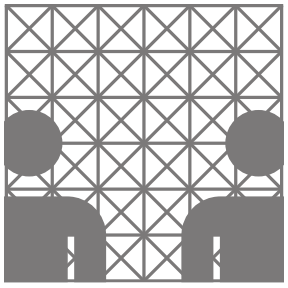
Vorlesung

Sprachdialogsysteme

Timo Baumann
baumann@informatik.uni-hamburg.de

https://nats-www.informatik.uni-hamburg.de/SDS20

Universität Hamburg, Department of Informatics
Language Technology Group



Heute

Leftovers und Reprise Sprachsynthese

- spezifische Schwierigkeiten der “Text-to-Speech”-Synthese

Grammatiken und Sprachverstehen

-

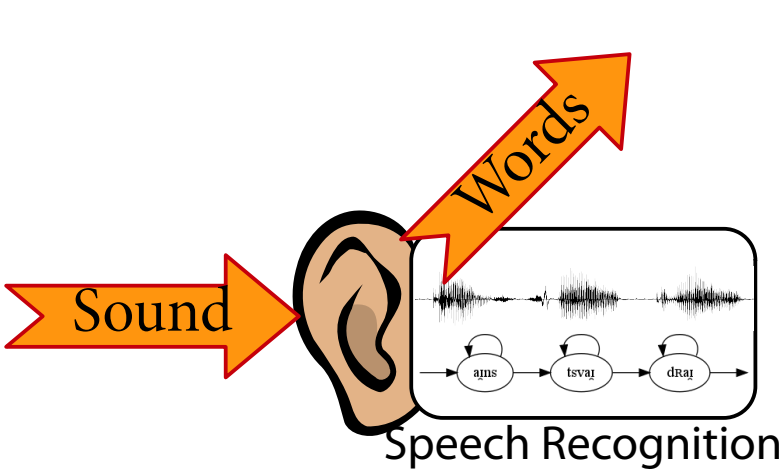
Beispiele

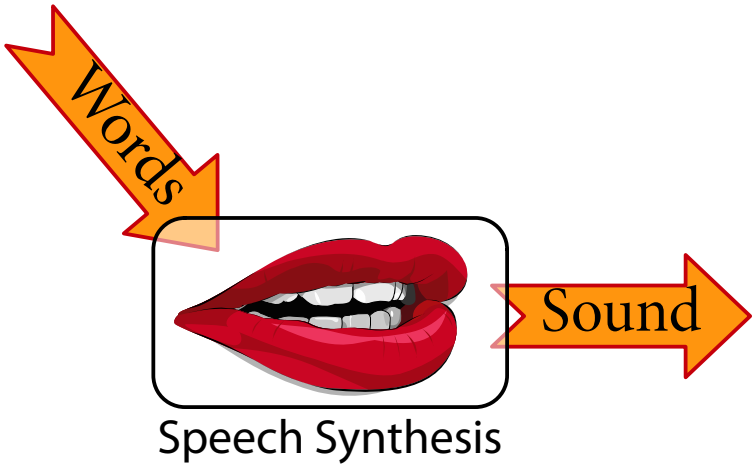
- der erste (digitale) singende Computer (IBM, 1961)
  → hand-optimiertes Vocoding
- aktuelle Implementierung derselben Technik: espeak
  → regel-basiertes Vocoding
- basierend auf Sprachaufnahmen: DreSS-FR, Mbrola
  → Diphon-Synthese
- moderne Variante: MaryTTS
  → generelle konkatenative Synthese (nicht bloß Diphone)
- smartere Version
  → HMM-basierte Synthese (Master-level course ;-)

Input und Output von Sprachdialogsystemen

- Erkennung
  - Reduktion des Signals auf Wörter

  ➔ *Abstrahieren* der Details



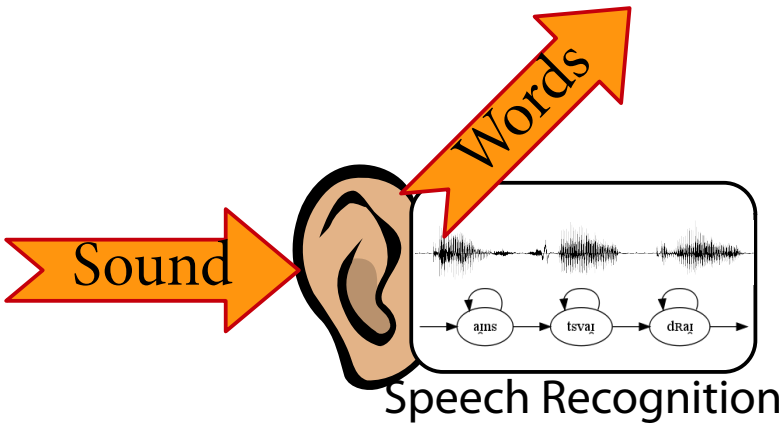


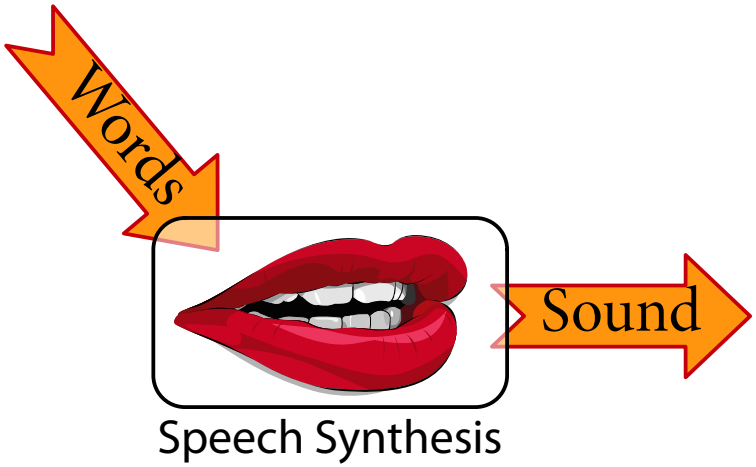
Input und Output von Sprachdialogsystemen

