

# Vorlesung

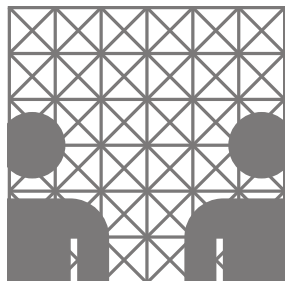
# Sprachdialogsysteme

Timo Baumann  
[baumann@informatik.uni-hamburg.de](mailto:baumann@informatik.uni-hamburg.de)



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

Universität Hamburg, Department of Informatics  
Language Technology Group



# Heute

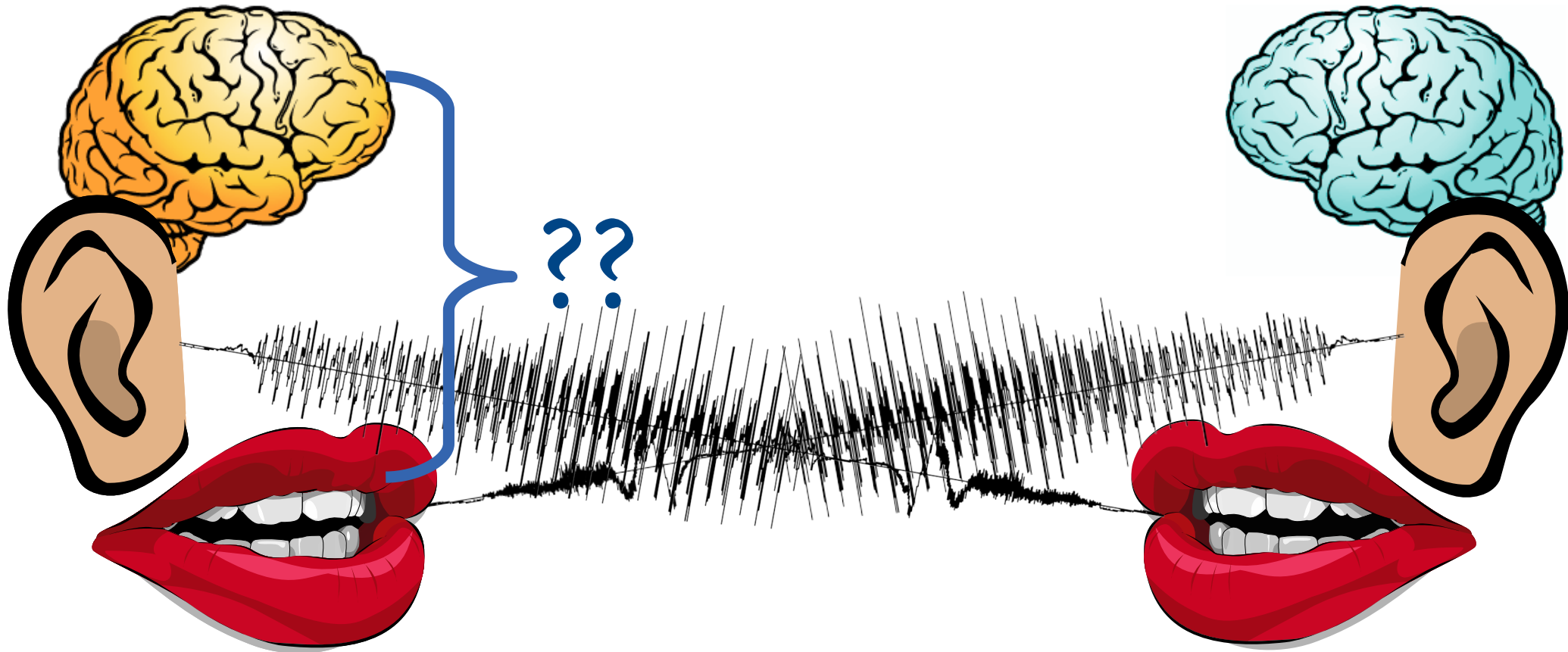
- Rückblick: Dialog → wer redet wann?
- Dialogsysteme: die traurige(?) Realität

Dialog: ein komplexes, interaktives System

# Dialog (vereinfacht)

Dialogagent

Dialogagent



# Turn-taking

- Fragestellung: wer spricht wann im Dialog?
  - „who holds the floor“ (Bühnensprache)
    - task heißt *floor-tracking* oder *end-of-turn-detection* oder einfach *endpointing*
- Aufgabenstellung: herausfinden ob der derzeitige Sprecher fertiggesprochen hat / ob es okay ist anzufangen zu sprechen

## Die Vielfalt an Turn-taking-Signalen:

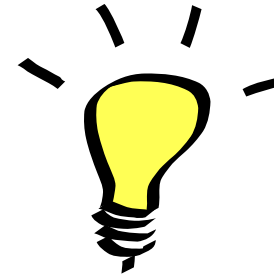
Sammeln Sie Indikatoren die darauf hinweisen,  
dass Ihr Redebeitrag abgeschlossen ist /  
ihr Zuhörer nun antworten darf.

Auf welcher sprachlichen Ebene verorten Sie sie jeweils?

# Modellierung eines Dialogagenten

# Leseaufgabe





find message that describes idea

**pragmatics** recover idea described by message

determine structure to convey meaning

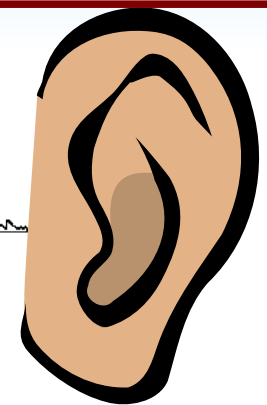
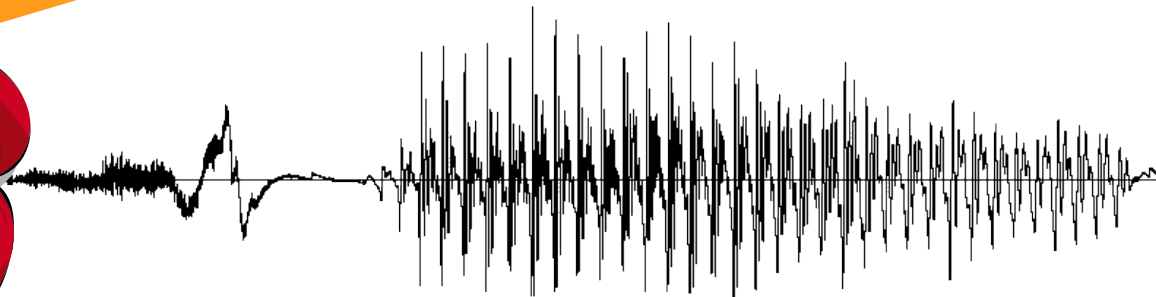
**semantics/  
lexicology** determine meaning of structure

