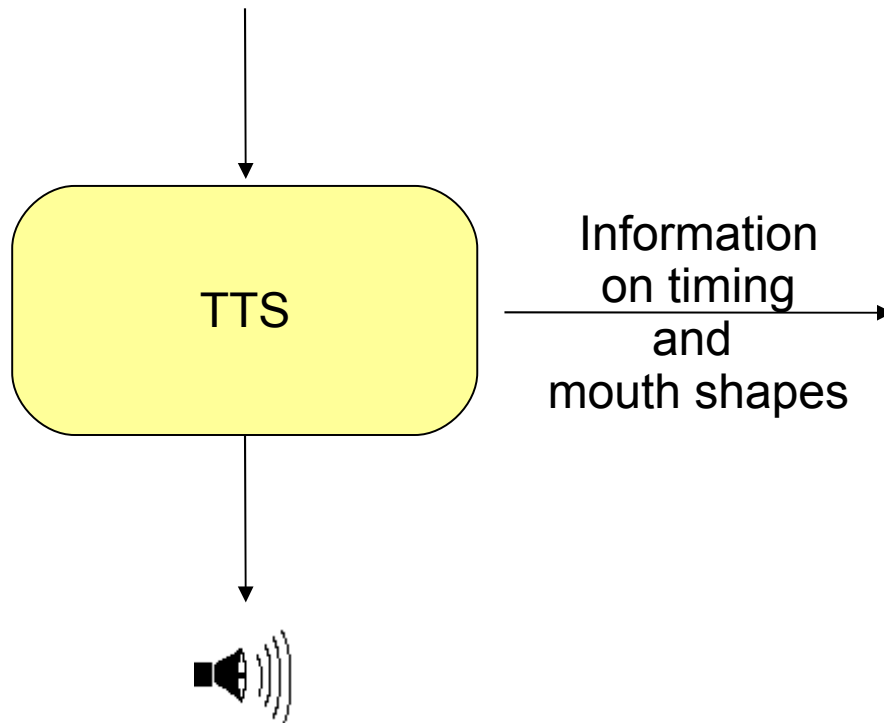


Applications of TTS

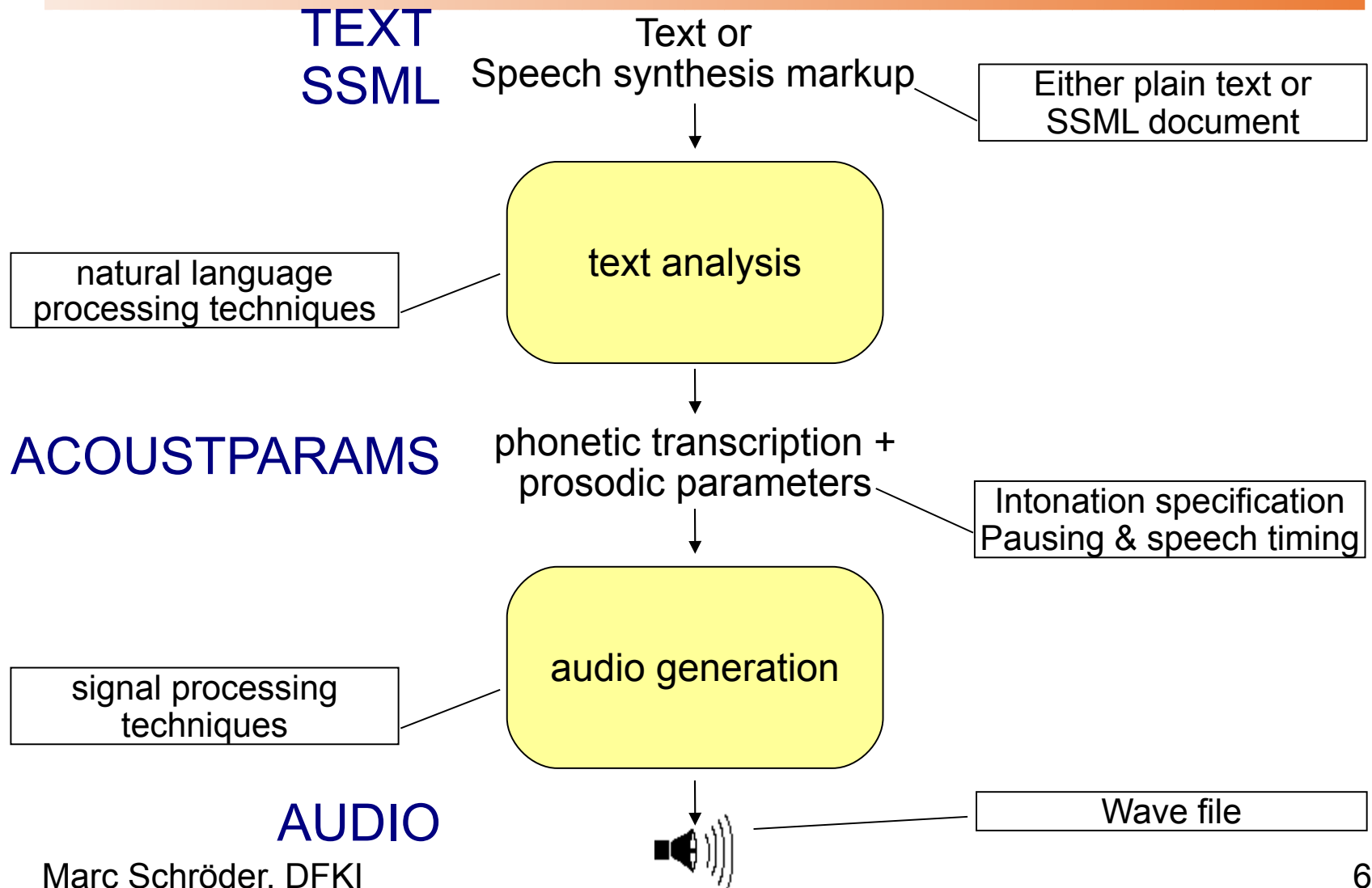
- ◆ Texts readers
 - ➔ for the blind
 - ➔ in eyes-free environments (e.g., while driving)
- ◆ Telephone-based voice portals
- ◆ Multi-modal interactive systems
 - ➔ talking heads
 - ➔ “embodied conversational agents” (ECAs)

A Talking Head

“Hello, nice to meet you.”



Structure of a TTS system



Structure of a TTS system: MARY TTS

◆ Text analysis

- ➔ Input markup parser TEXT or SSML → RAWMARYXML
- ➔ Shallow NLP RAWMARYXML → PARTSOFSPEECH
- ➔ Phonemiser PARTSOFSPEECH → ALLOPHONES
- ➔ Symbolic prosody ALLOPHONES → INTONATION
- ➔ Acoust. parameters INTONATION → ACOUSTPARAMS

◆ Audio generation

- ➔ waveform synthesis ACOUSTPARAMS → AUDIO

System structure: Input markup parser

TEXT or SSML → RAWMARYXML

- ◆ System-internal XML representation **MaryXML**
- ◆ => speech synthesis markup parsing is simple XML transformation
- ◆ Use XSLT => easily adaptable to new markup language

System structure: Shallow NLP

◆ Shallow NLP

- ➔ Tokeniser RAWMARYXML → TOKENS
 - sentence boundaries, “tokens” = word-like units
- ➔ Text normalisation TOKENS → WORDS
 - expanded, pronounceable forms (see next slide)
- ➔ Part-of-speech tagger WORDS → PARTSOF SPEECH

Preprocessing / Text normalisation

- Net patterns (email, web addresses) info@dfki.de
- Date patterns 23/07/2001
- Time patterns 12:24 h, 12:24
- Duration patterns 12:24 h, 12 h 24 min
- Currency patterns 12.95 €
- Measure patterns 123.09 km
- Telephone number patterns +49-681-85775-5303
- Number patterns (cardinal, ordinal, roman) 3 3rd III.
- Abbreviations engl.
- Special characters &

System structure: Phonemisation

- ◆ Phonemiser **PARTSOF SPEECH → PHONEMES**
 - lexicon lookup
 - letter-to-sound conversion
 - morphological decomposition
 - letter-to-sound rules
 - syllabification
 - word stress assignment
- ◆ Custom pronunciation **PHONEMES → ALLOPHONES**
 - slurring, non-standard pronunciation
 - potentially trainable from annotated data of a given person

System structure: Prosody

◆ “Prosody”?