- Erkennung
  - Reduktion des Signals auf Wörter

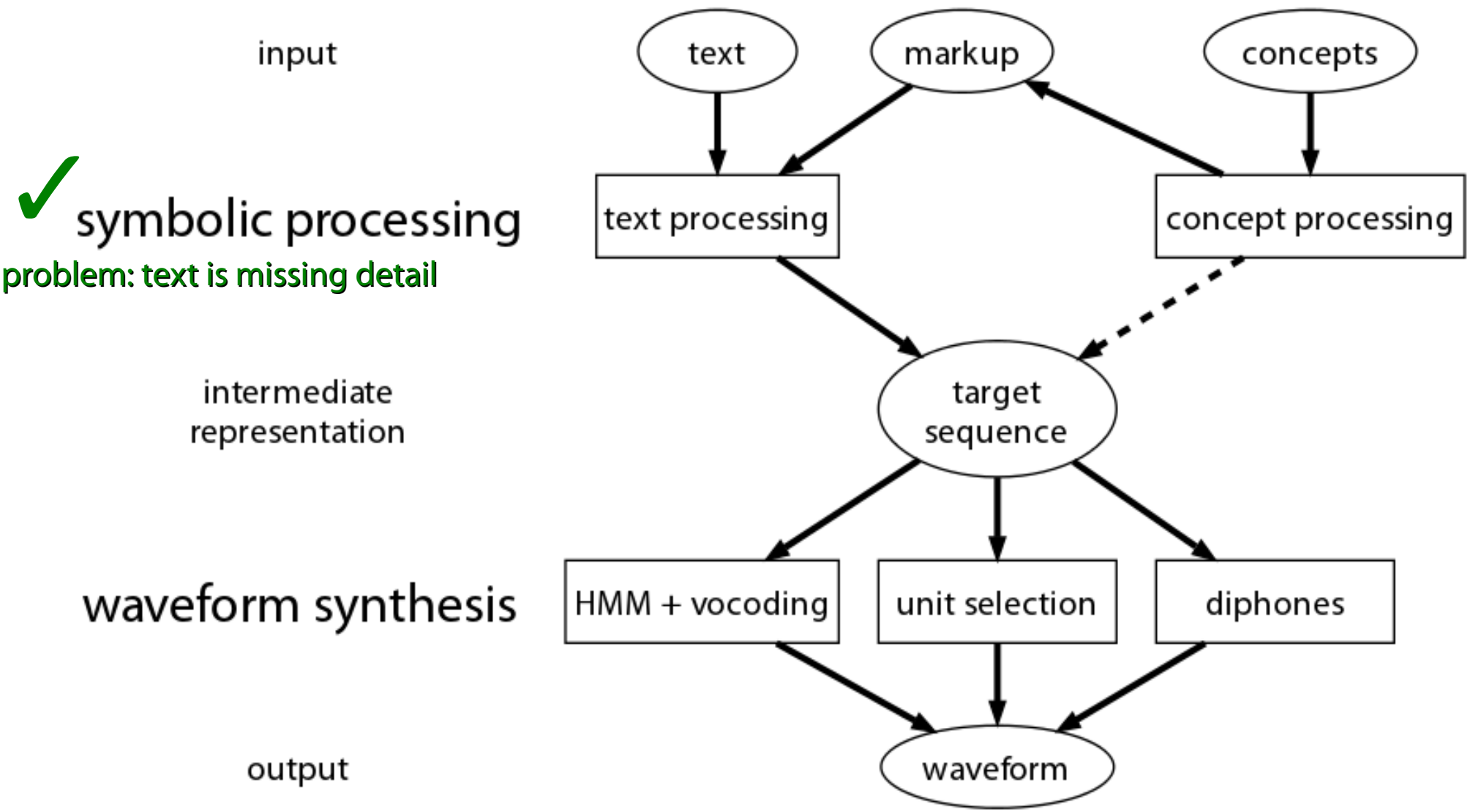
    ➔ *Abstrahieren* der Details

- Synthese
  - Wörter allein beschreiben das Signal nur ungenügend
  - Natürlichkeit *entsteht* aus den Details

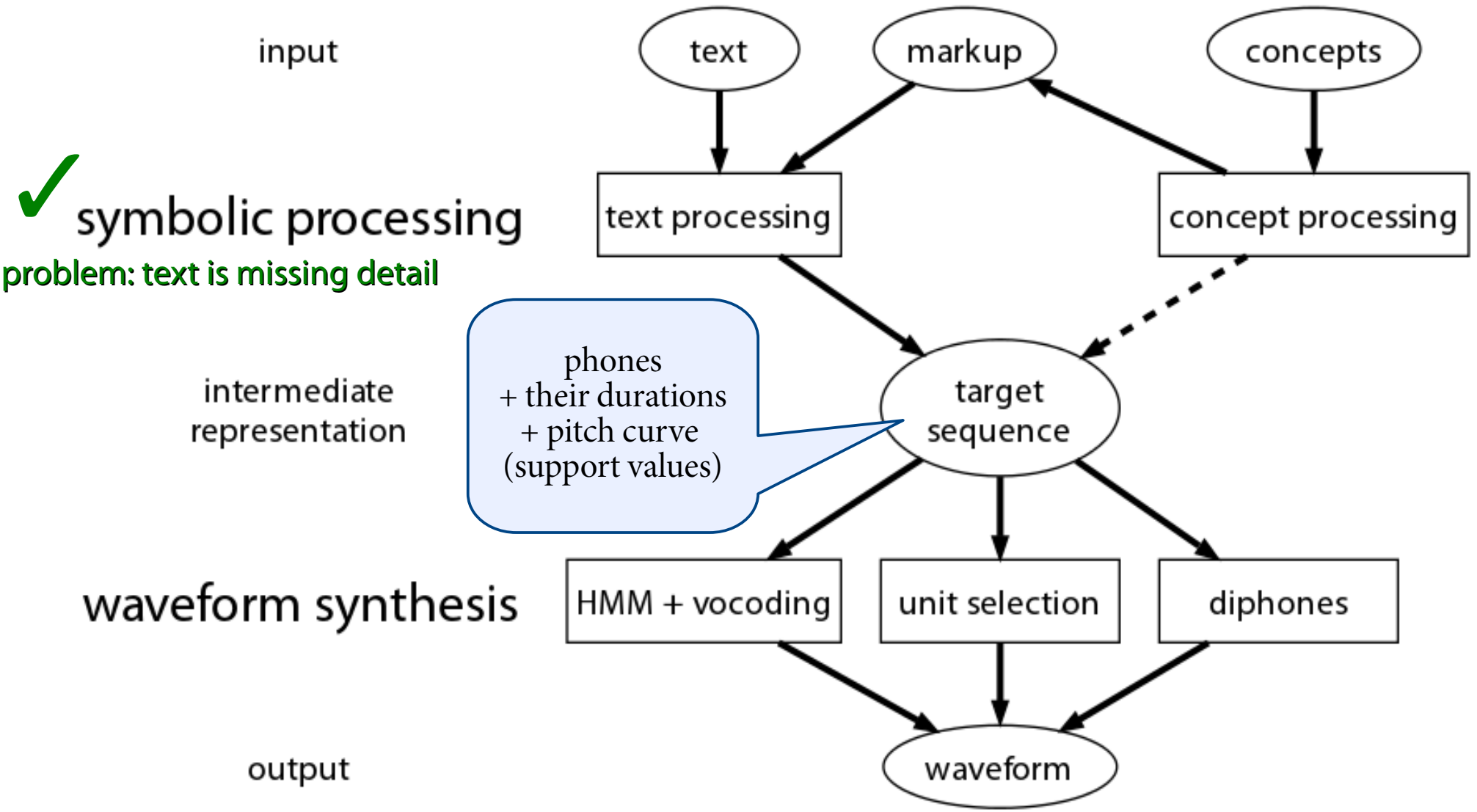




Process diagram of Speech Synthesis



Process diagram of Speech Synthesis



Sprachsignalsynthese

Sprachsignalsynthese

von der “target sequence” (Laute+Dauer+Tonhöhe)