sequentialize structure to word stream

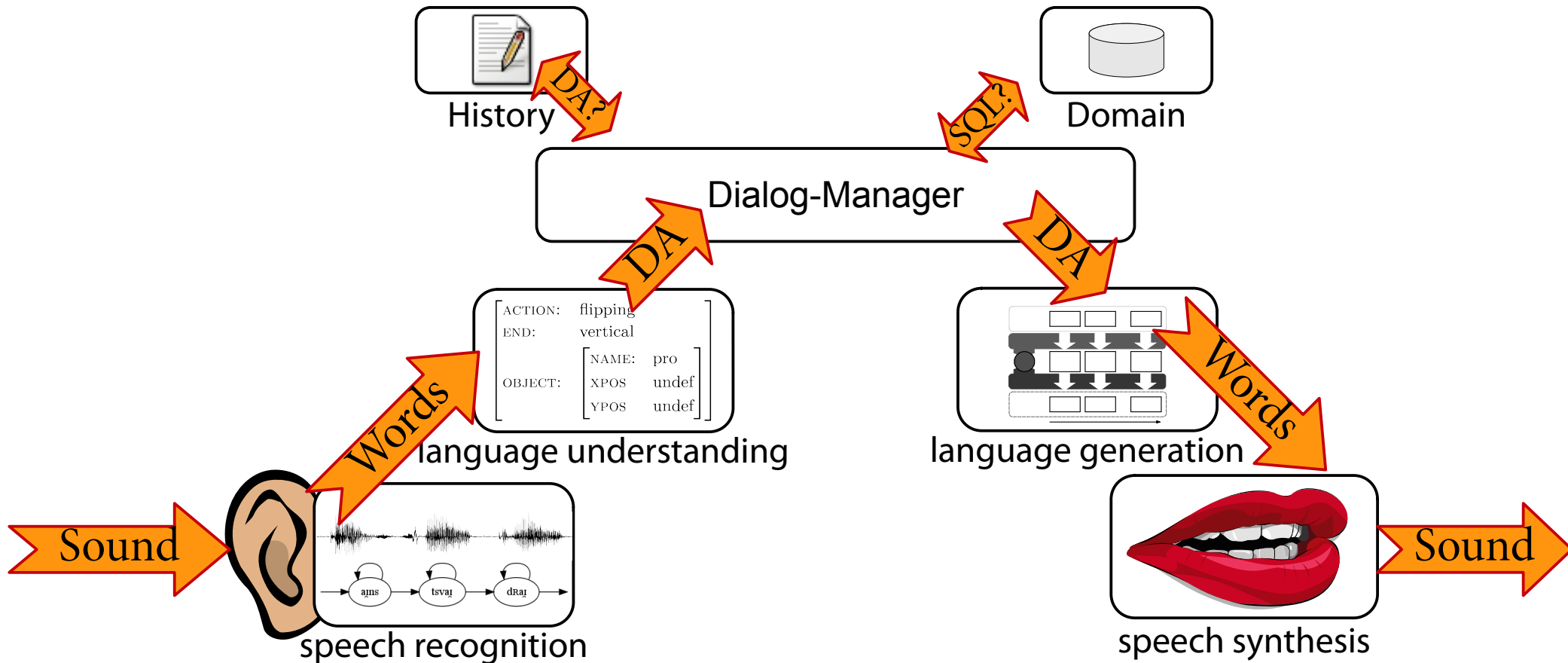
**syntax/  
morphology** recover structure of sequence

represent words through sounds

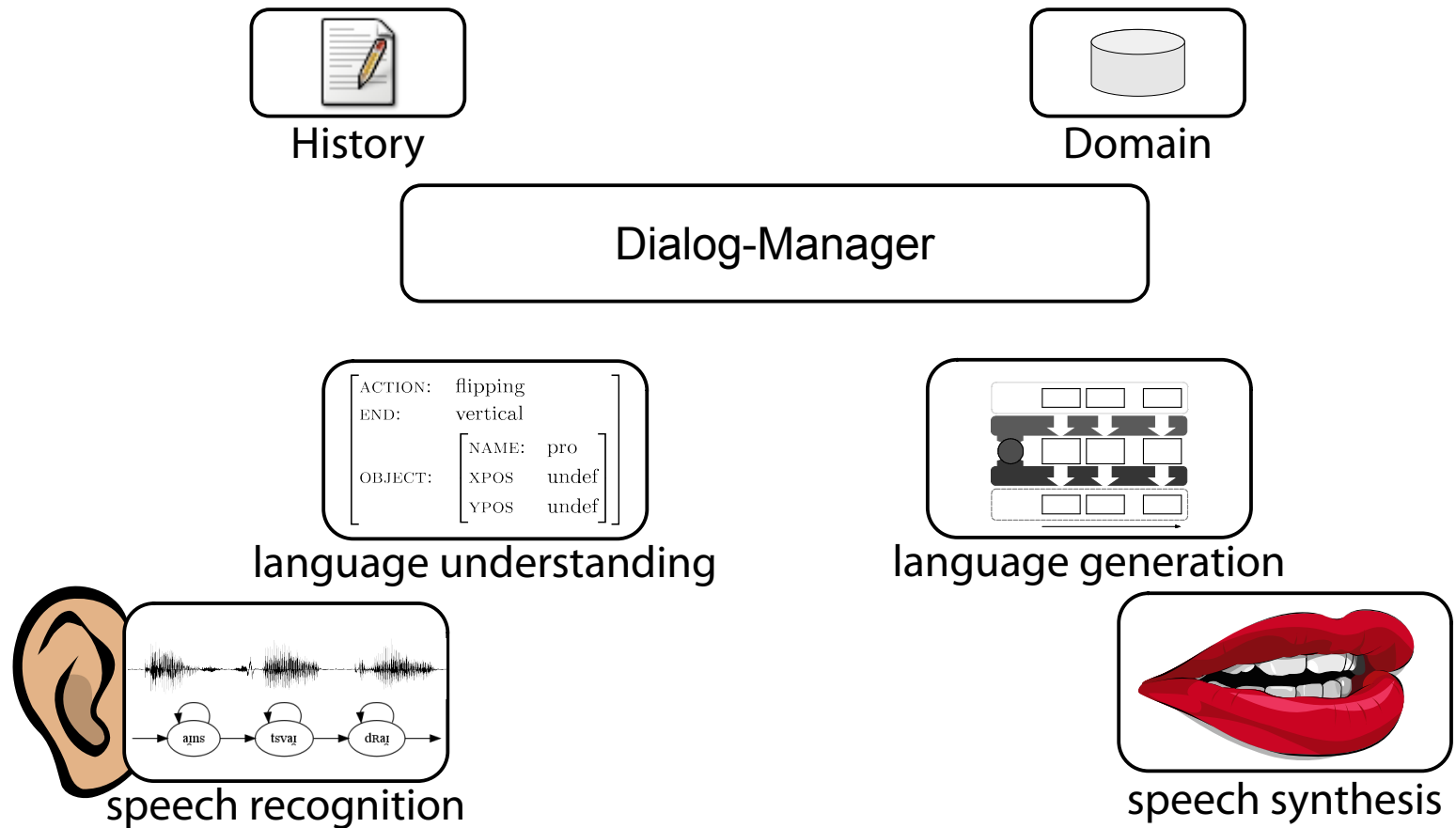
**phonology/  
phonetics** recombine sounds to words



# Modellierung eines Dialogagenten



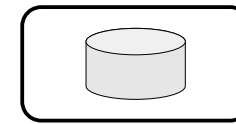
# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



