

Vorlesung

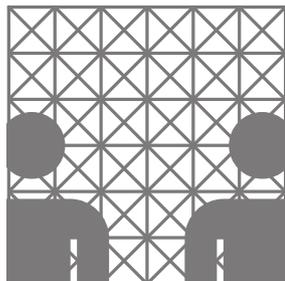
Sprachdialogsysteme

Timo Baumann
baumann@informatik.uni-hamburg.de



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

Universität Hamburg, Department of Informatics
Language Technology Group



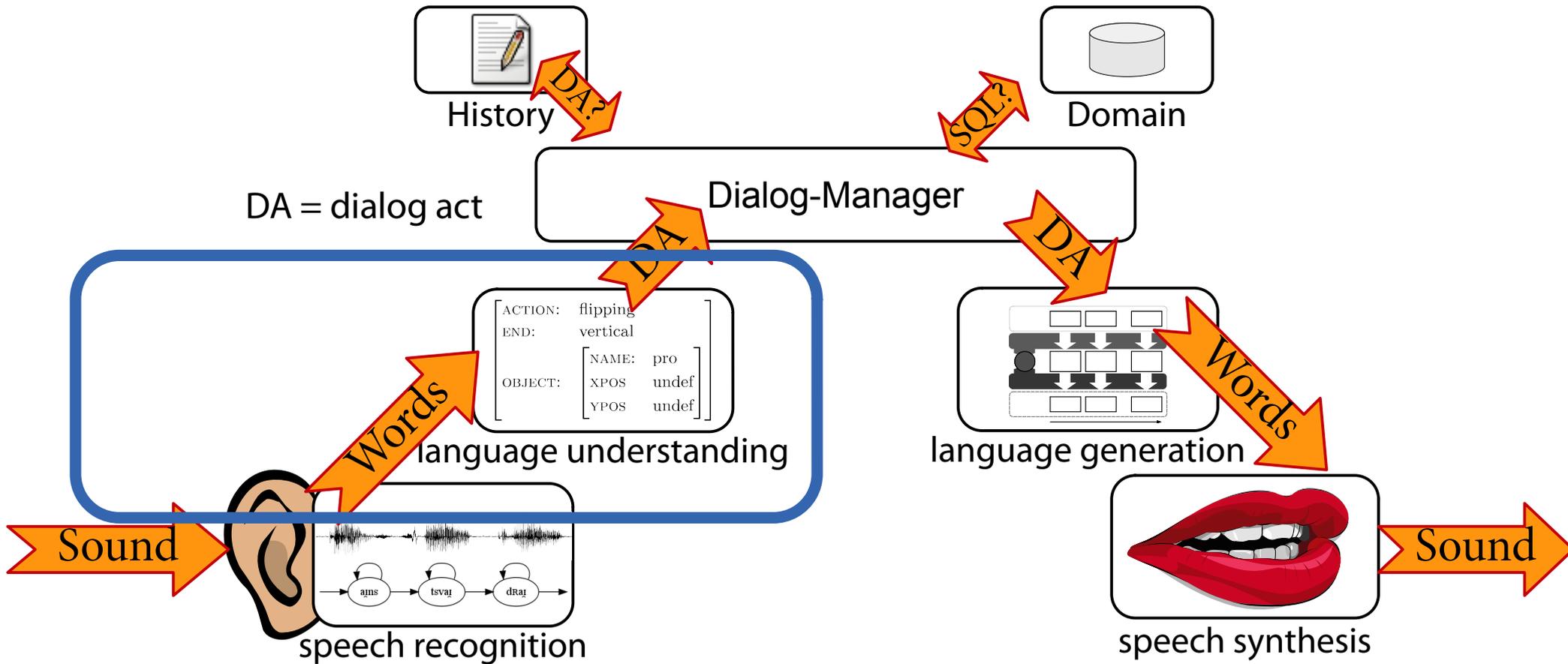
Heute

Leftovers und Reprise regelbasiertes Verstehen

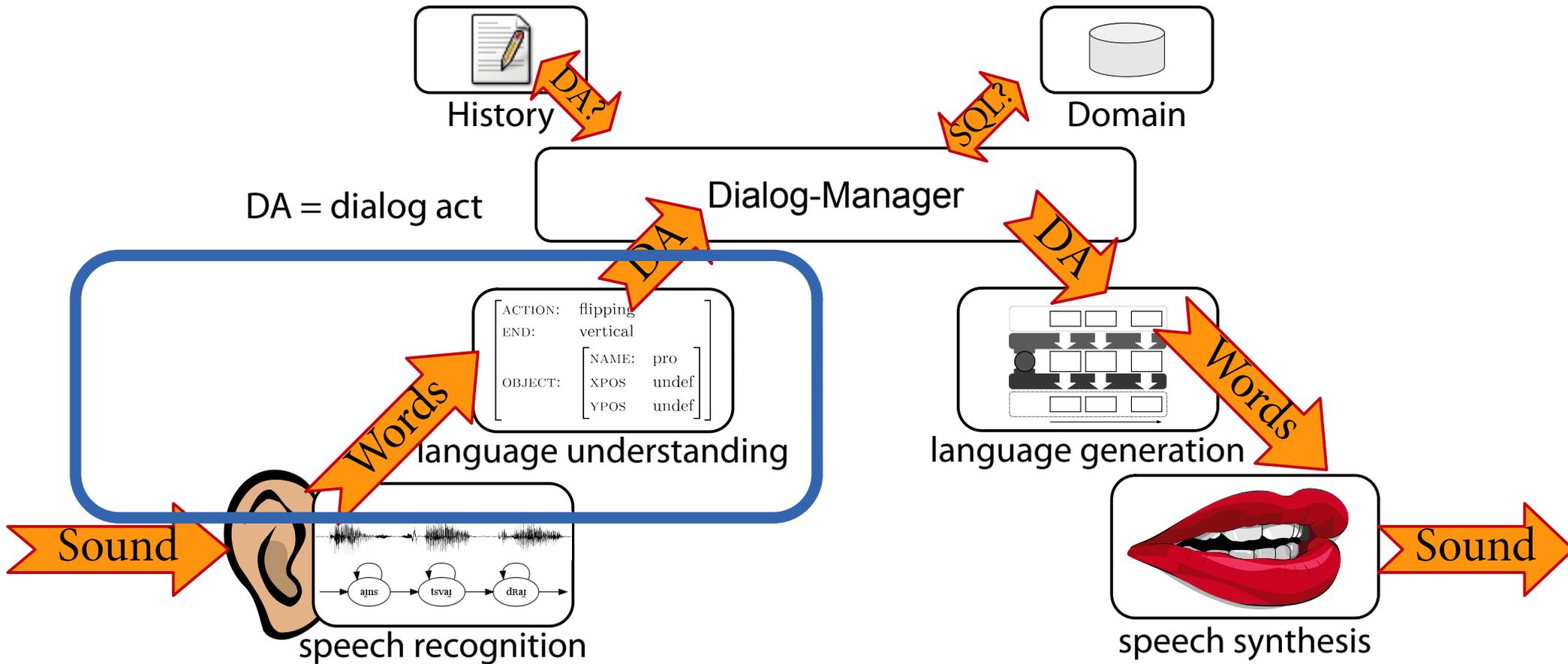
graphbasierte Dialogmodellierung

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:

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" ;
```

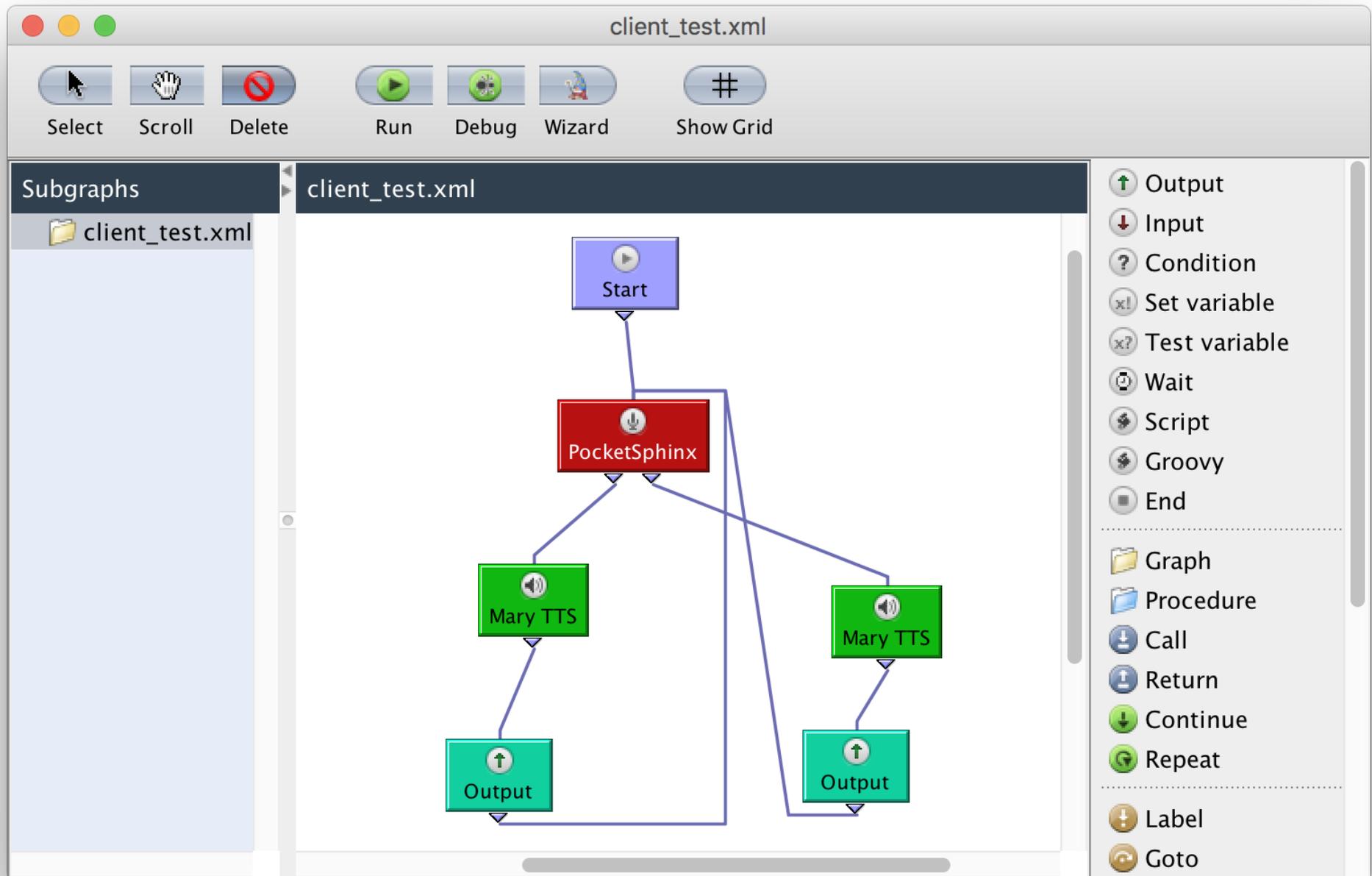
Quintessenz

- 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

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 structure involving "zähler=0", "Ausgabe", "Eingabe", "Bedingung", and "Teilbeschreibung". The flowchart includes various nodes for "None", "Color", "Form", "Farbe", "Zelle", "Spalte", "Menge", "Variable testen", "Spracherkennung", and "Script".
- PentoWorld (Top Right):** A window titled "PentoWorld" showing a game interface. It features a grid of colored blocks (blue, green, yellow, red) and a larger grid of grey blocks. A yellow smiley face icon is visible in the top right corner.
- Console (Bottom Right):** A console window titled "<Java EE> - Eclipse" showing the output of a Java application. The output includes timestamps, log levels, and messages such as "SimpleReco [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (15.08.2012 16:33:41)", "Hier bin ich", and "State changed: CONNECTED".

Graph

- Knoten
 - Sprachsyntheseknoten, die dazu führen, dass das System spricht
 - Spracherkennungsknoten, die dazu führen, dass das System eine Nutzeräußerung erkennen kann
- Kanten verbinden Knoten
- Dialogfluss bestimmt sich durch Zustandsübergänge
- Auswahl realisiert durch:
 - “Kanteninschriften”
 - mehrere “Ausgänge” der Knoten

Angemessenheit

- endliche Automaten *alleine* haben gravierende Beschränkungen:
- Frage nach Namen, erzähle Witz, Verabschiede (mit Namen)

Konversationale Skripte

- auch Menschen nutzen teils sehr rigide und inflexible Dialogführungsstrategien
- sog. konversationale Skripte
 - Begrüßungen, Verabschiedungen

Inflexibilität konversationaler Skripte (selbst erlebt)



Robustheit konversationaler Skripte



Weihnachtsaufgabe

- installieren Sie DialogOS: www.dialogos.app
- erstellen Sie ein Dialogmodell für eine selbstgewählte Anwendungsdomäne:
 -

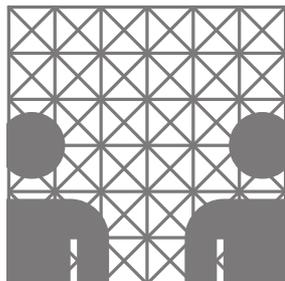
Vielen Dank.

baumann@informatik.uni-hamburg.de



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

Universität Hamburg, Department of Informatics
Language Technology Group



Notizen