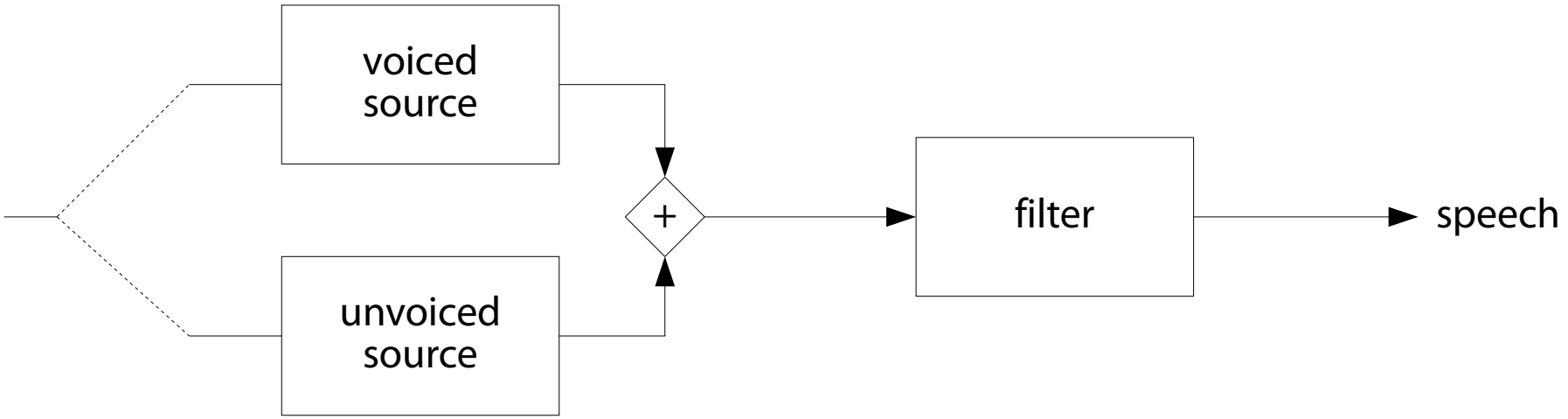
1. formantbasiert:

   Regeln bestimmen Umsetzung der Laute und Lautübergänge mittels *Formanten*, die grob die Lauteigenschaften widerspiegeln; Lauteigenschaften werden durch einen *Vocoder* synthetisiert

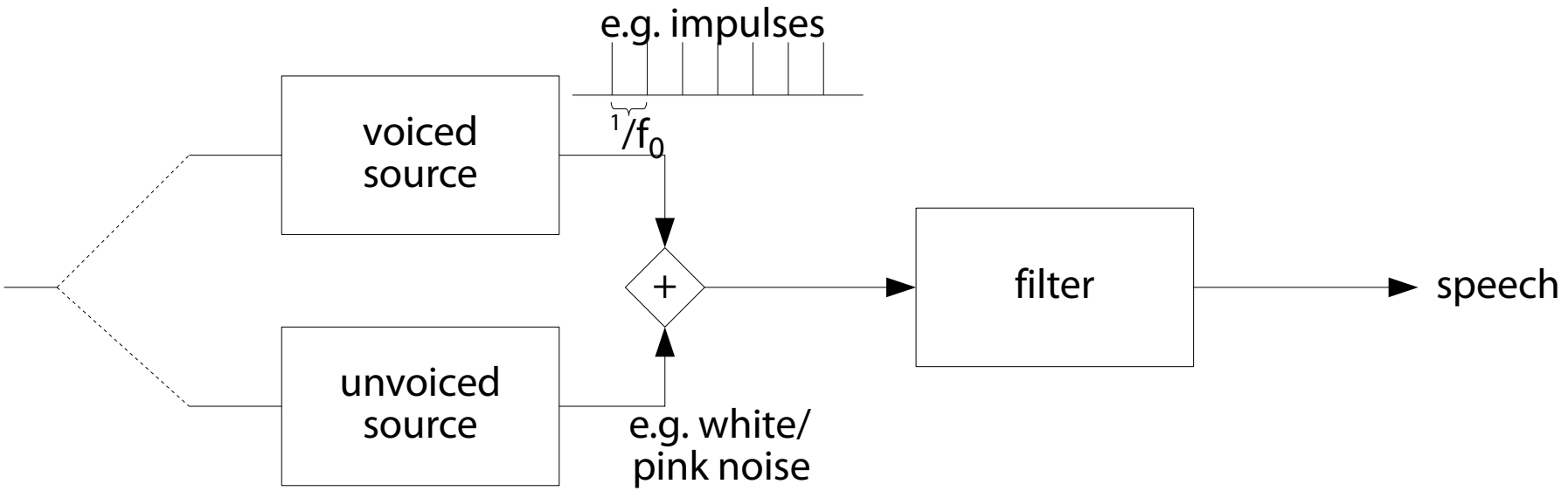
2. musterbasiert (Diphone oder allg. *konkatenative* Synthese):

   große Datenbank mit Sprachsegmenten;
   möglichst gut passende Segmente werden an möglichst gut passenden Stellen *verkettet* und nacheinander abgespielt

