

Vorlesung

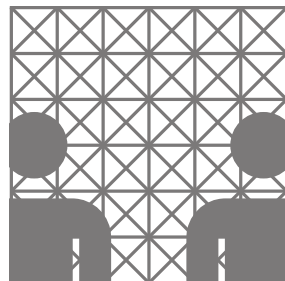
Sprachdialogsysteme

Timo Baumann
baumann@informatik.uni-hamburg.de



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

Universität Hamburg, Department of Informatics
Language Technology Group



Heute

Reprise Frame-Basierung

Frame-basiertes Dialogmanagement und VoiceXML

- Frames als zentrale Datenstruktur zur Verwaltung der Dialogsituation

Frame-Semantik

- Prädikate mit Argumenten
 - Prädikat bestimmt das generelle Ziel der Äußerung
 - Prädikat hat typische, zugeordnete Argumente
- *Luisa kauft Peter ein Eis.*
- Aufgabe für den Computer:
 - Prädikat bestimmen, passend dazu Argumente bestimmen
(aber das Prädikat erkennt man z.T. erst an den Argumenten)
 - umgekehrt: Text bestimmen gg. Prädikat und Argumente

Frame-Semantik und “Slots”

- Beziehungsgeflecht von Worten (und Konzepten) im Kontext
- *kaufen* im Kontext “kommerzielle Transaktion”. Slots:
 - Verkäufer
 - Käufer
 - Ware
 - Preis?
 - Datum?
 - Vertragsunterlagen?
- Perspektive: *kaufen* vs. *verkaufen* oder *schenken*
- Attribut-Werte können (zunächst noch) unbekannt sein

(echte) Frame-Semantik zur Generierung

- name[Aromi], eatType[restaurant], food[English], area[city centre] → “Aromi is an English restaurant in the city centre.”
- name[The Vaults], eatType[restaurant], food[Chinese] → ”A Chinese restaurant is The Vaults”
- name[The Mill], eatType[restaurant], food[English], priceRange[less than £20], area[city centre], familyFriendly[no], near[Raja Indian Cuisine]

auch viele andere Möglichkeiten, dasselbe auszudrücken.

typische Lösung: System generiert regelgeleitet aus Mustern oder mittels trainierten Wortsequenzmodellen

Dialog-Handlungen (Dialogue Acts)

