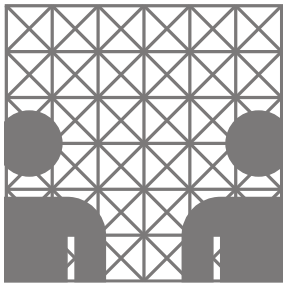
Vorlesung

Sprachdialogsysteme

Timo Baumann
baumann@informatik.uni-hamburg.de

https://nats-www.informatik.uni-hamburg.de/SDS19

Universität Hamburg, Department of Informatics
Language Technology Group



Heute

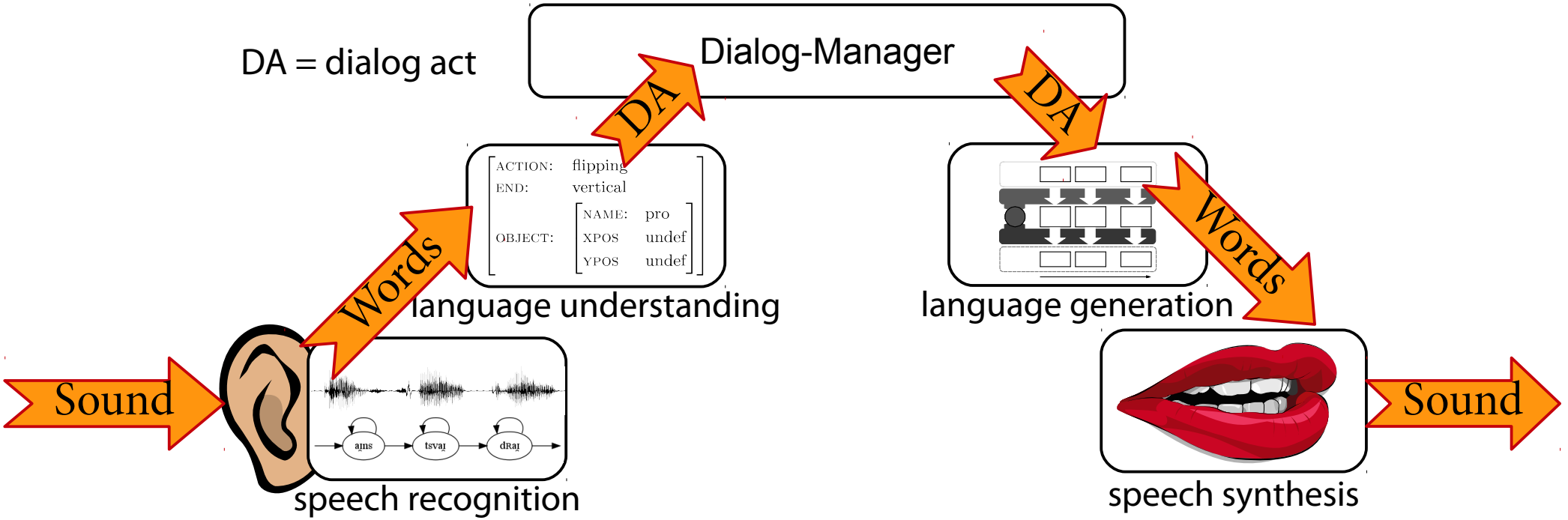
Tolle Dialogsysteme?