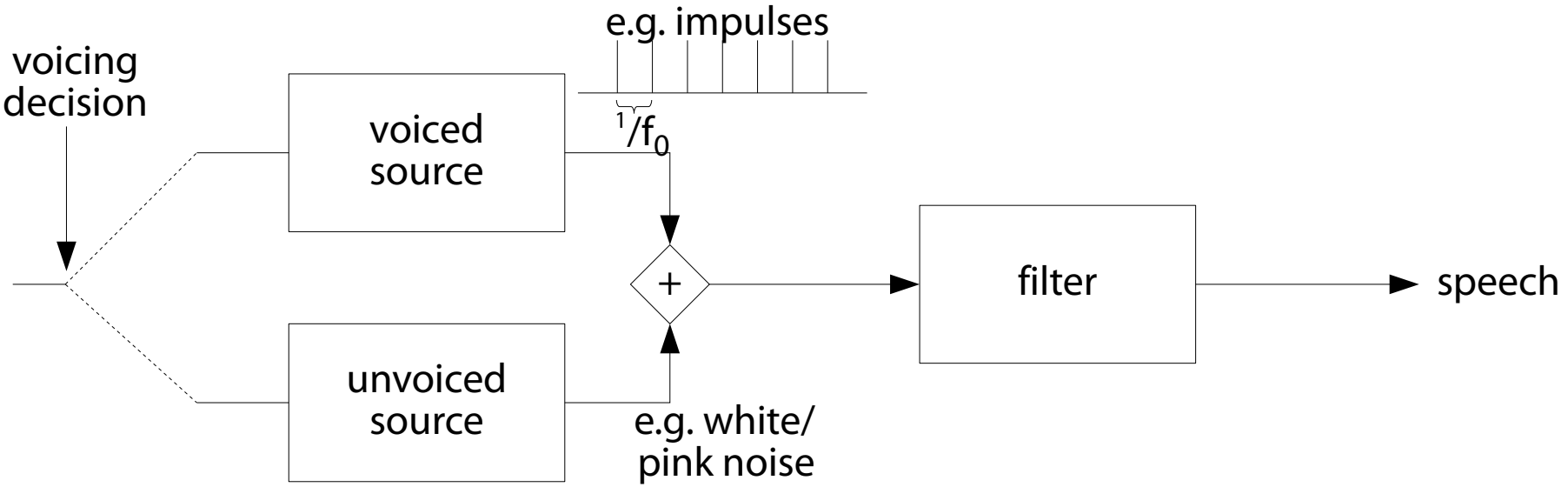
A Simple Vocoder Design



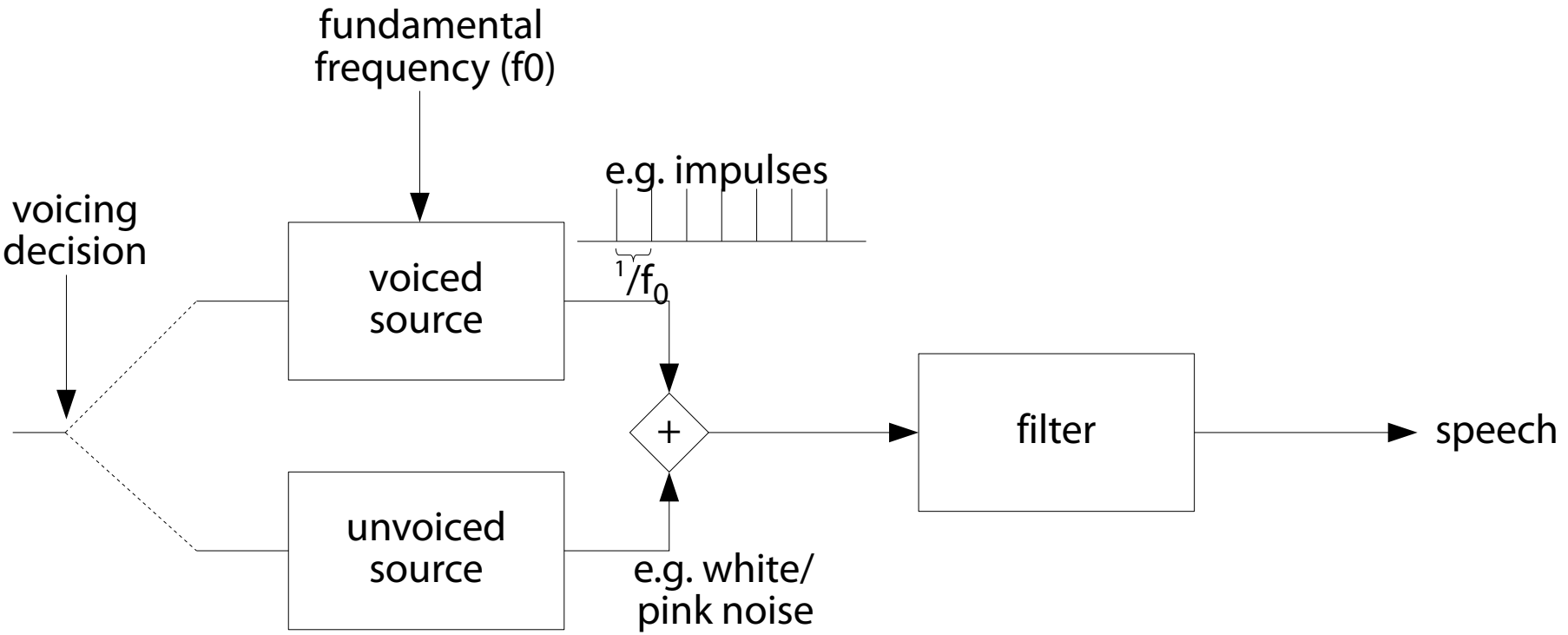
A Simple Vocoder Design



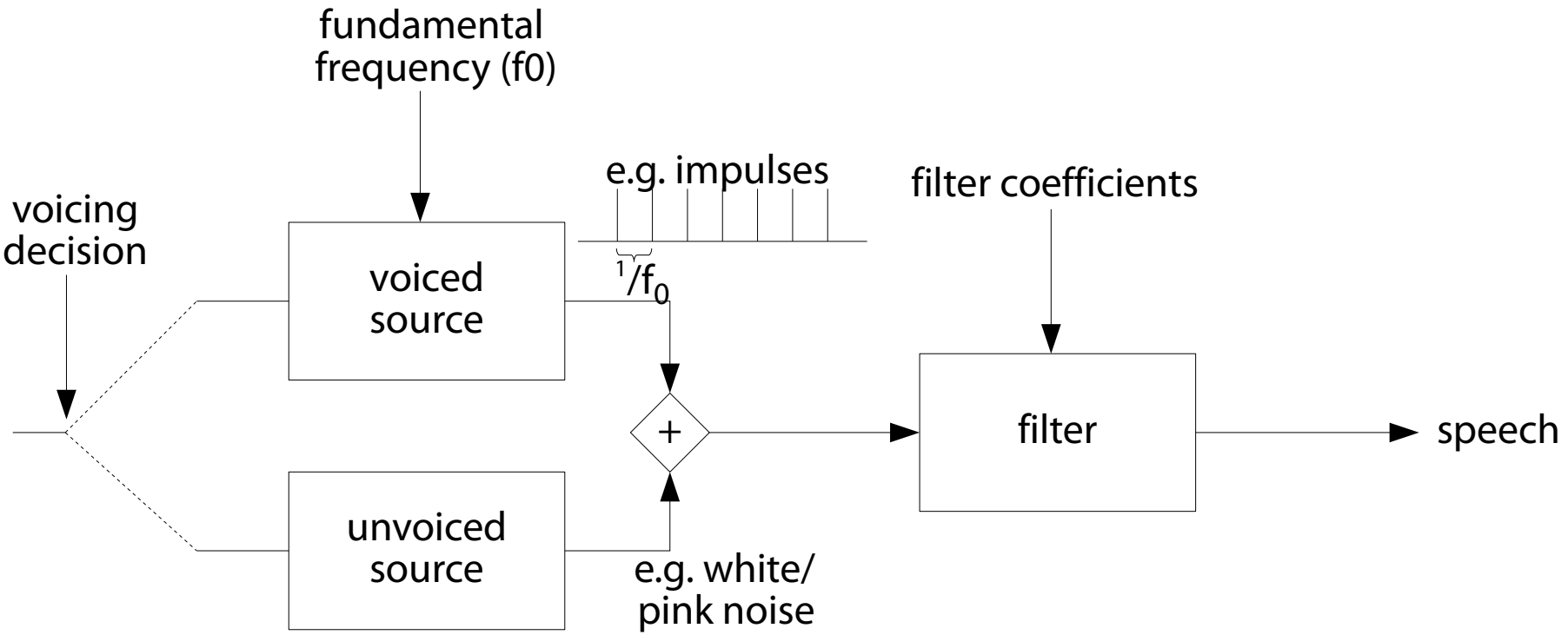
A Simple Vocoder Design



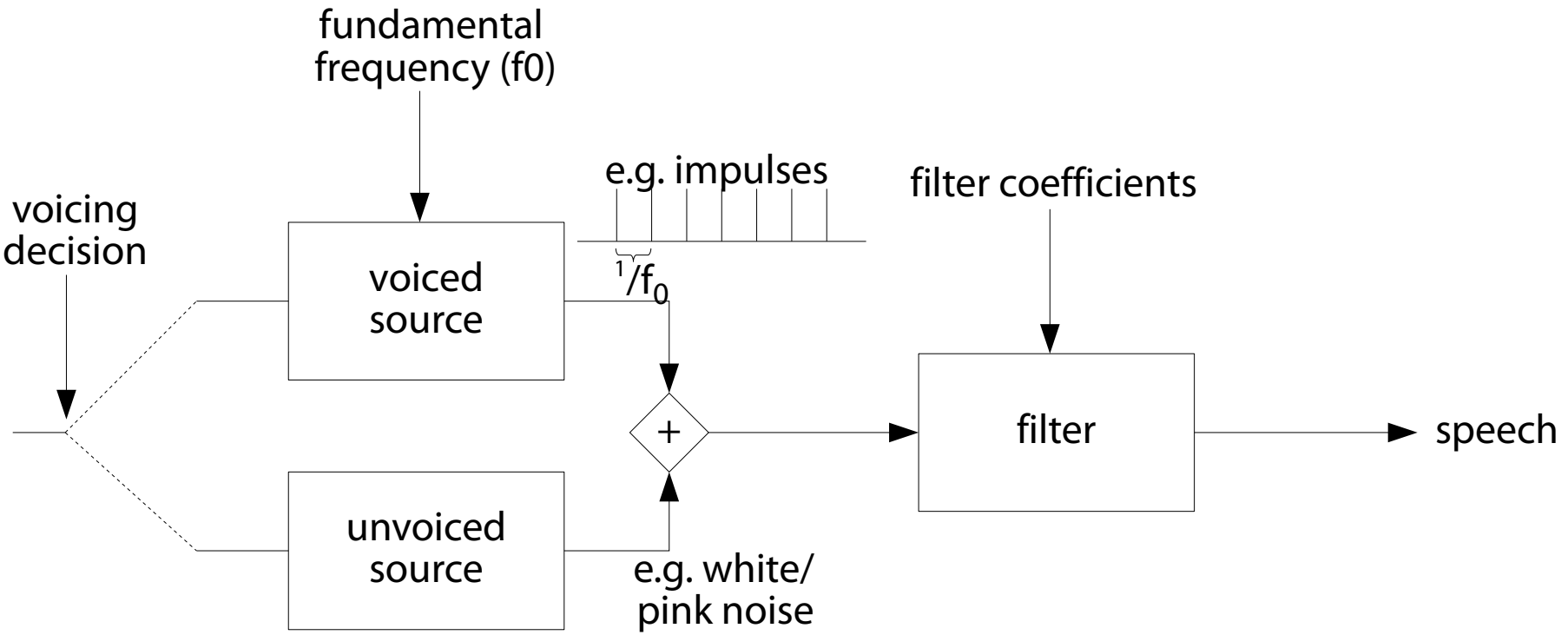
A Simple Vocoder Design