- intonation (accented syllables; high or low phrase boundaries)
- rhythmic effects (pauses, syllable durations)
- loudness, voice quality

◆ Symbolic prosody prediction

ALLOPHONES → INTONATION

- ➔ assign prosody by rule, based on
 - punctuation
 - part-of-speech
- ➔ modelled using “Tones and Break Indices” (ToBI)
 - tonal targets: accents, boundary tones
 - phrase breaks

System structure:

Calculation of acoustic parameters

◆ Duration prediction **INTONATION** → **DURATIONS**

⇒ segment duration predicted

- by rules
- or by decision trees

◆ Contour generation **DURATIONS** → **ACOUSTPARAMS**

⇒ fundamental frequency curve predicted

- by rules
- or by decision trees

System structure: Waveform synthesis

- ◆ **Waveform synthesis** ACOUSTPARAMS → AUDIO
 - ➔ several waveform generation technologies

Creating sound: Waveform synthesis technologies (1)

◆ Formant synthesis

- acoustic model of speech
- generate acoustic structure by rule
- robotic sound

Creating sound: Waveform synthesis technologies (2)

◆ Concatenative synthesis

➔ diphone synthesis

- glue pre-recorded “diphones” together
- adapt prosody through signal processing

➔ unit selection synthesis

- glue units from a large corpus of speech together
- prosody comes from the corpus, (nearly) no signal processing

Creating sound: Waveform synthesis technologies (3)

- ◆ **Statistical-parametric speech synthesis**
 - ➔ with Hidden Markov Models
 - ➔ models trained on speech corpora
 - ➔ no data needed at runtime => small footprint

Examples of speech synthesis technologies

◆ MARY TTS

→ unit selection



→ HMM-based



→ MBROLA diphones



→ expressive unit selection



◆ Commercial

→ unit selection

▪ IVONA



▪ Loquendo



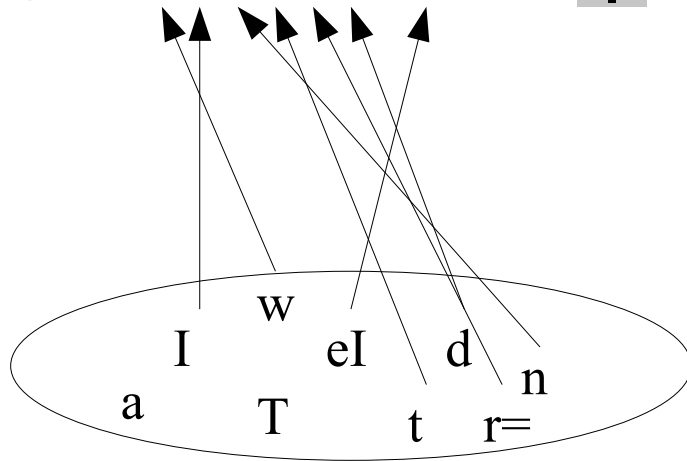
→ formant synthesis

▪ DecTalk



Concatenative synthesis: Isolated phones don't work

target: w I n t r= d eI



acoustic unit database
(units = **phone segments** recorded in isolation)

Concatenative synthesis: Diphones

target: w I n t r= d eI

_ -w w-I I-n n-t t-r= r=-d d-eI eI- _



_ -w (wonder)

w-I (will)

I-n (spin)

n-t (fountain)

t-r= (water)

r=-d (nerdy)

d-eI (date)

eI- _ (away)