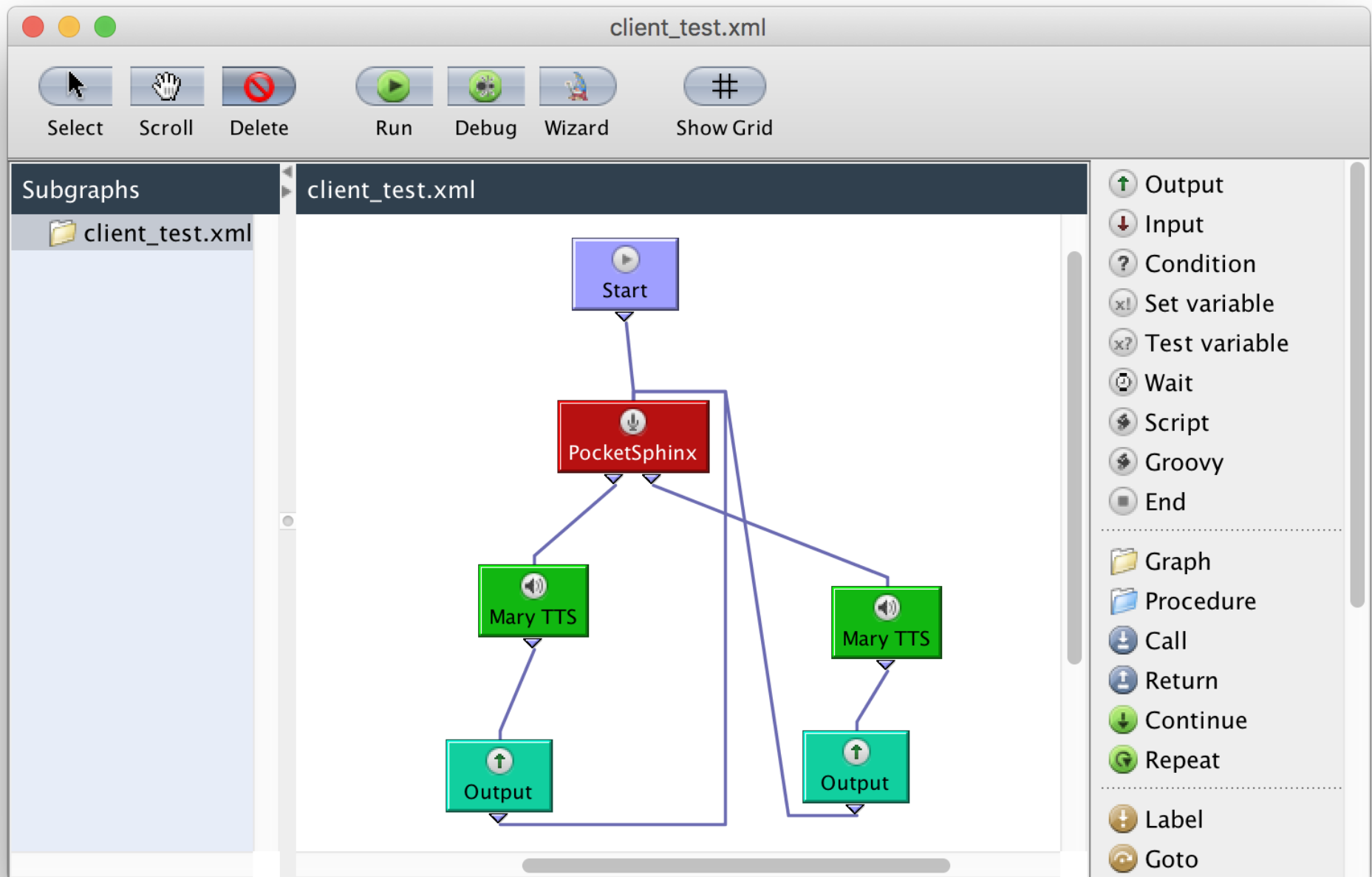
- Dialogäußerungen haben immer ein Ziel
 - Frame-Semantik *per se* hat kein Ziel
- Dialog-Handlungen drücken aus, was in Bezug auf den Frame erzielt werden soll, z.B.:
 - erbitten: “Wieviele Kugeln?”
 - informieren: “Ne Kugel Eis kostet ...”
 - Affirmation/Ablehnung: “ja”/“nein”
 - bestätigen: “Sie wollen also ...”
 - Auswahl: “Dann nehme ich ...”
 - ...
 - welche DAs ein System modellieren sollten, hängt von der Dialogsituation und -domäne ab

Dialog-Handlungen und Frames

- Handlung: *Typ* des Redebeitrags
- Frame: *Inhalt* des Redebeitrags

- Inhalt des Redebeitrags: “Luisa kauft Peter ein Eis.”
- Handlung: ?

Graph-basierte Dialogmodellierung



komplexeres Beispiel

The image displays a complex software development environment with three main components:

- Flowchart (Left):** A detailed flowchart titled "dialogos.txt" showing a sequence of operations. It starts with "Start" and "Beginn", followed by a loop involving "zaehler+1" and "zaehler". The flowchart includes various decision points (e.g., "None?", "Color?", "Form?") and actions (e.g., "Teilbeschreibung", "Success?", "Menge", "Farbe", "Form", "Menge", "Spalte", "Variable testen", "Spracherkennung", "Script").
- PentoWorld (Top Right):** A window titled "PentoWorld" showing a game interface. It features several colored shapes (blue, green, yellow, red) and a large grid of grey squares. A yellow smiley face icon is visible in the top right corner.
- Console (Bottom Right):** A console window titled "SimpleReco [Java Application]" showing the output of a Java application. The output includes timestamps, log levels, and messages such as "INFO inpro.apps.SimpleReco - RESULT: <s> Hier bin ich" and "State changed: CONNECTED".