A Simple Vocoder Design

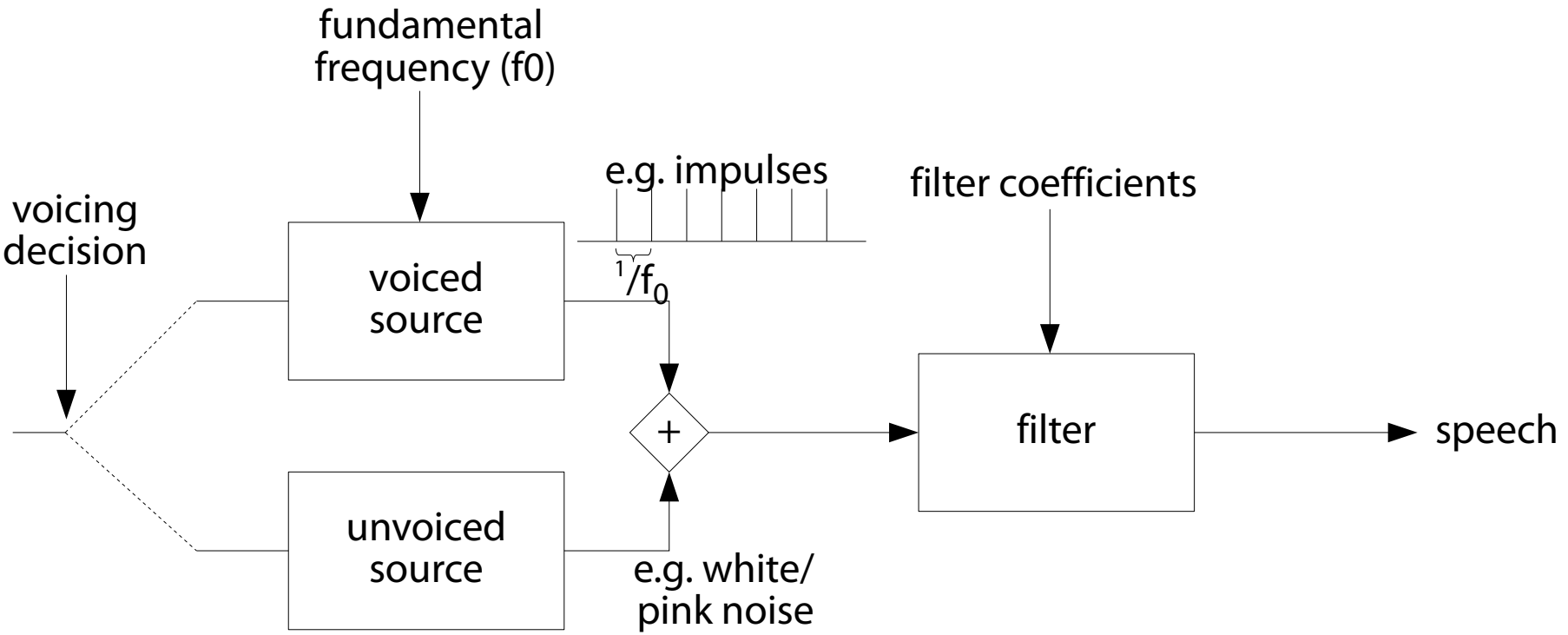


A Simple Vocoder Design



- few parameters in the standard model
  - still, good parameters are the bottleneck (remember eSpeak?)
- extensions: mixed voicing, model for primary signal, ...