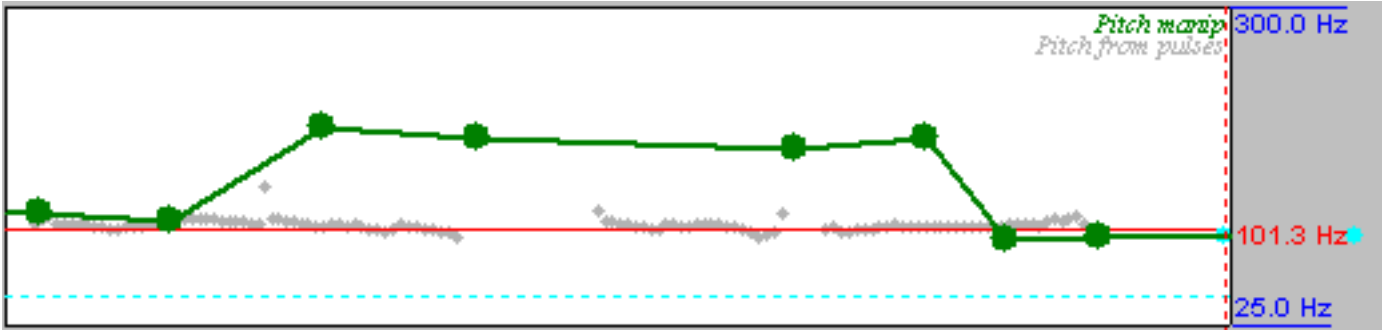
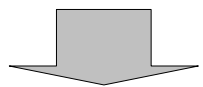
Diphones =

sound segments
from the middle of one phone
to the middle of the next phone

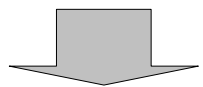
acoustic unit database
units = **diphone segments**
recorded in carrier words
(flat intonation)

Concatenative synthesis: Diphones (2)