Einschränkungen zustandsbasiertes Dialogmanagement

- System-Initiative: bzw. alles muss explizit vorgesehen sein
- aufwendige Fall-Unterscheidungen für (vorgesehene) Flexibilität

Die Frame-Eisdiele

{
sorte: {"Van, Scho, Zit"}
kugeln: int
streusel: boolean
preis: float
...
}

Die Frame-Eisdiele

| | | | |
|---|---------------------------|----------|---|
| { | sorte: {"Van, Scho, Zit"} | ← input | } |
| | kugeln: int | ← input | |
| | streusel: boolean | ← input | |
| | preis: float | ← output | |
| | ... | ... | |

Die Frame-Eisdiele

| | | | |
|---|---------------------------|----------|---|
| { | sorte: {"Van, Scho, Zit"} | ← input | } |
| | kugeln: int | ← input | |
| | streusel: boolean | ← input | |
| | preis: float | ← output | |
| | ... | ... | |

Die Frame-Eisdiele

| | | |
|---------------------------|----------|--------------------|
| sorte: {"Van, Scho, Zit"} | ← input | ← wie nach fragen? |
| kugeln: int | ← input | ← wie nach fragen? |
| streusel: boolean | ← input | ← wie nach fragen? |
| preis: float | ← output | ← wie ausdrücken? |
| ... | ... | ... |

Die Frame-Eisdiele

| | | | | |
|---|---------------------------|----------|--------------------|---|
| { | sorte: {"Van, Scho, Zit"} | ← input | ← wie nach fragen? | } |
| | kugeln: int | ← input | ← wie nach fragen? | |
| | streusel: boolean | ← input | ← wie nach fragen? | |
| | preis: float | ← output | ← wie ausdrücken? | |
| | ... | ... | ... | |

Frame-basierte DM

- erforderlicher Information wird als (unterspezifizierter) Frame dargestellt
 - Systemäußerungen fragen einzelne Framewerte ab
 - Nutzeräußerungen werden mit Frame unifiziert (also neue Information dazugebracht)
 - **Eintragung von Werte unabhängig davon, ob sie zur Frage gehören**
 - Systemäußerungen können Neueintragungen bestätigen

• S: “Welche Sorte Eis möchtest Du?”

• N: “3 Kugeln Vanille.”

• S: ?