History

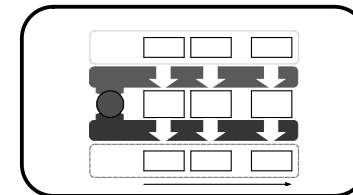


Domain

Dialog-Manager

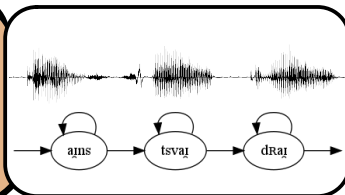
```
[ ACTION:  flipping  
  END:    vertical  
  OBJECT: [ NAME:  pro  
           XPOS: undef  
           YPOS: undef ] ]
```

language understanding



language generation

Sound

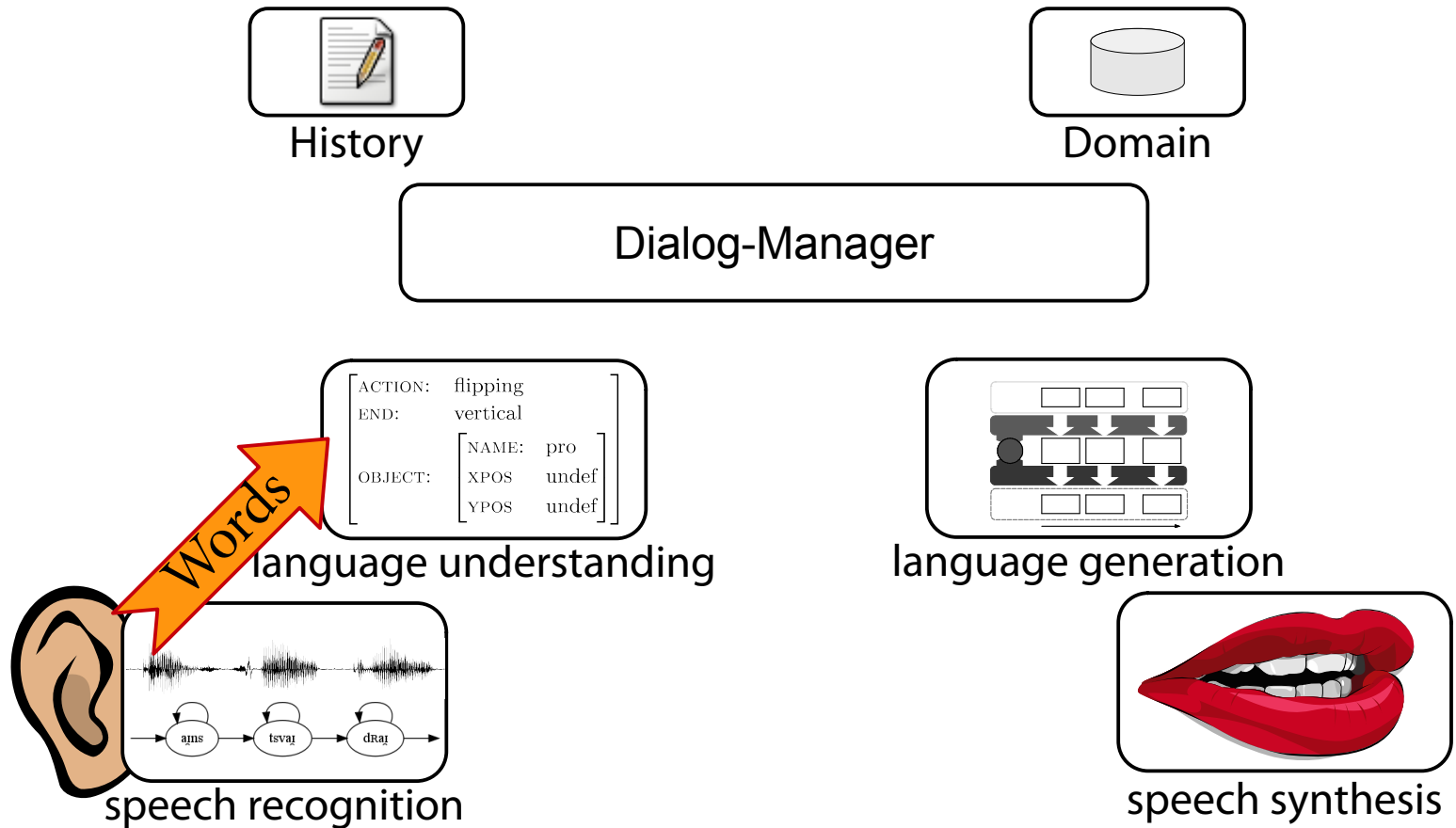


speech recognition

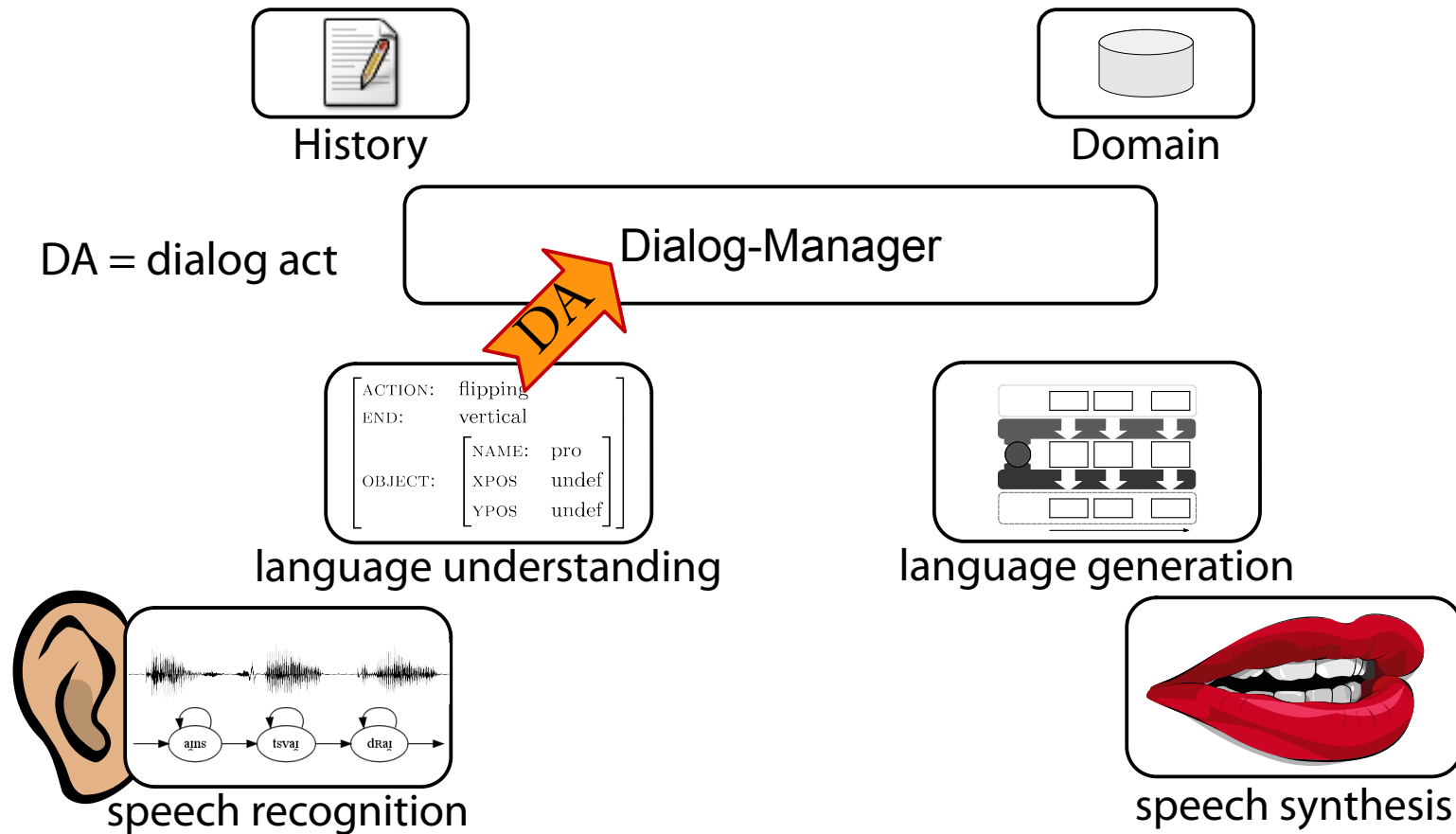


speech synthesis

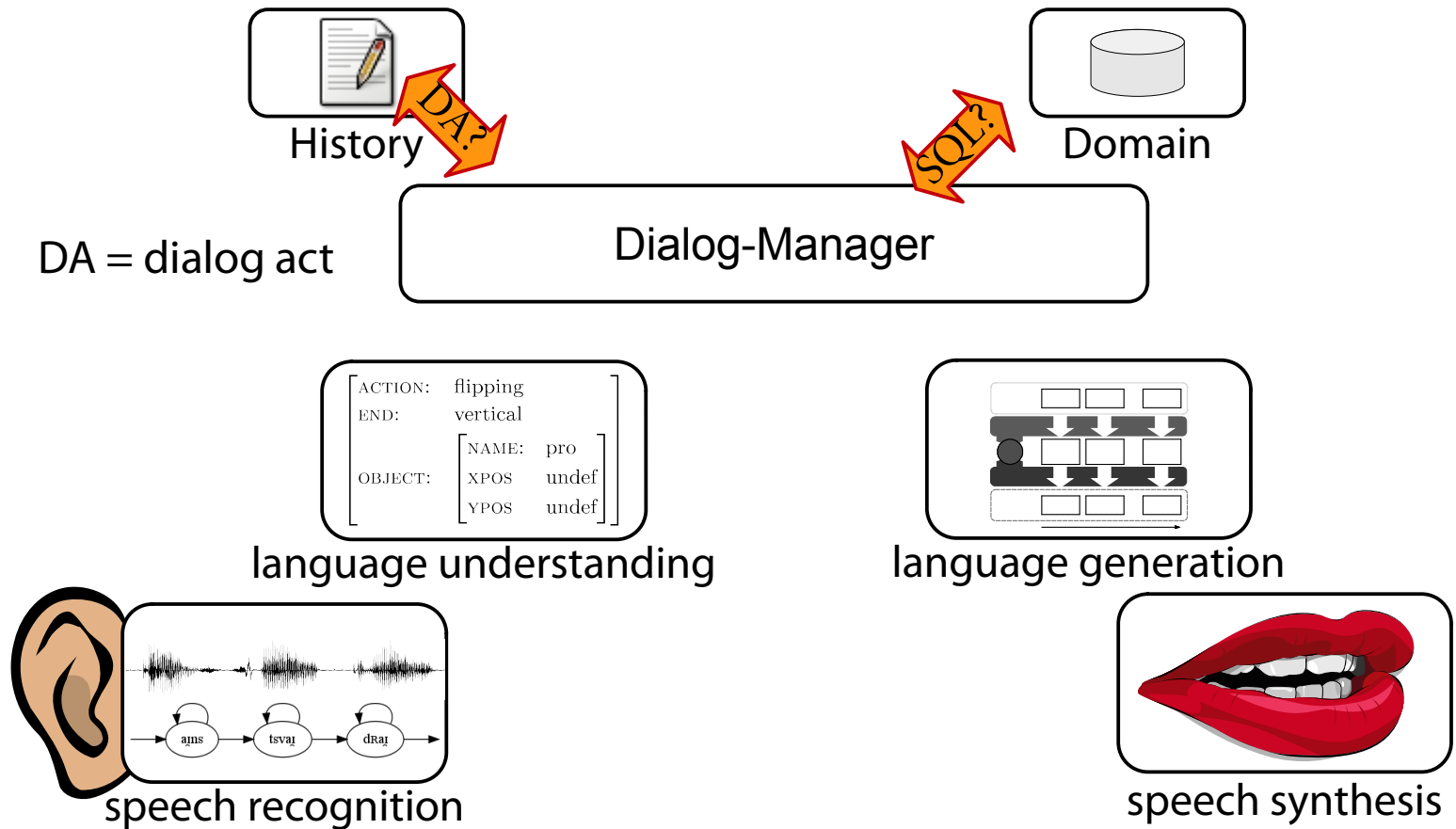
# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



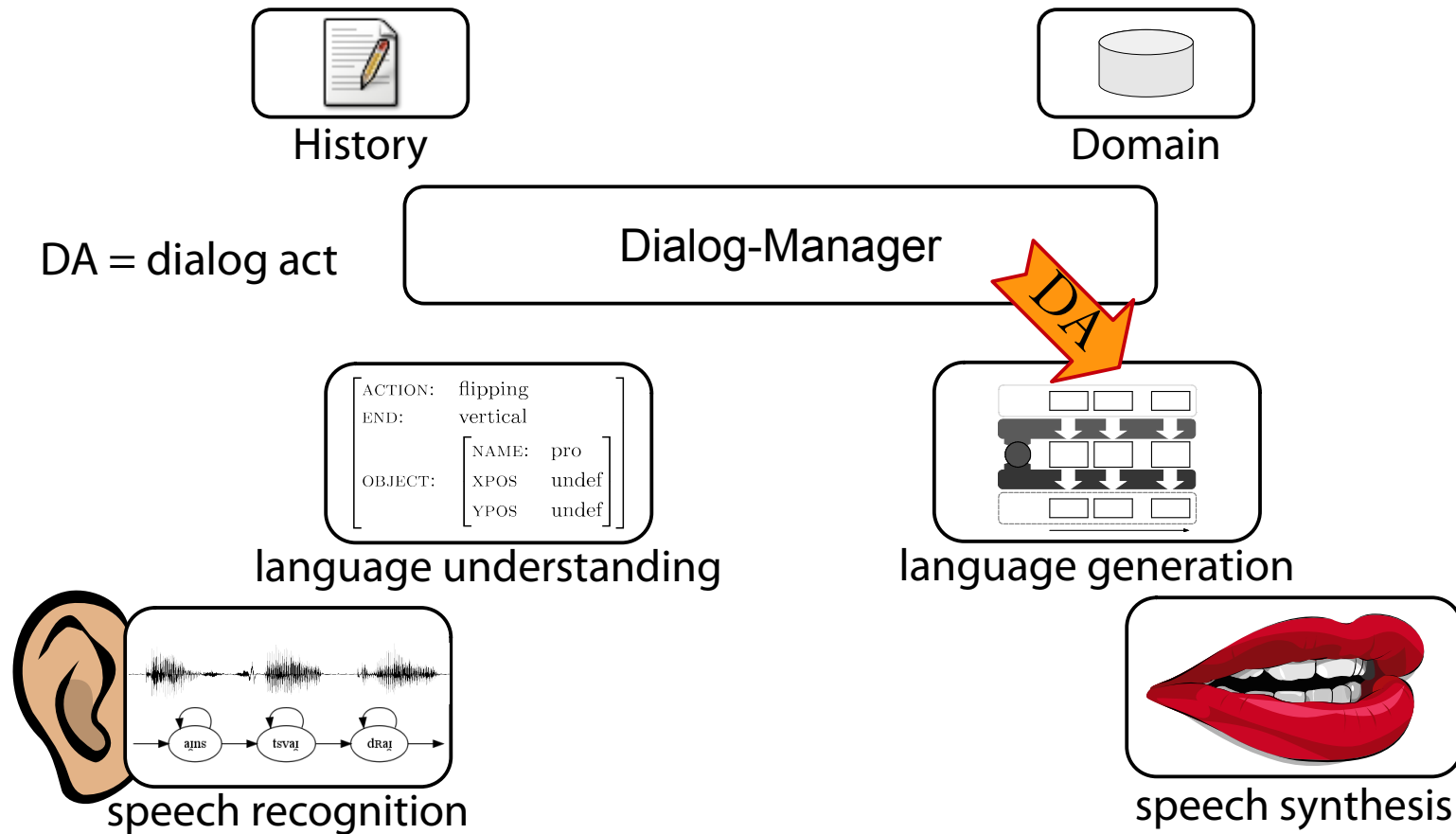
# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)