Reprise zum Dialogmanagement

Tolle Dialogsysteme

mit DialogOS

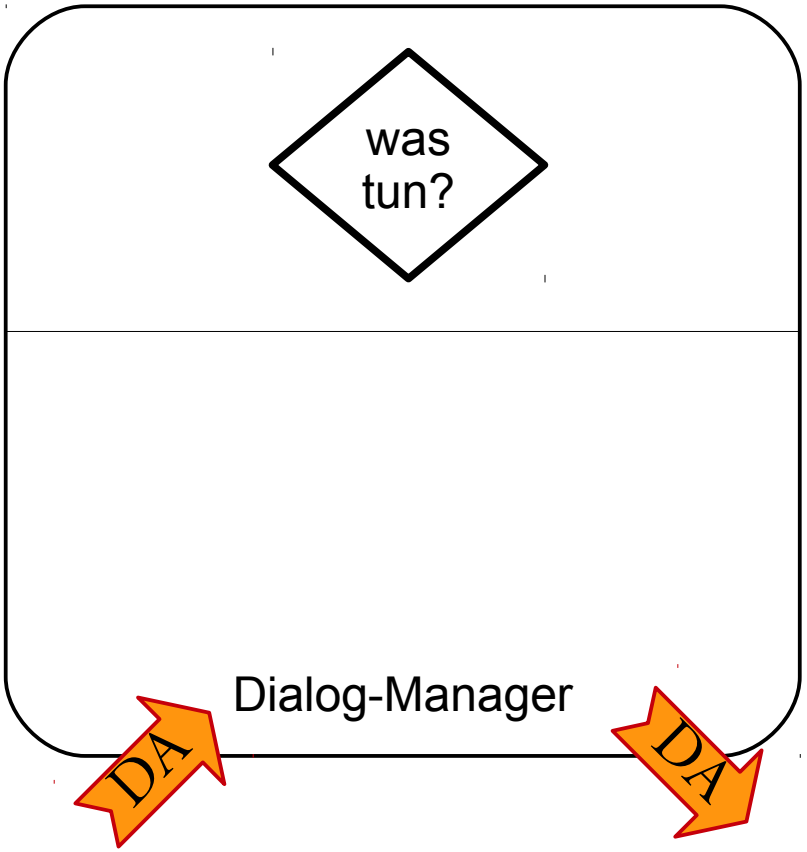
Ein einfacher Dialogagent

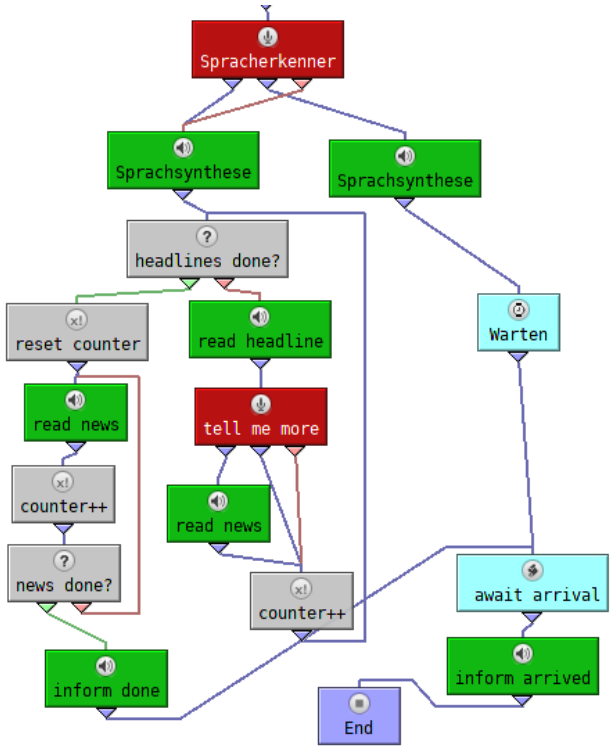


Komponenten im Dialogmanager

Dialogpolicy

Dialogzustand



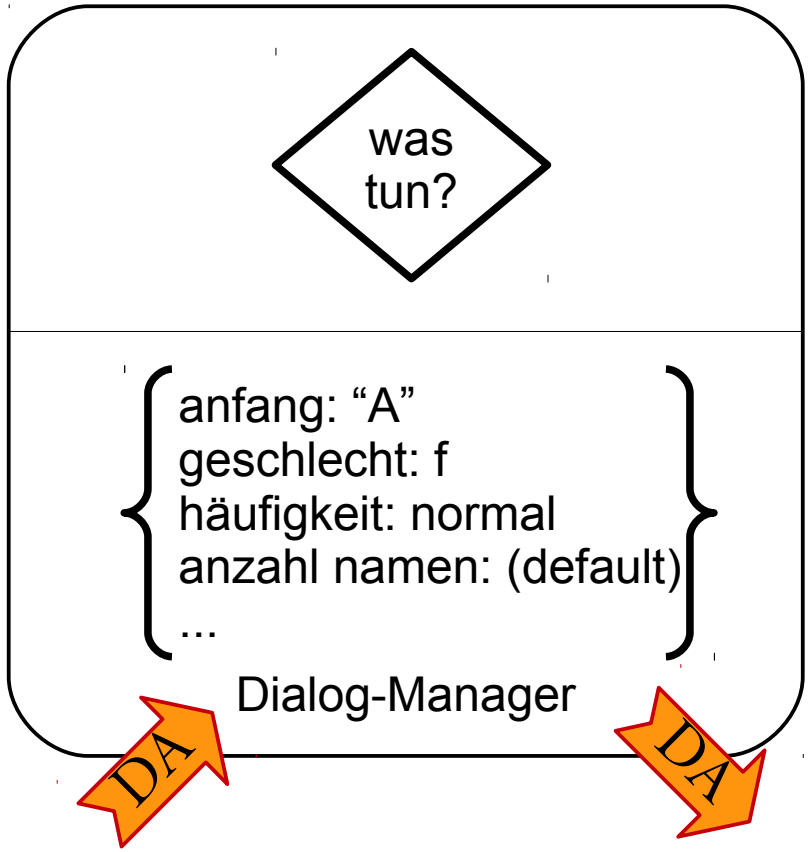


endlicher Automat