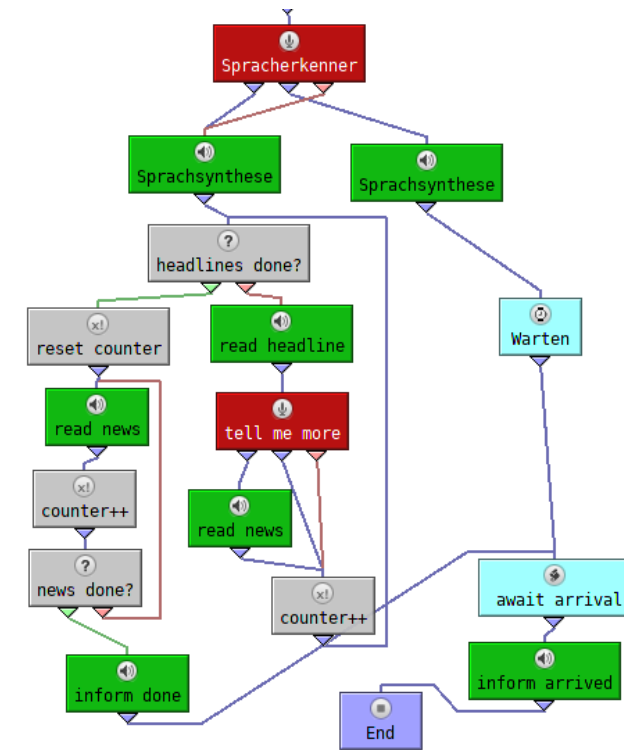
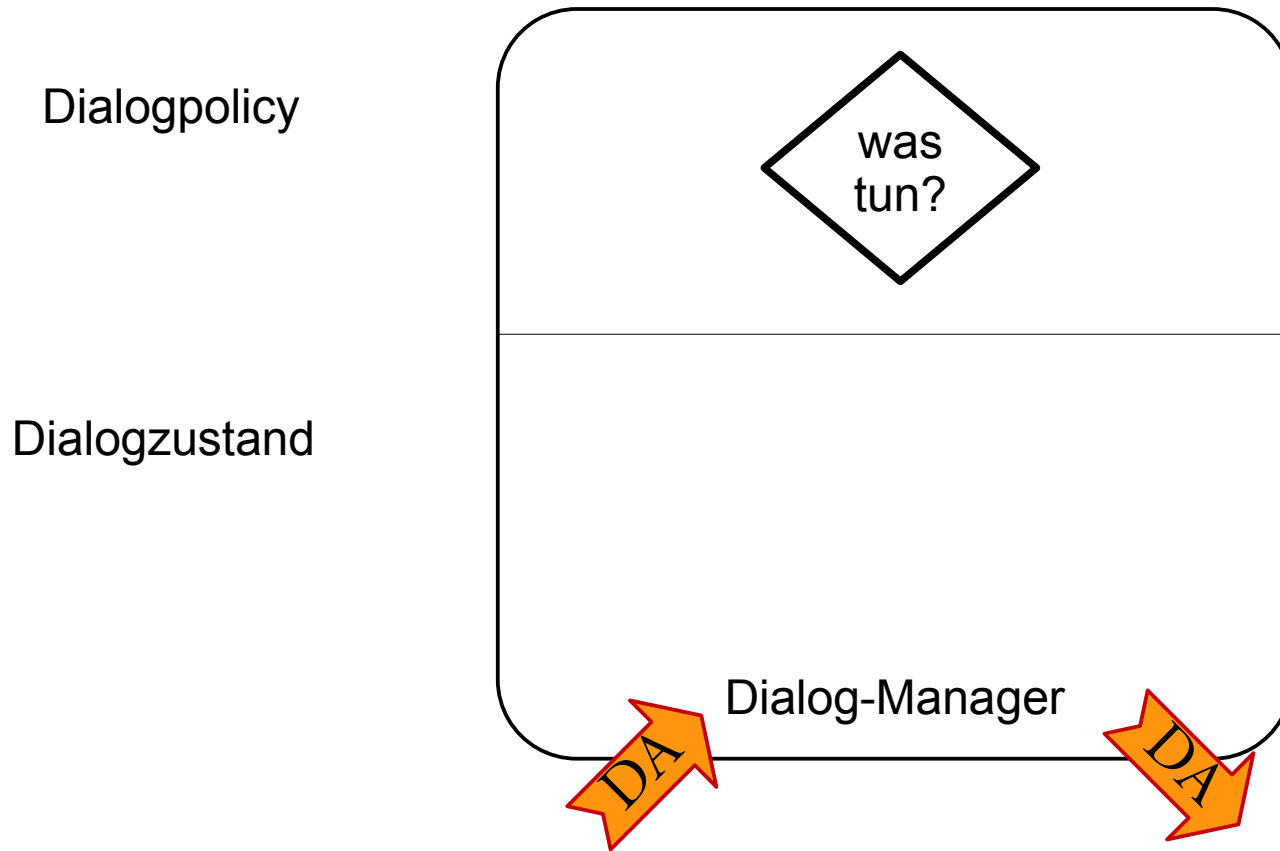
{
sorte: {"Van, Scho, Zit"}
kugeln: int
streusel: boolean
preis: float
...
}

Frame-basierte DM

- unterstützt *Overanswering*, erlaubt offene Fragen
- unterstützt variable Reihung der Abfragen
- typisch: Monitoring des Dialogverlaufs und graduelle Einschränkung des Dialogs falls nötig (frei → beschränkter)
- möglich: Individualisierung der Systemprompts