A Simple Vocoder Design

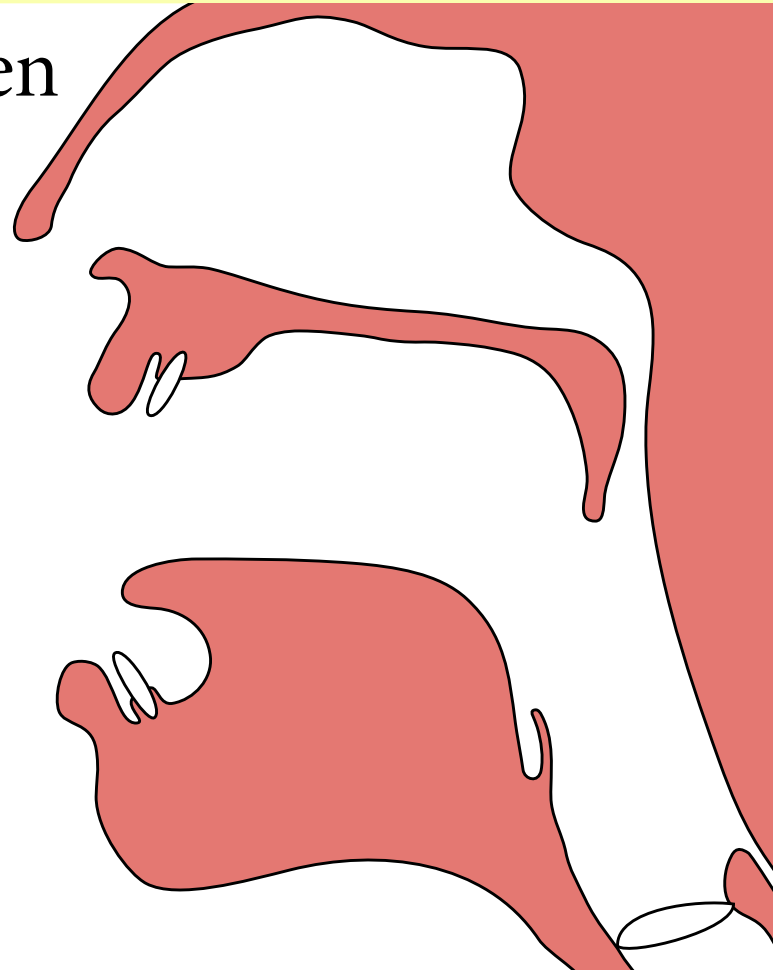


- few parameters in the standard model
  - still, good parameters are the bottleneck (remember eSpeak?)
- extensions: mixed voicing, model for primary signal, ...

Speech Production: Source-Filter Model

- Stimmlippen erzeugen obertonreichen Primärschall

- Vokaltrakt filtert das Signal, sodass nur erwünschte Frequenzen erhalten bleiben



Diphone Synthesis

- Concatenation of short speech snippets
- units from center of a phone to center of the next:
  _h+ha:+a:l+lo:+o:_+_v+vi:+i:g+ge:+e:t+ts+s_
  - concatenation within "stable" phase of the phone
  - coarticulation is (largely) covered
- 40 phones → ~1600 diphones!
  - recorded from one speaker → one voice
  - additional signal processing for duration+pitch change

General Concatenative Synthesis

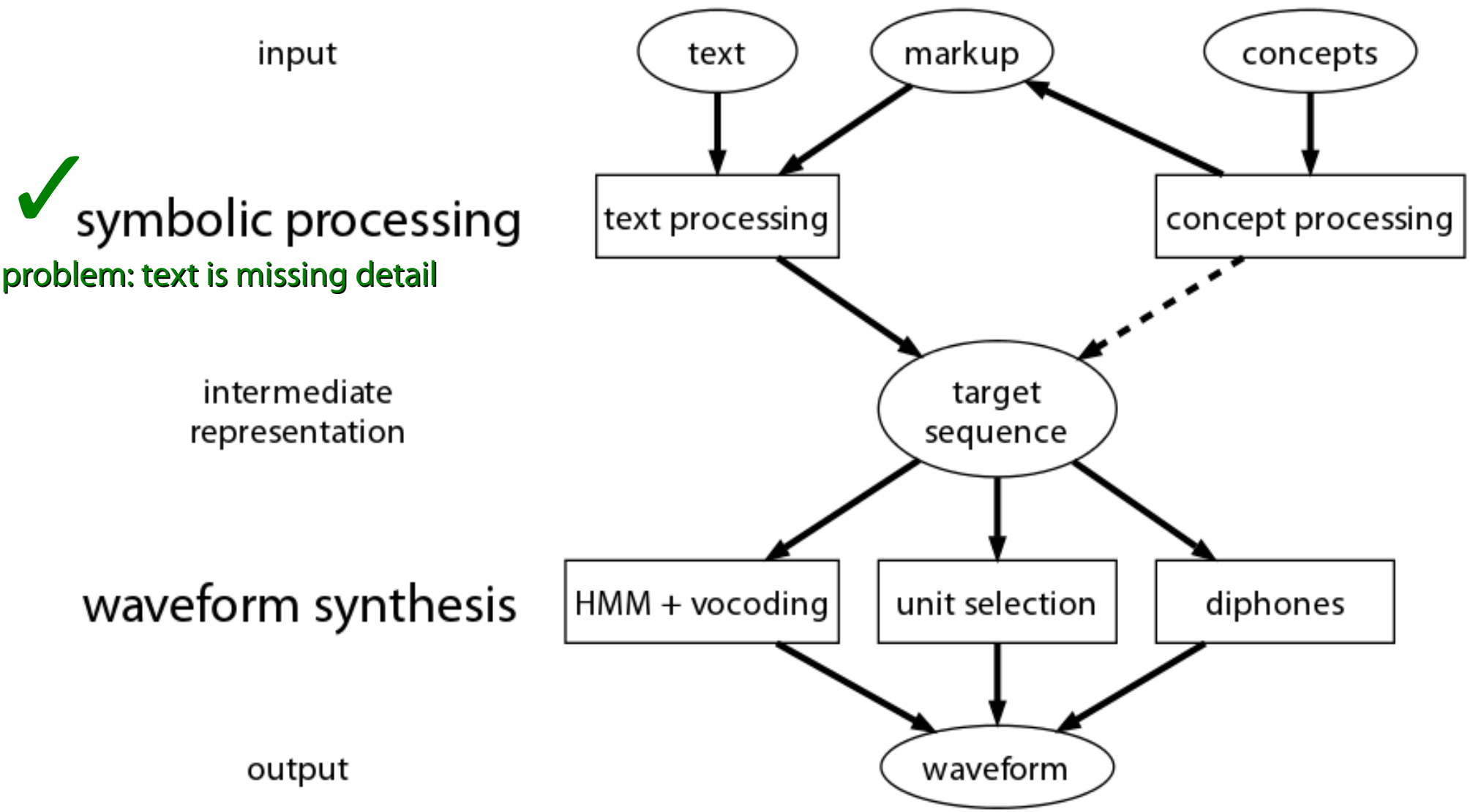
- alternatives for the mapping target → speech snippets
  - more speech material in database
  - selection of material that better fits the target sequence
- selection becomes a search of best concatenation
  - costs of fit of concatenation between snippets
  - costs of fit of snippets to target sequence
- computationally expensive (search)
  - very high memory demands (500MB+ per voice)
- results can be very natural sounding

Welche Form der Synthese finden Sie (als WissenschaftlerIn)