Komponenten im Dialogmanager: Frame-basiertes Management

Dialogpolicy

Dialogzustand



Vorteile/Nachteile

endlicher Automat

Frame-basiert

FSM-basiertes DM

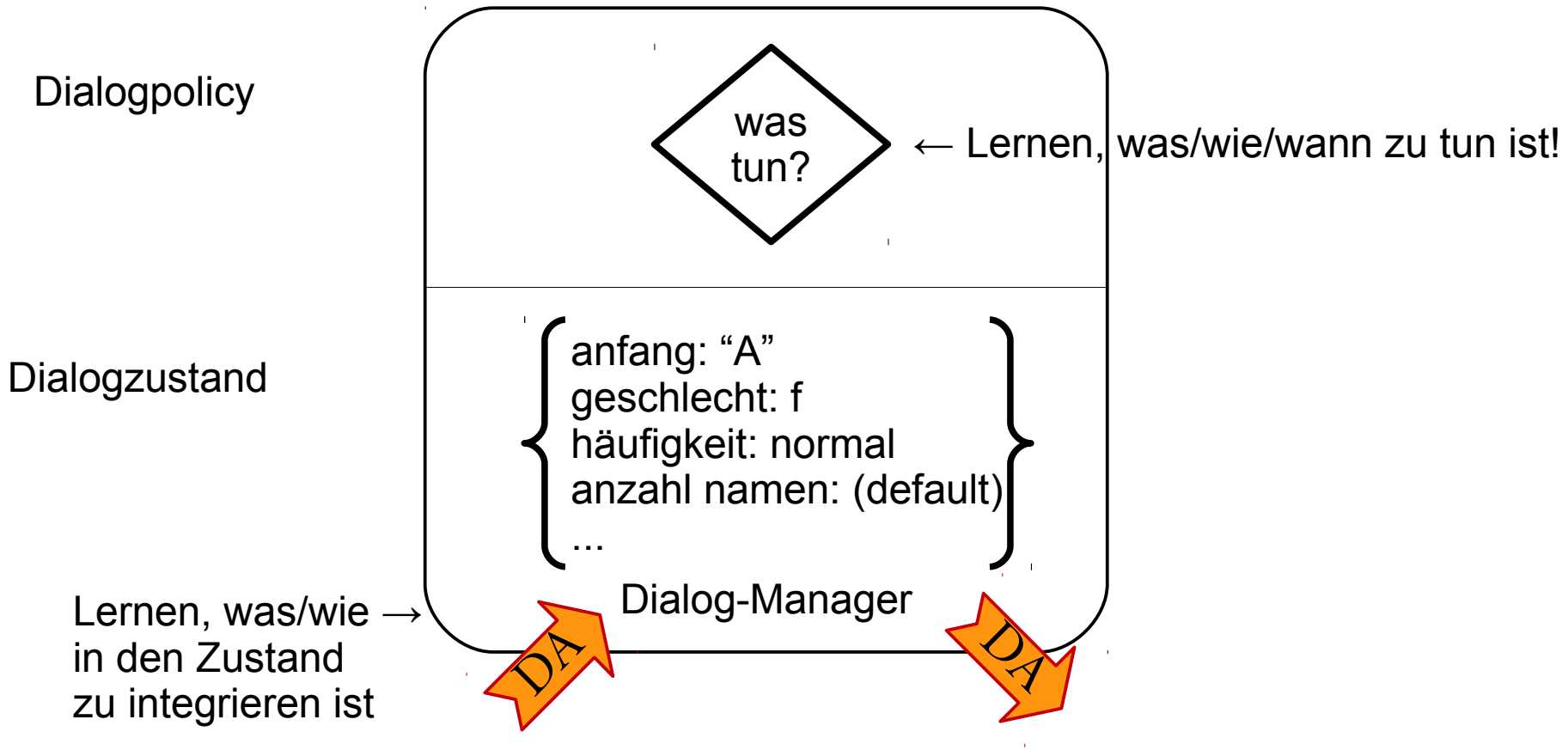
- spezialisierte State-Machine (kann auf Spezifika der Dialogdomäne eingehen)
- aufwendige Fallunterscheidungen
- Vorhersehen des Nutzerverhaltens