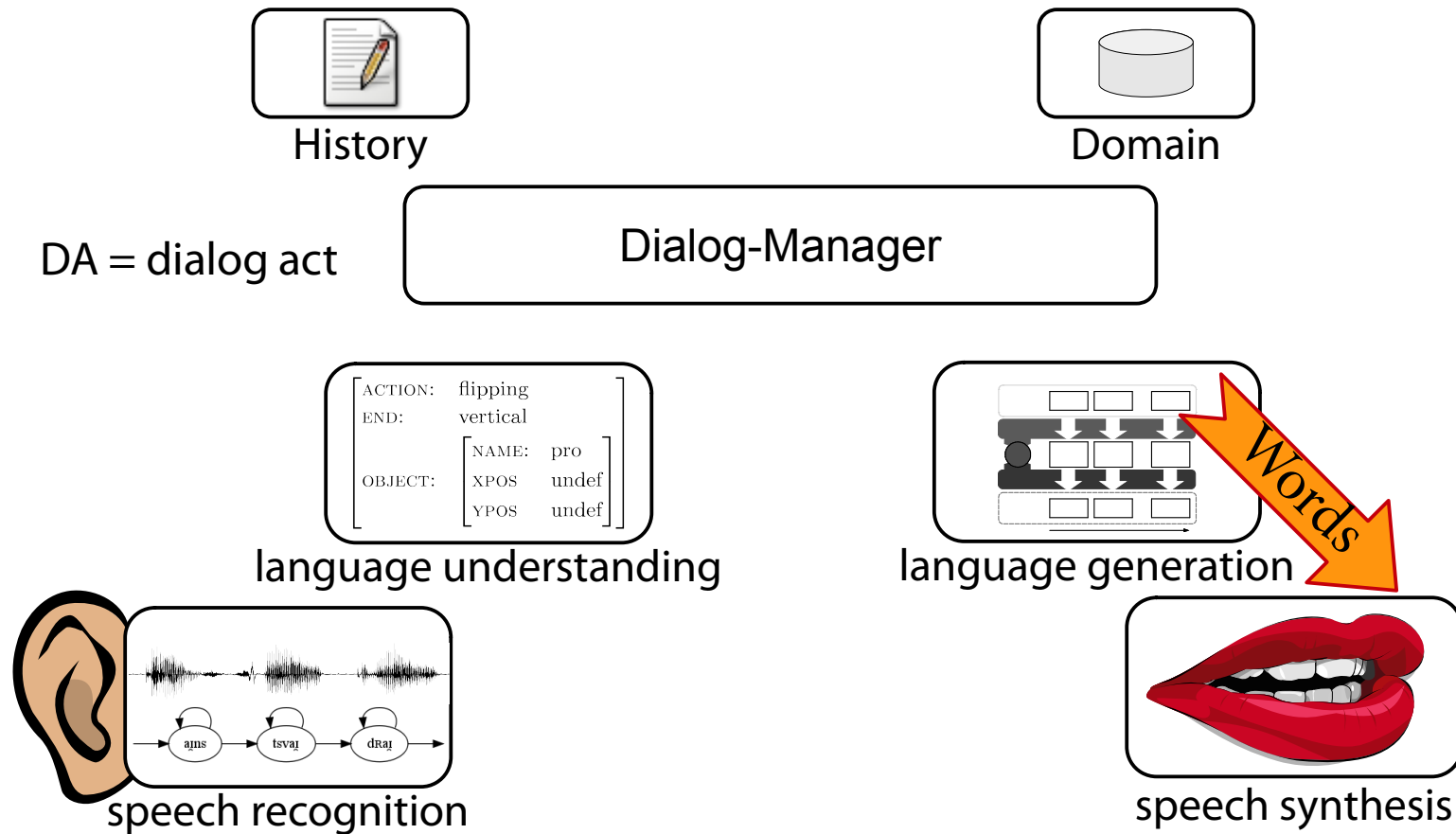


# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)