target: w I n t r= d eI
w w-I I-n n-t t-r= r=-d d-eI eI-



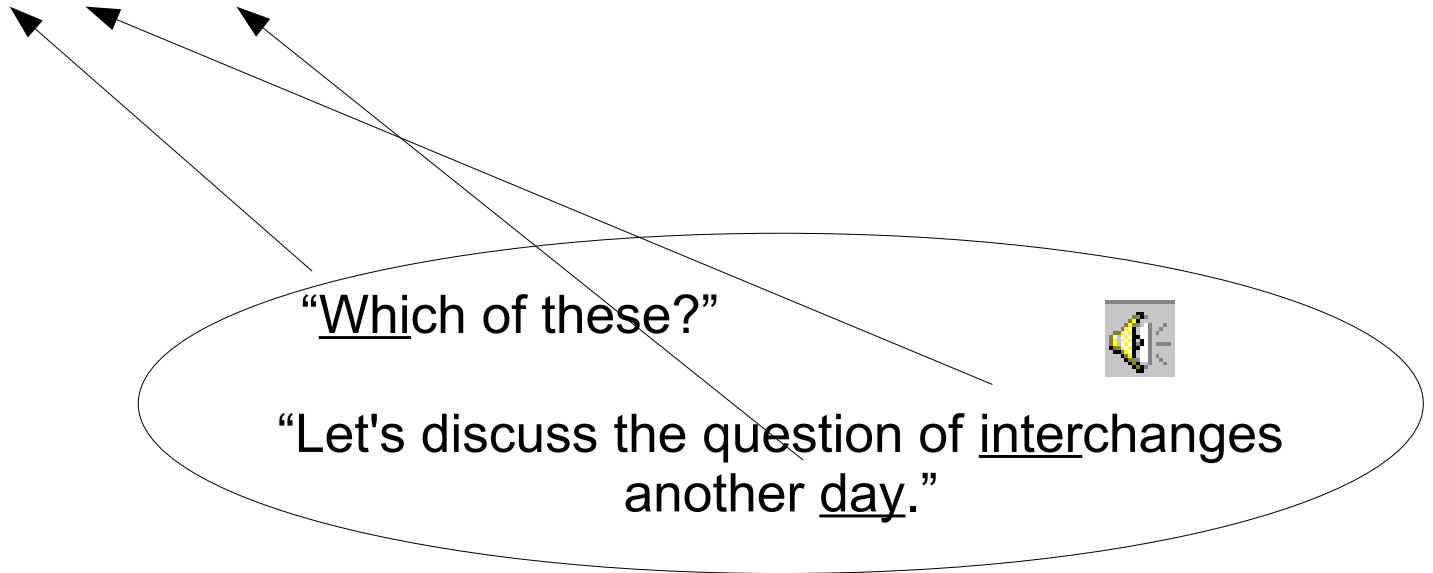
PSOLA
pitch
manipulation



Concatenative synthesis

Unit selection

target: w I n t r = d e I



acoustic unit database

units = **(di-)phone segments** recorded in natural sentences (natural intonation)

AI Poker: The voices of Sam and Max



Sam:

- ➔ Unit Selection Synthesis
- ➔ Voice specifically recorded for AI Poker
- ➔ Natural sound within poker domain

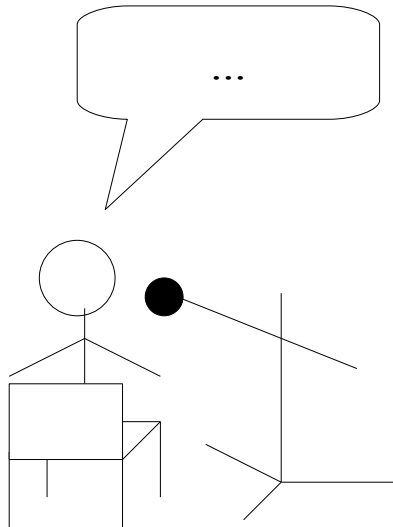
Max:

- ➔ HMM-based synthesis
- ➔ Sound quality is limited but constant with any text

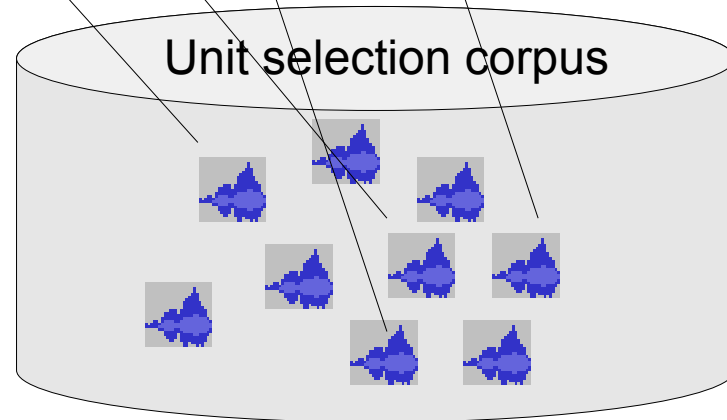


Sam's voice: Unit selection synthesis

"Ich habe zwei Paare."