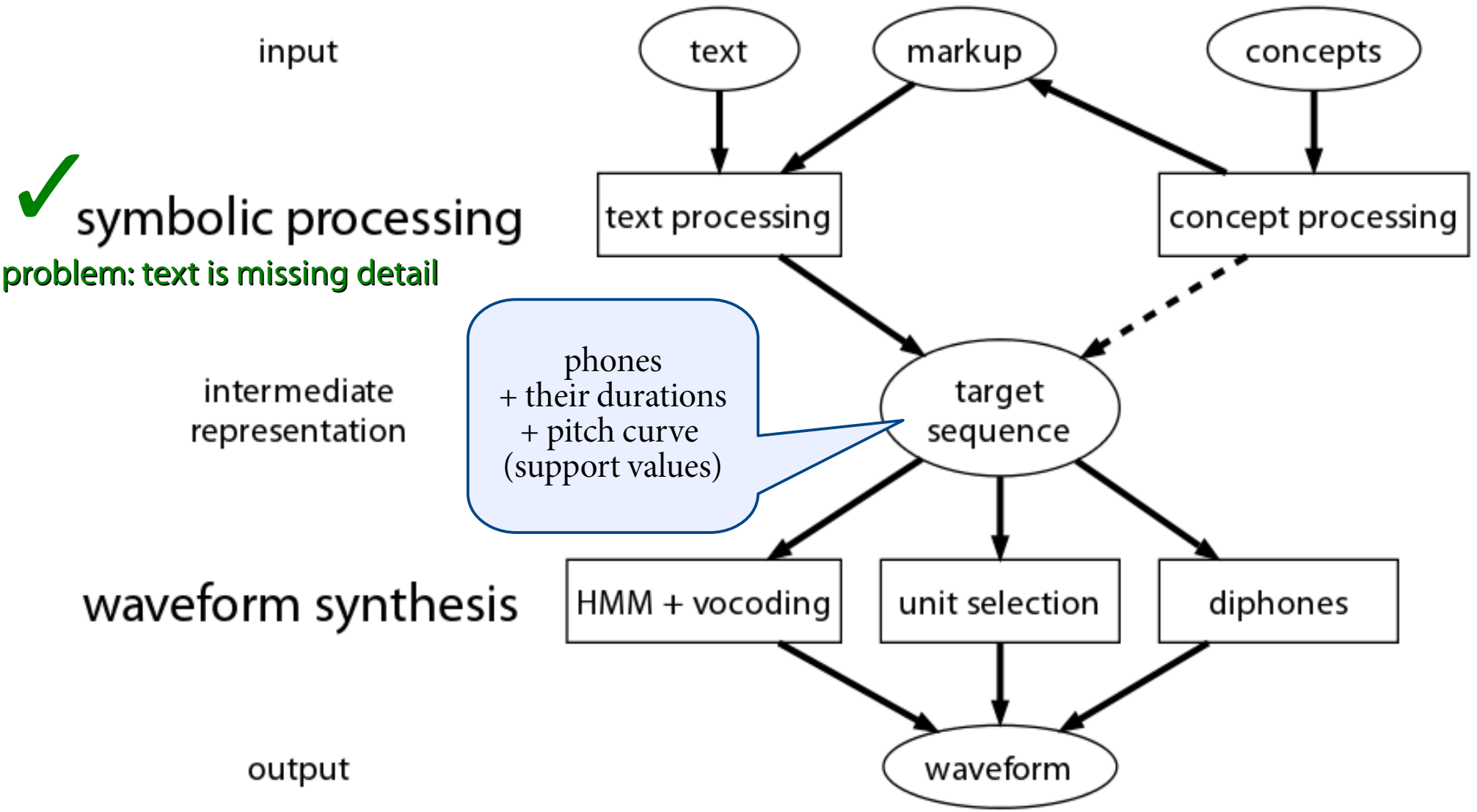
attraktiver,

formant- oder musterbasierte Synthese? Warum?