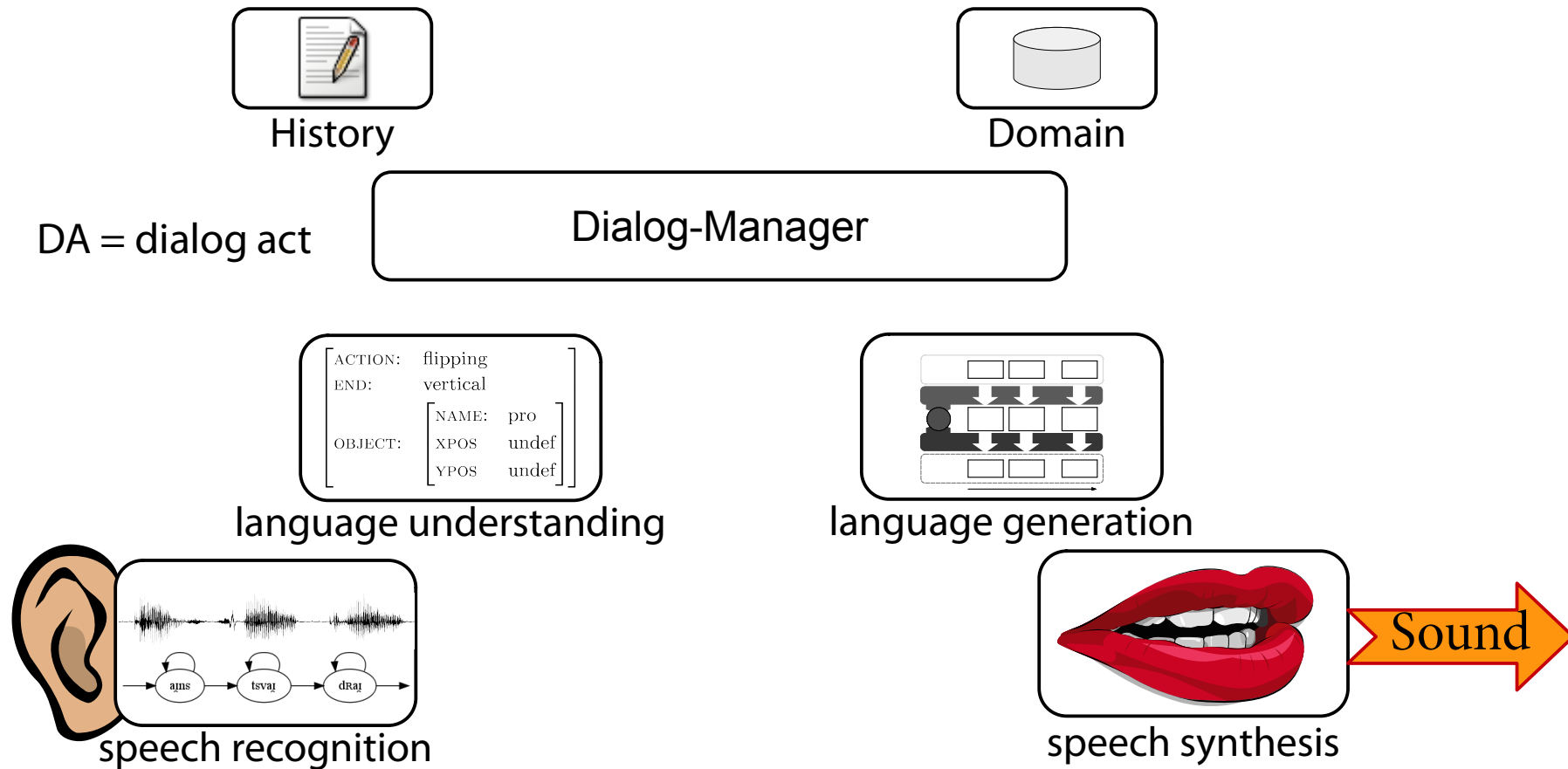




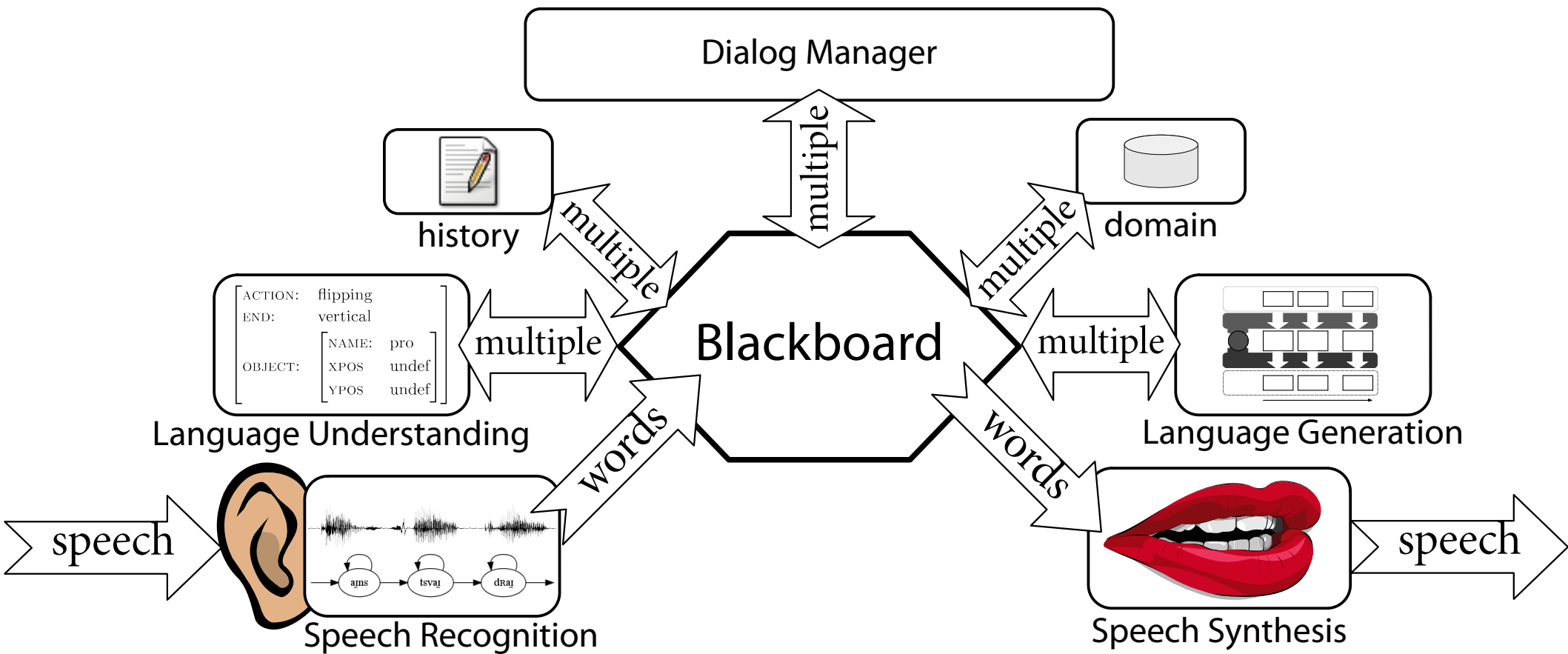
# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



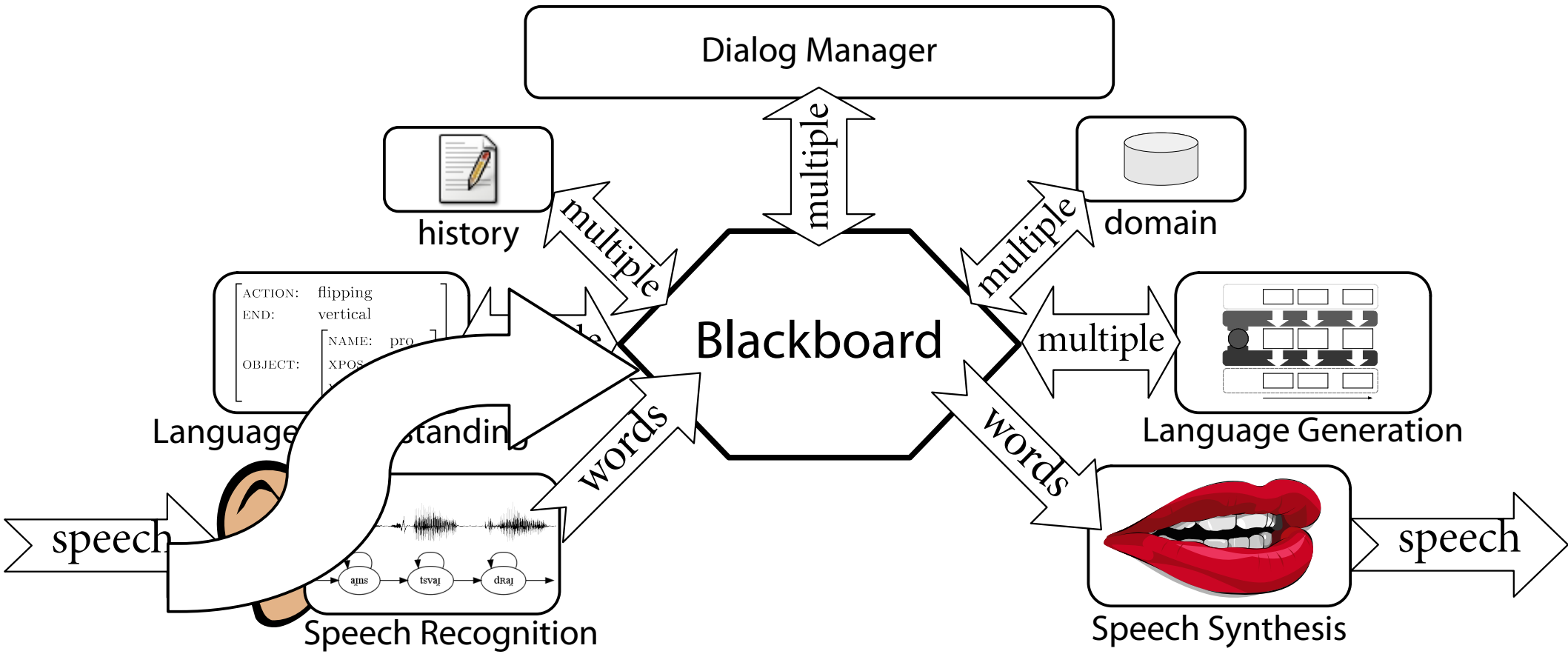
# Ein einfacher Dialogagent (Wasserfallmodell / Pipelinemodell)



# Blackboard-basierte Architektur



# Blackboard-basierte Architektur



## Reduktionismus vs. Konnektionismus

Sammeln Sie Argumente für und gegen stark verknüpfte Dialogsystemarchitekturen.