Frame-basiertes DM

- generische State-Machine (aufwendig zu erstellen, dann aber sehr praktisch)
- keine fachspezifischen Abweichungen möglich

- Unsicherheit die sich aus möglichen Verarbeitungsfehlern ergibt
- kein Umgang mit Vagheit sprachlicher Äußerungen
- keine Anpassung an den Nutzer
- kein “Lernen” nützlichen Verhaltens

Probabilistisches Dialogmanagement



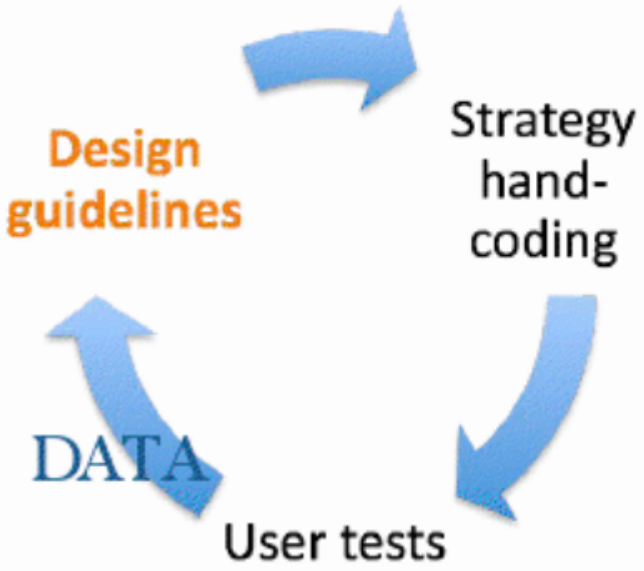
Allgemeines Vorgehen zum prob. DM:

- automatische Optimierung der Policy (maschinelles Lernen)
- basierend auf Interaktion mit echten oder simulierten Nutzern
- mittels reinforcement learning (ein Verfahren für Sequenzlernprobleme)