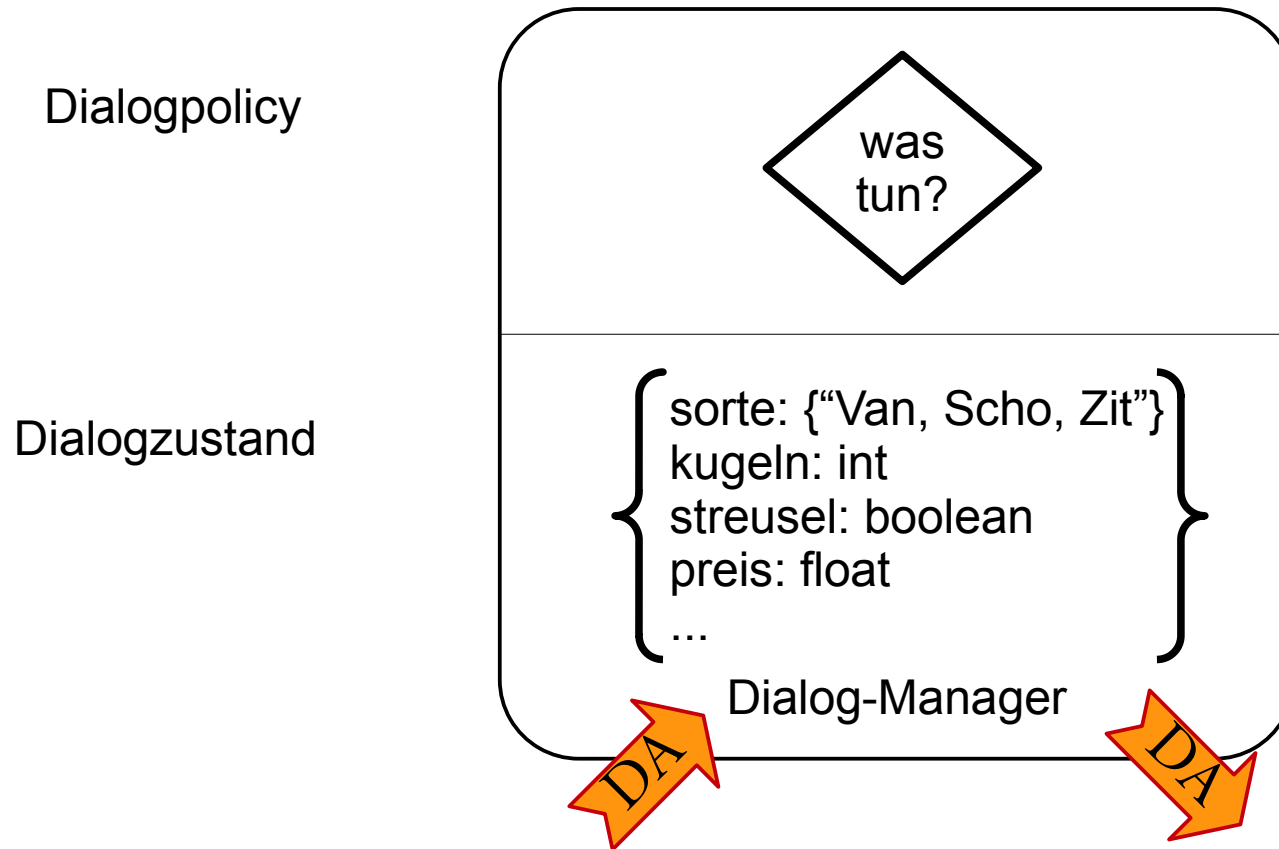
| | | |
|---------------------------|----------|--------------------|
| sorte: {"Van, Scho, Zit"} | ← input | ← wie nach fragen? |
| kugeln: int | ← input | ← wie nach fragen? |
| streusel: boolean | ← input | ← wie nach fragen? |
| preis: float | ← output | ← wie ausdrücken? |
| ... | ... | ... |

Komponenten im Dialogmanager



endlicher Automat

Komponenten im Dialogmanager: Frame-basiertes Management



Frame-Semantik

Dialog-Zustand:

- bestimmt aktuell relevanten Frame (oder mögliche Frames)
- bestimmt Ausgabe-Frame basierend auf Eingabe-Frame
- zum Beispiel: [kaufen : [Verkäufer: ich ; Käufer: Anrufer ; Ware : ? ; Preis : ?]]

Eingabeseite:

- Äußerung → Frame-Inhalt
“Hallo, ich möchte
3 Kugeln mit Sahne.”

Ausgabeseite:

- Frame-Inhalt → Äußerung
“Okay.
Das kostet 4 Euro 40.”

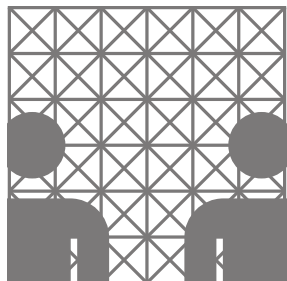
Vielen Dank.

baumann@informatik.uni-hamburg.de



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

Universität Hamburg, Department of Informatics
Language Technology Group



Notizen

Desired Learning Outcomes

- Die Studierenden können Frames entwerfen, welche die erforderliche Information in der Dialogsituation abbilden
- Die Studierenden verstehen, wie ein Dialogmodell auf Basis von Frames (und Frame-Semantik) eine Mischung aus System- und Nutzerinitiative ermöglicht.