Welche Verknüpfungen sind zwingend?  
welche sind wünschenswert?  
Welche Nachteile ergeben sich?

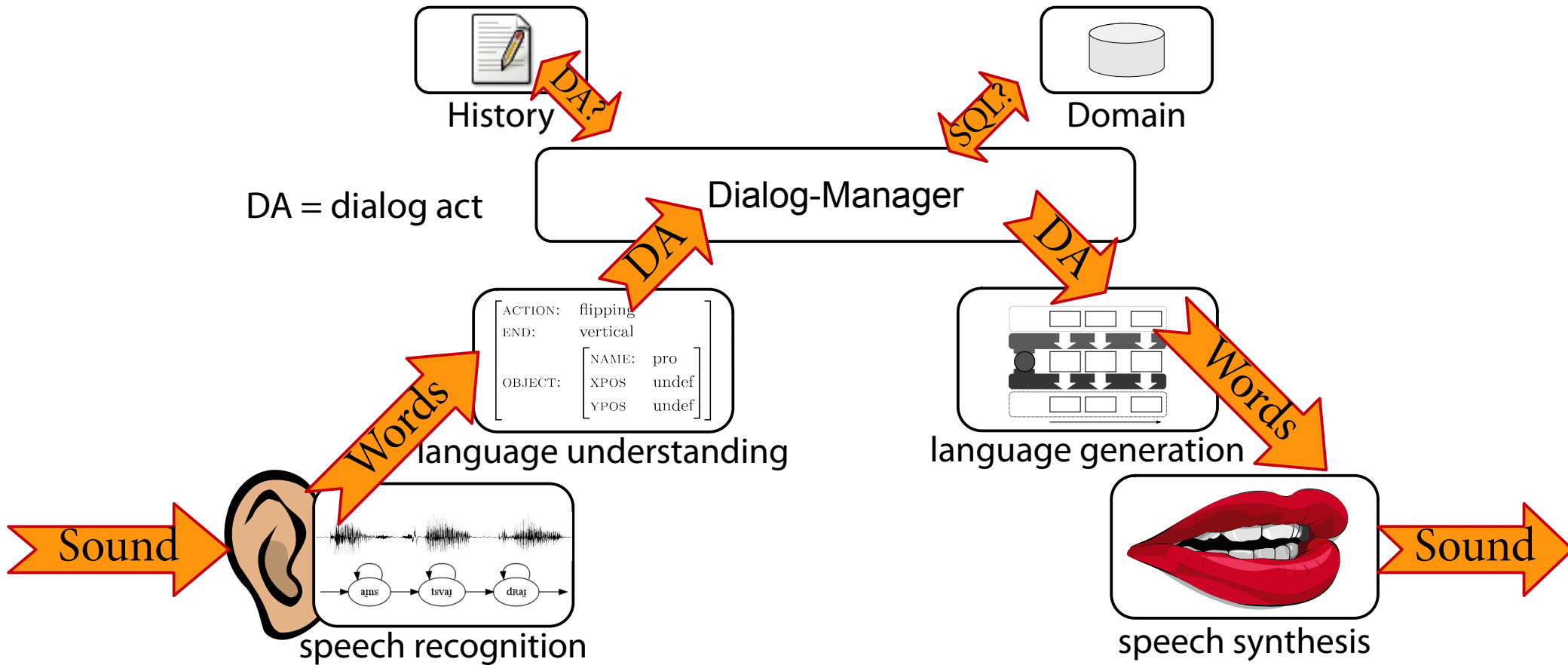
# Pipeline

vs.

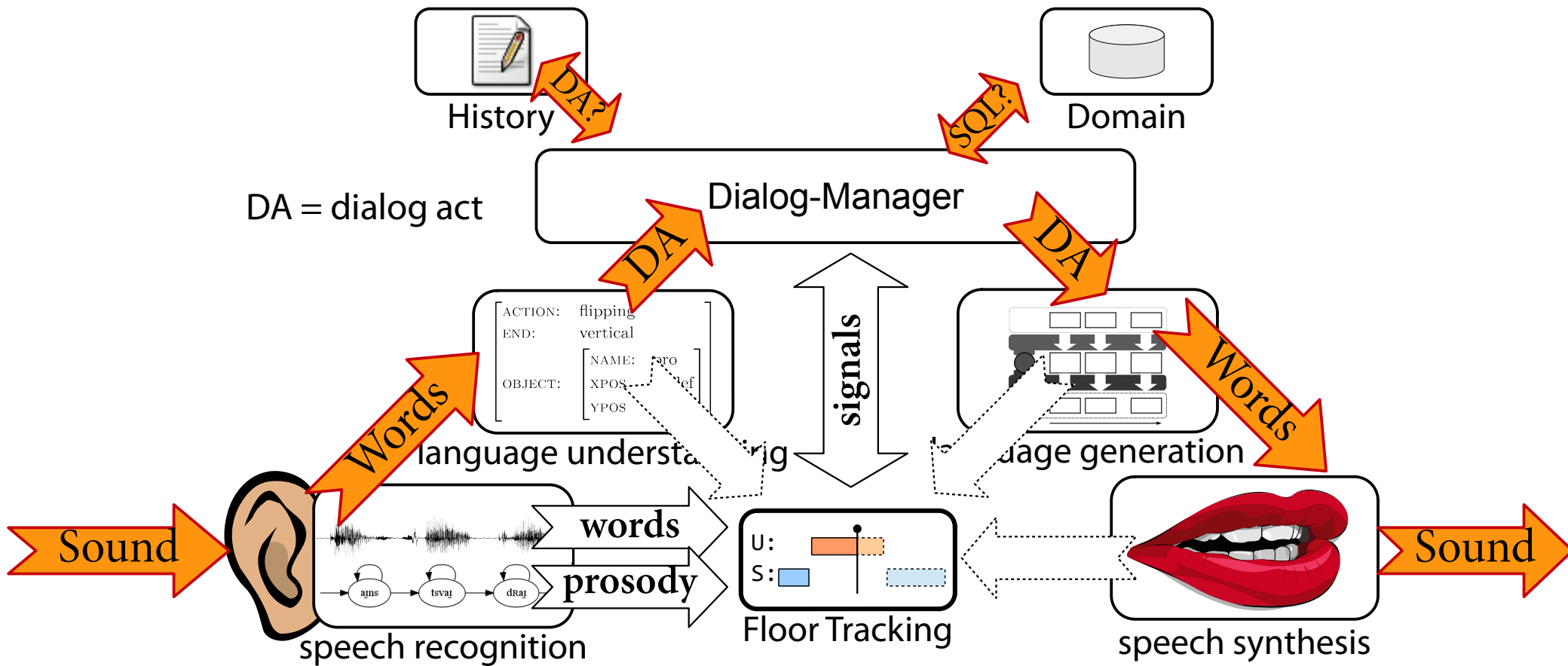
# Blackboard

- konzeptuell (sehr!) einfach
  - Jedes Modul hat genau einen Inputtyp und Outputtyp
  - Verwendbarkeit bestehender Module
  - Nebenläufigkeit ist einfach
  - löst das Problem nur unvollständig
- konzeptuell einfach (aber komplexe Interaktionen!)
  - Module können Output aller anderen nutzen
  - bedingte Wiederverwendbarkeit bestehender Module
  - Nebenläufigkeit ist schwierig
  - löst das Problem (im Prinzip)