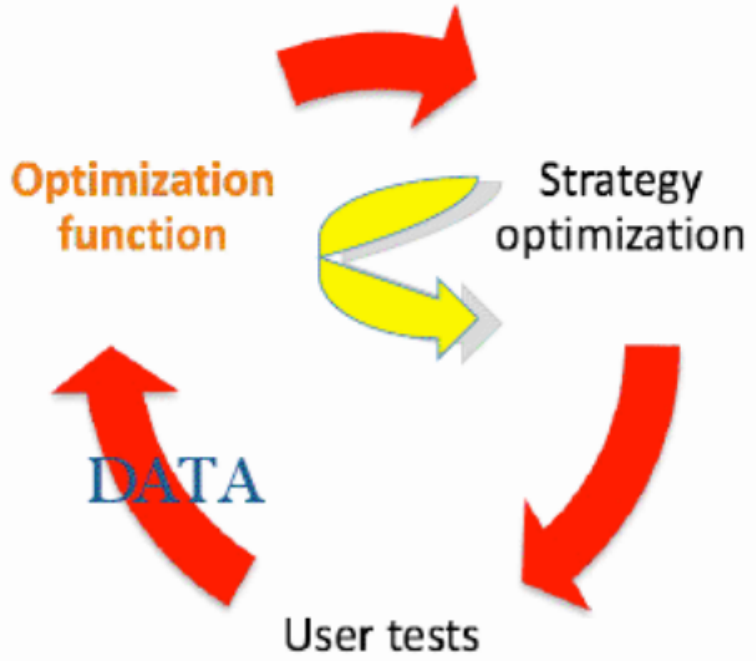
- am Anfang: "dumme" Dialog-Policy,
- Interaktion mit Nutzern, darauf basierend positives/negatives Feedback ("rewards")
- Identifizierung von Fehlern in der Policy (warum ging es schief?)
- Korrektur der Fehler in der Policy
- repeat

Vergleich der Strategien

Conventional software life cycle



Automatic strategy optimisation



Design by "Best practices"

(Paek 2007)

Automatic design by optimization function

(= “programming by reward”)

[slide borrowed from O. Lemon]

Vorgehen

- *Planung*sproblem charakterisiert durch:
  - Zustandsraum (alle möglichen Dialogzustände)
  - Aktionsraum (alle möglichen Systemäußerungen)
  - Ziele (kodiert durch Rewards)
- Markov-Decision-Process
  - Zustand s aus S
  - Aktion a aus A
  - policy: p: S -> A

    probabilistisch, also gewichtete Zufallsauswahl;
    dann kann man die Gewichte lernen

Auswahl im Aktionsraum

- zunächst ein Klassifikationsproblem
  - Dialogzustand enthält notwendige Information
  - mögliche Systemantworten sind bekannt
  - Schwierigkeit vollständige Trainingsdaten zu bekommen
  - häufig gibt es mehrere mögliche nächste Aktionen
- sich widerstrebende Ziele:
  - möglichst sicher das richtige Flugticket buchen
    = den Nutzer sicherlich richtig verstanden haben
  - möglichst schneller und einfacher Dialog ohne unendlich viele Rückfragen
  - Reward = (korrekte Buchung ? +10 : -30) – Länge des Dialogs
  - man weiß erst am Ende ob das Ziel erreicht ist

Herausforderungen

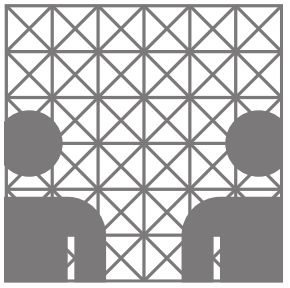
- Trainingsdaten
  - bzw. “Trainer”, der auf Systemfehler eingehen muss
  - man kann nicht einfach aus fertigen Dialogen lernen
  - häufig Rückgriff auf “simulierte Nutzer” (=einfache, regelbasierte Systeme)

Vielen Dank.

baumann@informatik.uni-hamburg.de

https://nats-www.informatik.uni-hamburg.de/SDS19

Universität Hamburg, Department of Informatics
Language Technology Group



Notizen

Further Reading

- Sprachverstehen:
  - Gokhan Tur & Renato De Mori: Spoken Language Understanding, insbesondere Kapitel 3.
  - Jurafsky & Martin, Chapter 26, Sections 3 und 4

Desired Learning Outcomes

- Die Studierenden können Frames entwerfen, die mögliche Sachverhalte in bestimmten Situationen abbilden
- Die Studierenden verstehen die linguistische Unterscheidung zwischen Dialog-Handlungen und Frames