several hours of speech recordings

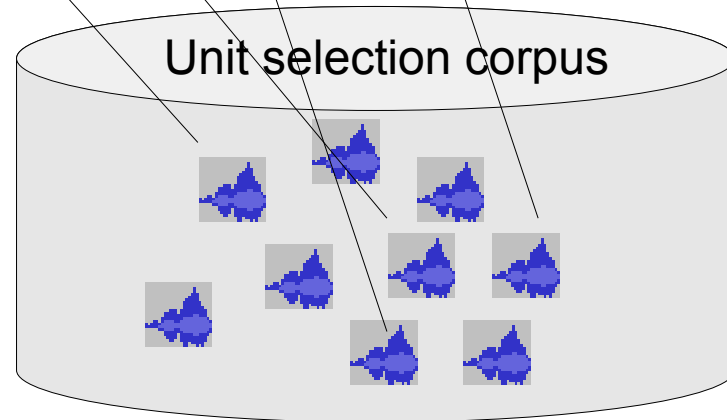
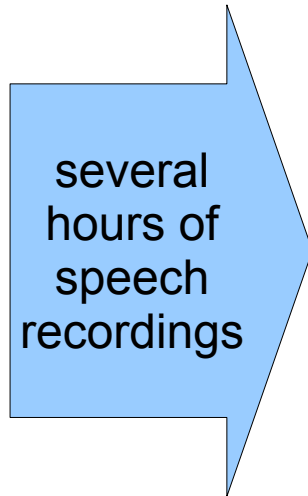
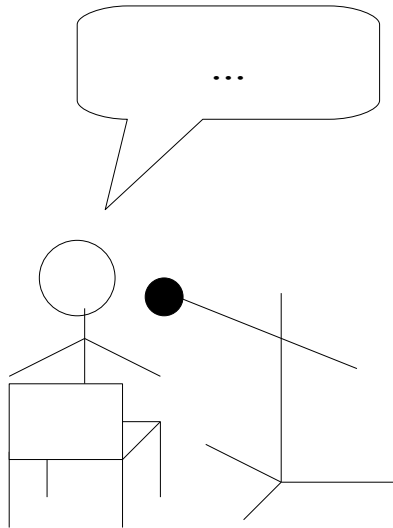


=> very good quality within the poker domain!



Sam's voice: Unit selection synthesis

“Ich kann auch ganz andere Sachen...”

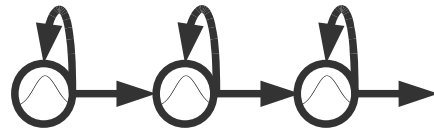


reduced quality with arbitrary text

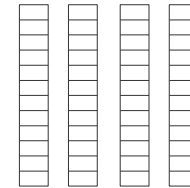
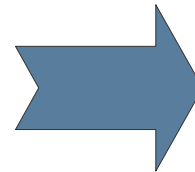
Max's voice: HMM-based synthesis



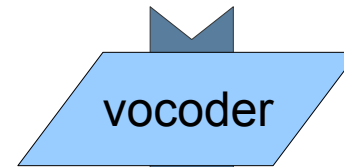
“Ich habe zwei Paare.”



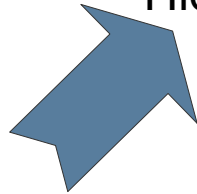
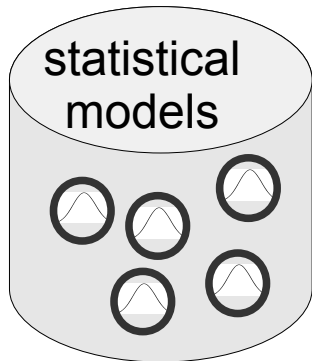
Hidden Markov Models