# Wo ist das Floor-Management?

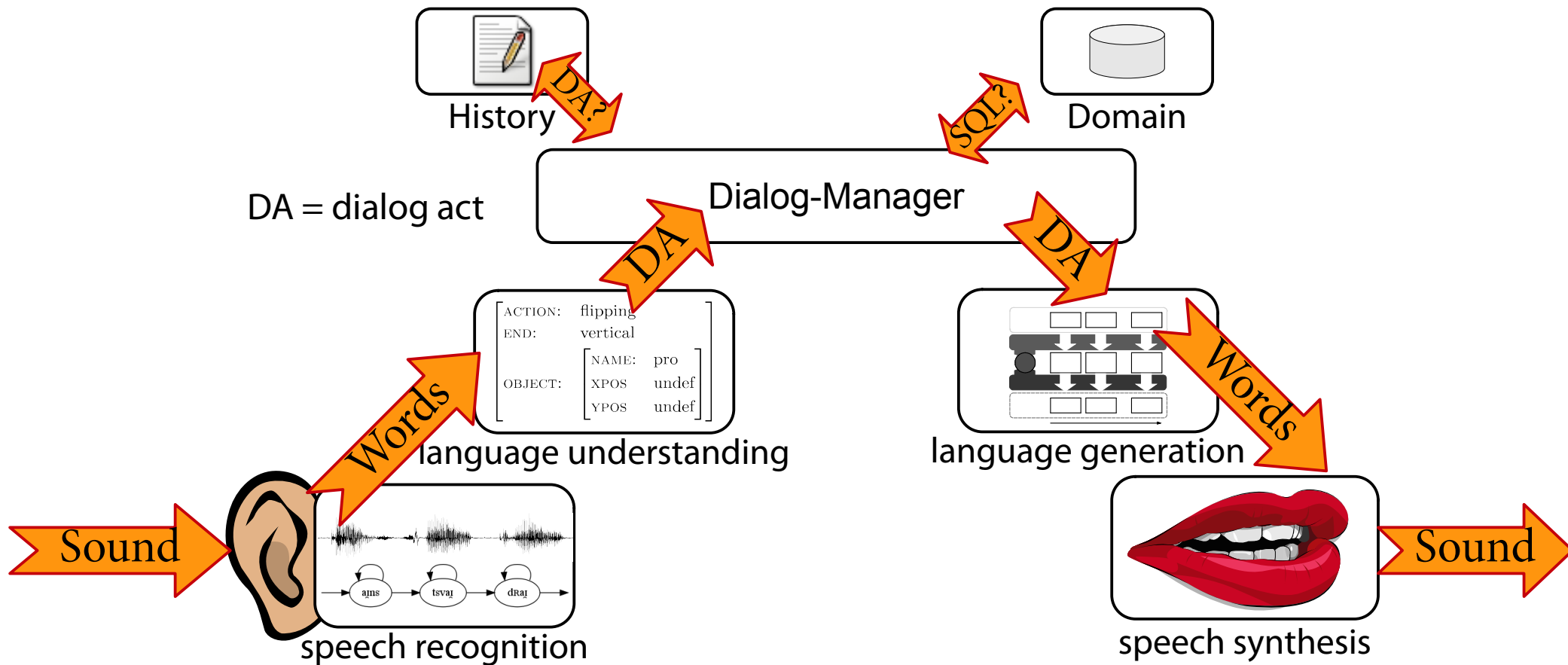


# Wo ist das Floor-Management?





# Oder soll man es bleibenlassen?



Umsetzung: fixer timeout (z.B. 0,3 Sekunden) bestimmt Ende der Äußerung.

Wieso funktionieren  
Dialogsysteme überhaupt?

# Wieso funktionieren Dialogsysteme überhaupt?

- Dialoginteraktion ist sehr robust:
  - Menschen gleichen Unzulänglichkeit des Interaktionspartners **bis zu einem gewissen Grad** aus
  - Unzulänglichkeit eines Aspekts (wie z.B. Turn-taking) schlägt nicht unbedingt direkt (bzw. linear) auf die Systemperformance durch
- Systemtheoretische Betrachtung:
  - Gesamtperformance ist NICHT Summe der Teilperformances
  - stattdessen: komplexe Interaktion zwischen den Teilmodulen des sprachlichen Systems
  - mit manchen Sachen kommt man noch “relativ ungeschoren” davon, an anderen Stellen können kleine Änderungen große Effekte haben
- Interaktionsmodell des Dialogsystems ist “schlimm” aber nicht “zu schlimm”
  - Mensch passt sich an die langsame Ping-Pong-Interaktion an (zum Beispiel: kein Feedback)
  - Mensch spricht deutlicher und einfacher um verstanden zu werden
  - ...

# Zusammenfassung

- die meisten Dialogsysteme sind *modular* and *pipeline-basiert* (möglich: einige stark verknüpfte Module, die miteinander “mini-Blackboards” nutzen)
- die meisten Systeme nutzen nur sehr offensichtliche turn-taking-Signale: sprechen wenn es lange (z.B. 0,3 Sekunden) keiner gesprochen hat, aufhören zu sprechen sobald jemand spricht
- Turn-taking ist ein komplexes Phänomen, nutzt Merkmale aus allen(?) linguistischen Teilsystemen
- Turn-taking ist sehr robust: Attraktion des Systemzustands zu stabilen Zuständen
  - Dialogsysteme nutzen diese Robustheit schamlos aus



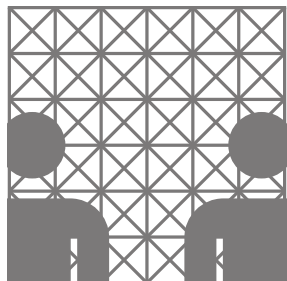
Vielen Dank.

[baumann@informatik.uni-hamburg.de](mailto:baumann@informatik.uni-hamburg.de)



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

Universität Hamburg, Department of Informatics  
Language Technology Group



# Notizen

# Further Reading

- Introduction to Dialogue and Linguistics:
  - the relevant chapters in: Jurafsky and Martin (2009): *Speech and Language Processing*. Pearson International. InfBib: A JUR 4204x.
  - Jokinen and McTear (2010): *Spoken Dialogue Systems*. Synthesis Lectures on HLT. InfBib:
- Systems theoretic views on complex systems in general and on language in particular:
  - Bertalanffy (1972): „The History and Status of General Systems Theory“. In: *The Academy of Management Journal* 15(4), pp. 407-426. via Google Scholar.
  - Larsen-Freeman and Cameron (2008): *Complex Systems and Applied Linguistics*, Oxford University Press. StaBi: A 2009 / 7836.
- Critical views on machine learning for building complex systems:
  - Sculley et al. (2014): „Machine Learning: The High Interest Credit Card of Technical Debt“, Software-Engineering for Machine Learning Workshop at NIPS 2014, <http://research.google.com/pubs/pub43146.html>



# Desired Learning Outcomes

- dialog is – to some extent – modular and SDSs mimick this modularity
  - turn-taking cannot easily be allocated to a „module“ but it emerges from the interaction
  - this is conventionally ignored for SDSs, yet human-agent interaction still works because of attraction / accommodation
- students grasp the idea of emergence in complex systems and attraction as a principle to control such systems