Process diagram of Speech Synthesis

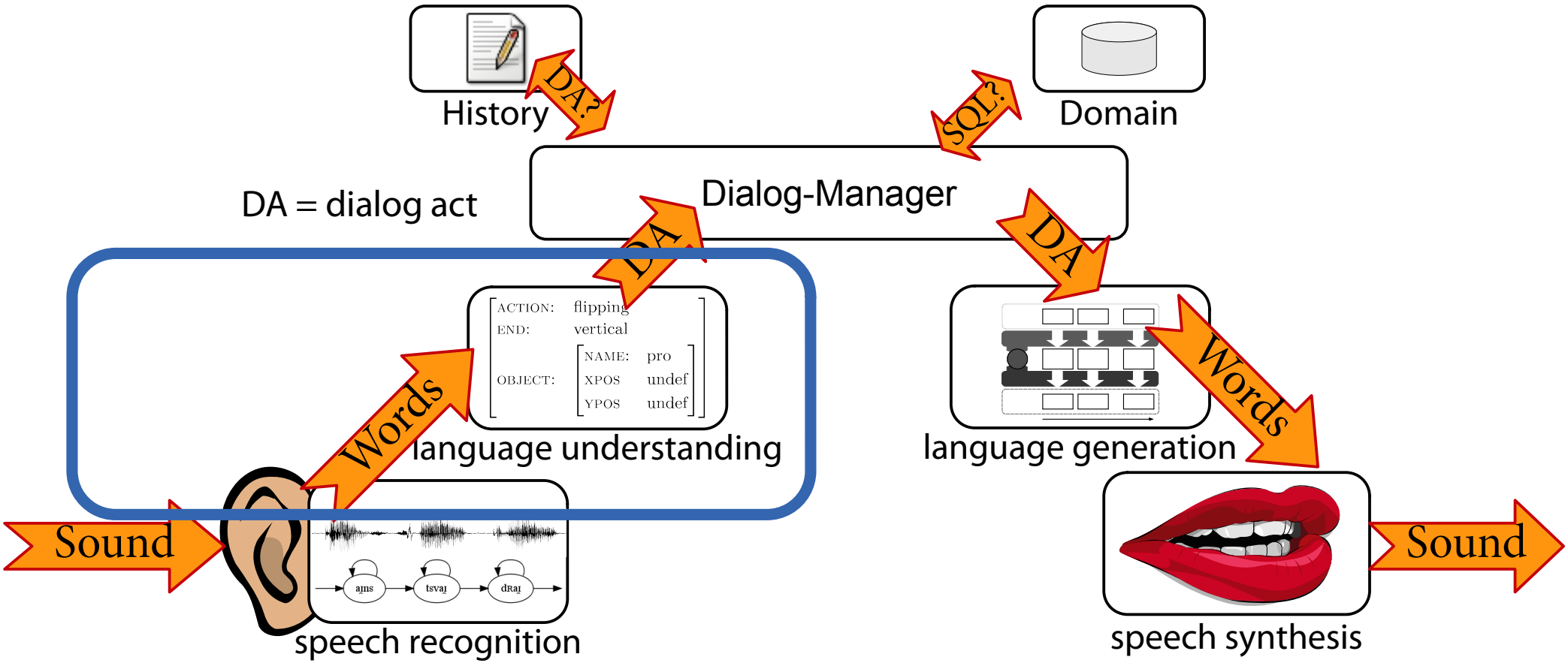


Process diagram of Speech Synthesis

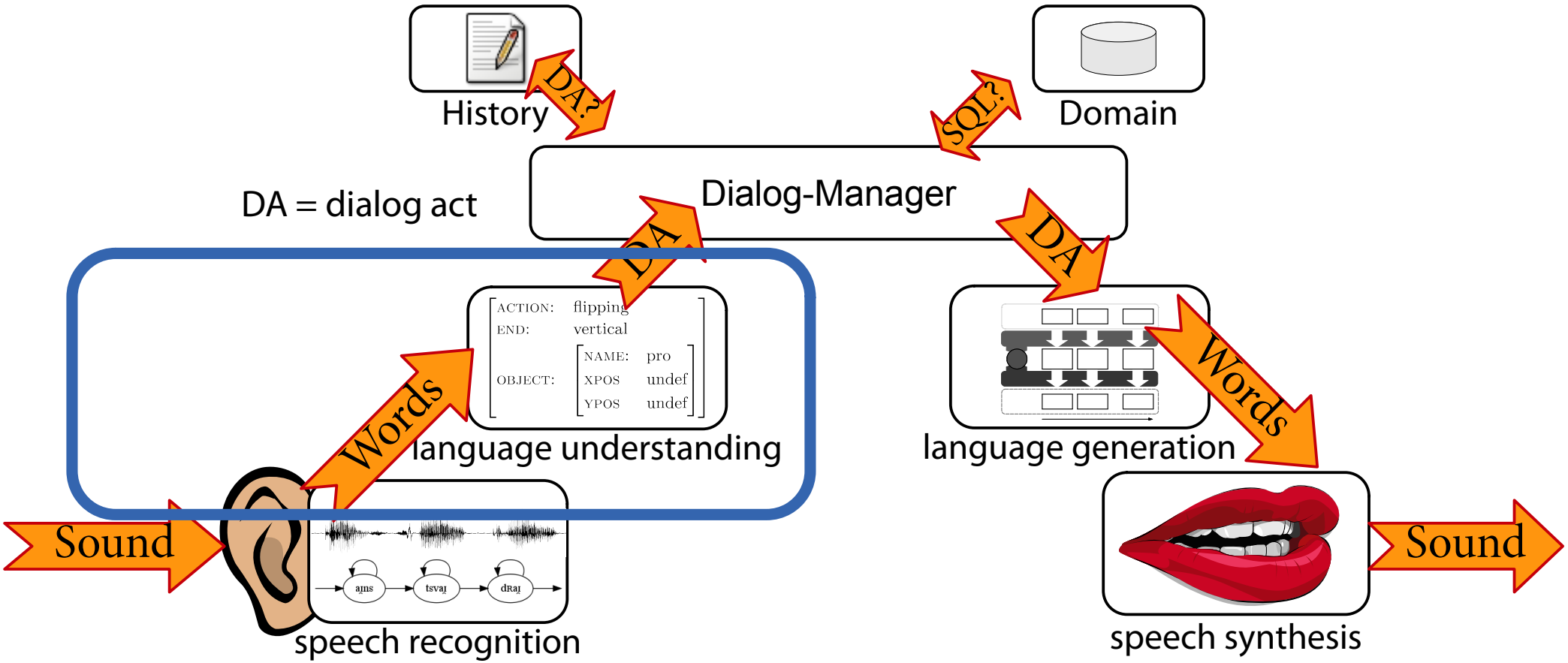


Grammatiken und Sprachverstehen

Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



je weniger Möglichkeiten, desto weniger kann die Spracherkennung verkehrt machen.

regelbasiertes Sprachverstehen

- reguläre Ausdrücke:

  “ich (möchte|mag|hasse) himbeereis [nicht]”

  –

- Grammatiken:

```
S  -> NP VP
NP -> ich | himbeereis | du | ...
    | [Art] N | Pronomen | ...
VP -> V NP | V NP nicht
V  -> möchte | mag | hasse
```

Verstehen mit regulären Ausdrücken

- unterschiedliche Ausdrücke je Bedeutung,

  ```
  ja | meinetwegen | okay
  nein | bitte nicht | nur wenn es sein muss
  ```

  die "matchen" (also passen), oder eben nicht

  – ggfs. "Matching" von Klammerausdrücken und Übernahme des Werts

  ```
  ich möchte (eine|zwei|drei) Kugel[n] Eis.
  ```

  → Auswahl in Klammern wird in Variable gespeichert

- Spracherkennung

  alle möglichen Ausdrücke bilden die Sprache, die die Spracherkennung erkennen kann.

Verstehen mit Grammatiken

- eine kontextfreie Grammatik die alle möglichen Äußerungen *generiert*
  - CFGs sind ausdrucksstärker als reguläre Sprachen, es gibt (verschachtelte Nebensatz-)Konstruktionen, die nicht regulär aber kontextfrei sind
  - sinnvoll für komplexe Sprachanfragen
- Verstehen wird in die Grammatik eingebettet:

```
language "Deutsch";
root $rechnen;
$rechnen = ($zahlA plus $zahlB) { $ = $zahlA + $zahlB; };
$zahlA = $ziffer { $ = parseInt($ziffer); };
$zahlB = $ziffer { $ = parseInt($ziffer); };
$ziffer = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
```

Verstehen mit Grammatiken

- eine kontextfreie Grammatik die alle möglichen Äußerungen *generiert*
  - CFGs sind ausdrucksstärker als reguläre Sprachen, es gibt (verschachtelte Nebensatz-)Konstruktionen, die nicht regulär aber kontextfrei sind
  - sinnvoll für komplexe Sprachanfragen
- Verstehen wird in die Grammatik eingebettet:

Quintessenz

- je weniger die Spracherkennung verstehen kann, desto geringer die Gefahr, dass sie etwas falsch versteht
  - regelbasierte Systeme schränken die Spracherkennung enorm ein
  - Abweichungen beim Sprechen von dem was verstanden werden kann führen zu Nichterkennung: es wird *garnichts* verstanden

- Abwägung: je mehr erkannt werden kann,
  - desto mehr mögliche Äußerungen von Nutzern sind abgedeckt
  - desto wahrscheinlicher kommt es zu Fehlerkennungen

Selber machen

- überlegen Sie sich einen Kontext:
  - z.B. Tresen vor der Eisdiele:
    d(ie|er) Verkäufer[in] hat gerade gefragt "Was möchtest Du?"

1. überlegt, was ihr in der Situation sagen würdet

2. schreibt (strukturell und inhaltlich) unterschiedliche Antworten auf

3. bildet Regeln, welche die möglichen Antworten in diesem Kontext ermöglichen (aber möglichst wenig anderes)

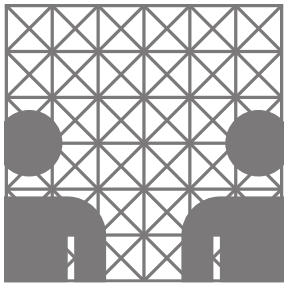
4. erweitert oder beschränkt ggfs. die Regeln um eine realistische Abwägung zwischen Abdeckung aller möglichen und Einschränkung auf wahrscheinliche Äußerungen zu erreichen

Vielen Dank.

baumann@informatik.uni-hamburg.de

https://nats-www.informatik.uni-hamburg.de/SDS20

Universität Hamburg, Department of Informatics
Language Technology Group



Notizen

Further Reading

- Speech Synthesis in General:
  - P. Taylor (2009): *Text-to-Speech Synthesis*. Cambridge Univ Press. ISBN: 978-0521899277. InfBib: A TAY 43070 (accessible introduction to the topic)
  - Rabiner & Juang (1993): *Fundamentals of Speech Recognition*. Prentice Hall. Stabi: A 1994/994. (in-depth mathematical approach)
  - Dong Yu, Li Deng (2015): *Automatic Speech Recognition: A Deep Learning Approach*. Springer. InfBib: A AUT 51465 (NN-based methods)
- The MaryTTS Speech Synthesis System:
  - Schröder & Trouvain (2003): "The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching", *Int. J. of Speech Technology* **6**(3).

Desired Learning Outcomes

- Ziel der Sprachsynthese ist es, die natürliche Varianz von Sprache zu erzeugen
  - dies ist das Gegenteil vom Ziel der Spracherkennung, die versucht Varianz aufzulösen!
- Probleme/Ambiguitäten linguistischer Vorverarbeitung:
  - Aussprachevarianten
  - Prosodie und Informationsstruktur sowie Emotionalität
  - Synthesetechniken: Formant- und Diphonsynthese