acoustic
feature vectors



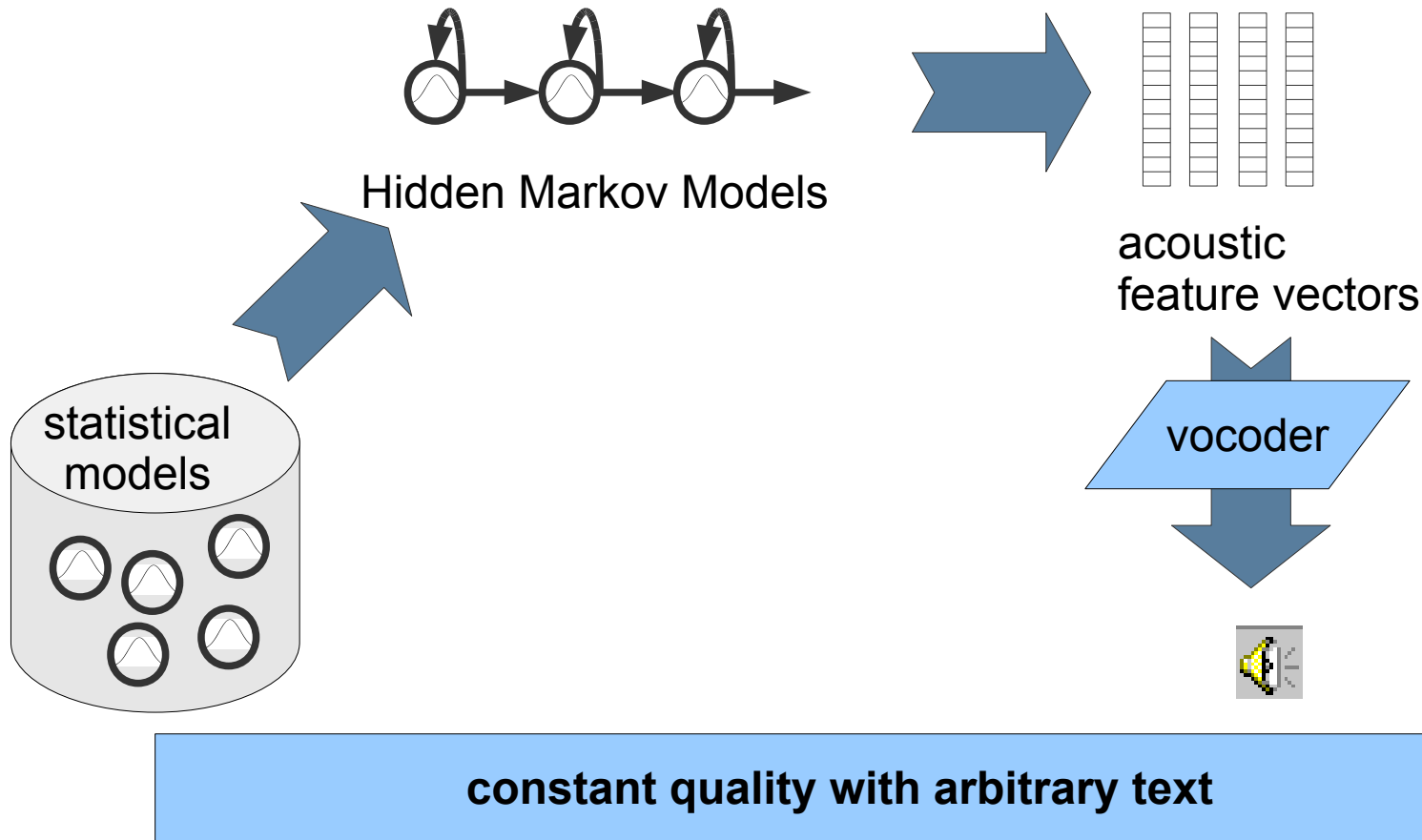
vocoder



Max's voice: HMM-based synthesis



“Ich kann auch ganz andere Sachen...”



MARY TTS 4.0

- ◆ Pure Java

- ➔ Runs on any platform with Java 5

- ◆ Client-server architecture

- ➔ http interface – your browser is a MARY client

- ◆ Multilingual, with UTF-8 support
















- ➔ English (US and GB)

- ➔ German Willkommen

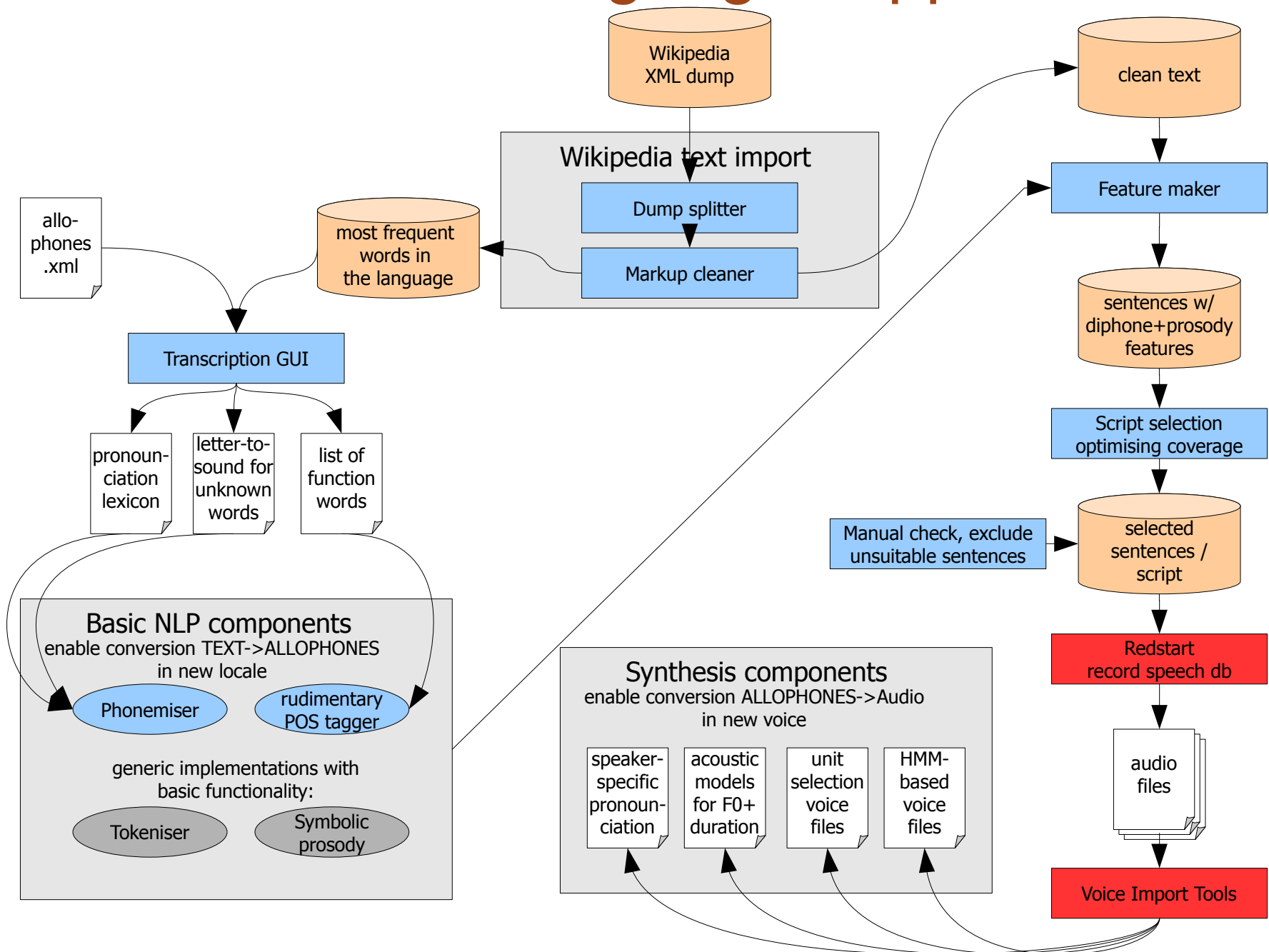
- ➔ Turkish Konuşma

- ➔ Telugu స్పీచ్ సిస్టమ్స్

Audio effects in MARY 4.0

- ◆ Some can be applied to any voice 
 - vocal tract length (longer  – shorter )
 - Robot effect 
 - Whisper effect 
 - Jet pilot 
- ◆ More effects for HMM-based voices 
 - pitch level (higher  – lower )
 - pitch range (wider  – narrower )
 - speaking rate (faster  – slower )
- ◆ Can be parameterised & combined to create characteristic voices  

MARY TTS: New language support workflow



What you will learn to do in the MARY Tutorial

- ◆ Installing the MARY system
 - ➔ languages and voices
- ◆ Interacting with MARY using the web client
 - ➔ basic experimentation
 - ➔ interactive test of audio effects
 - ➔ interactive documentation of http interface
- ◆ Triggering TTS from your own software
 - ➔ http interface
 - ➔ Java client code
 - ➔ selecting language, voice and effects in requests

What you will learn to do in the MARY Tutorial (2)

- ◆ Using timing information:
REALISED_ACOUSTPARAMS and
REALISED_DURATIONS
- ◆ Performance: caching