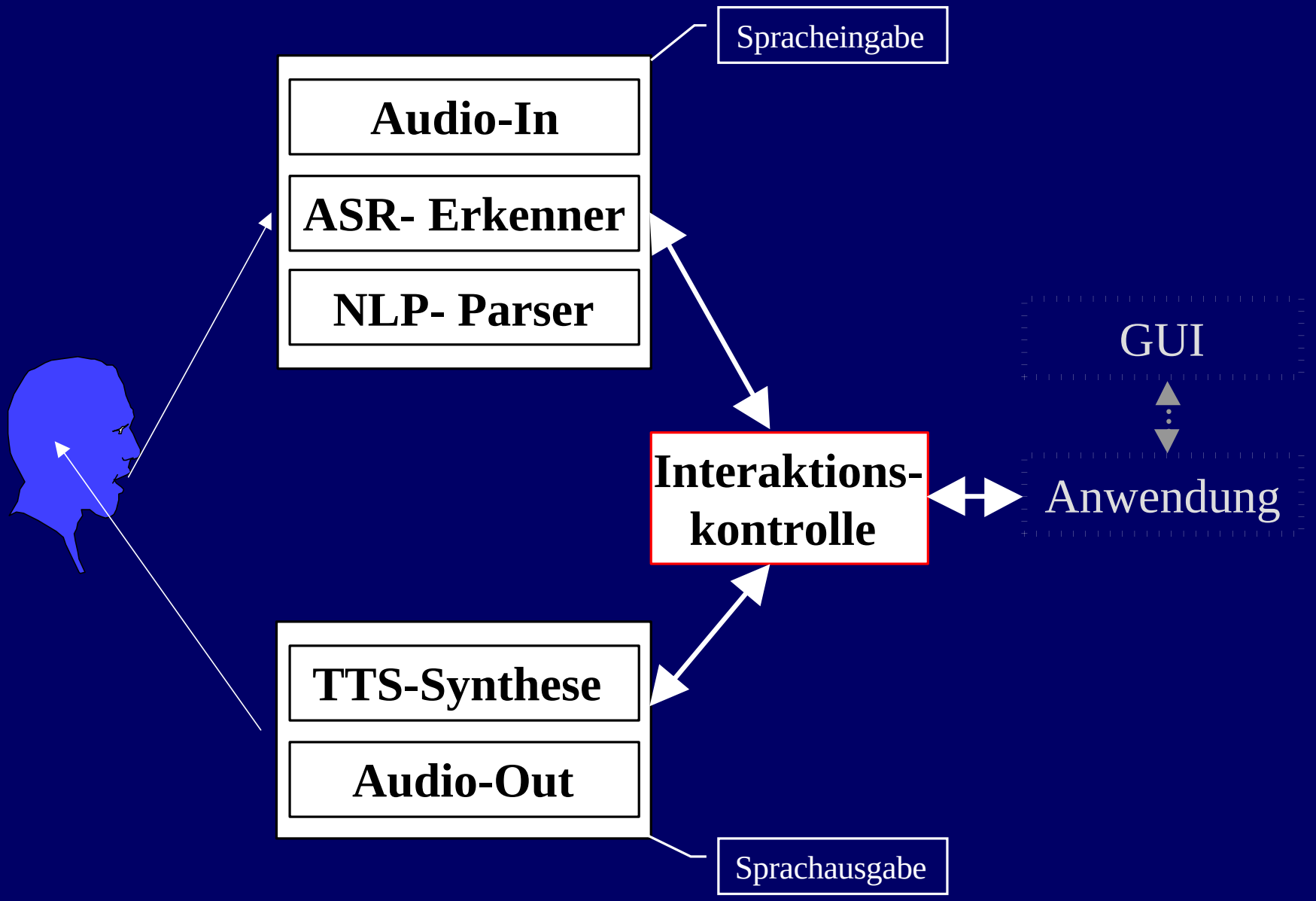
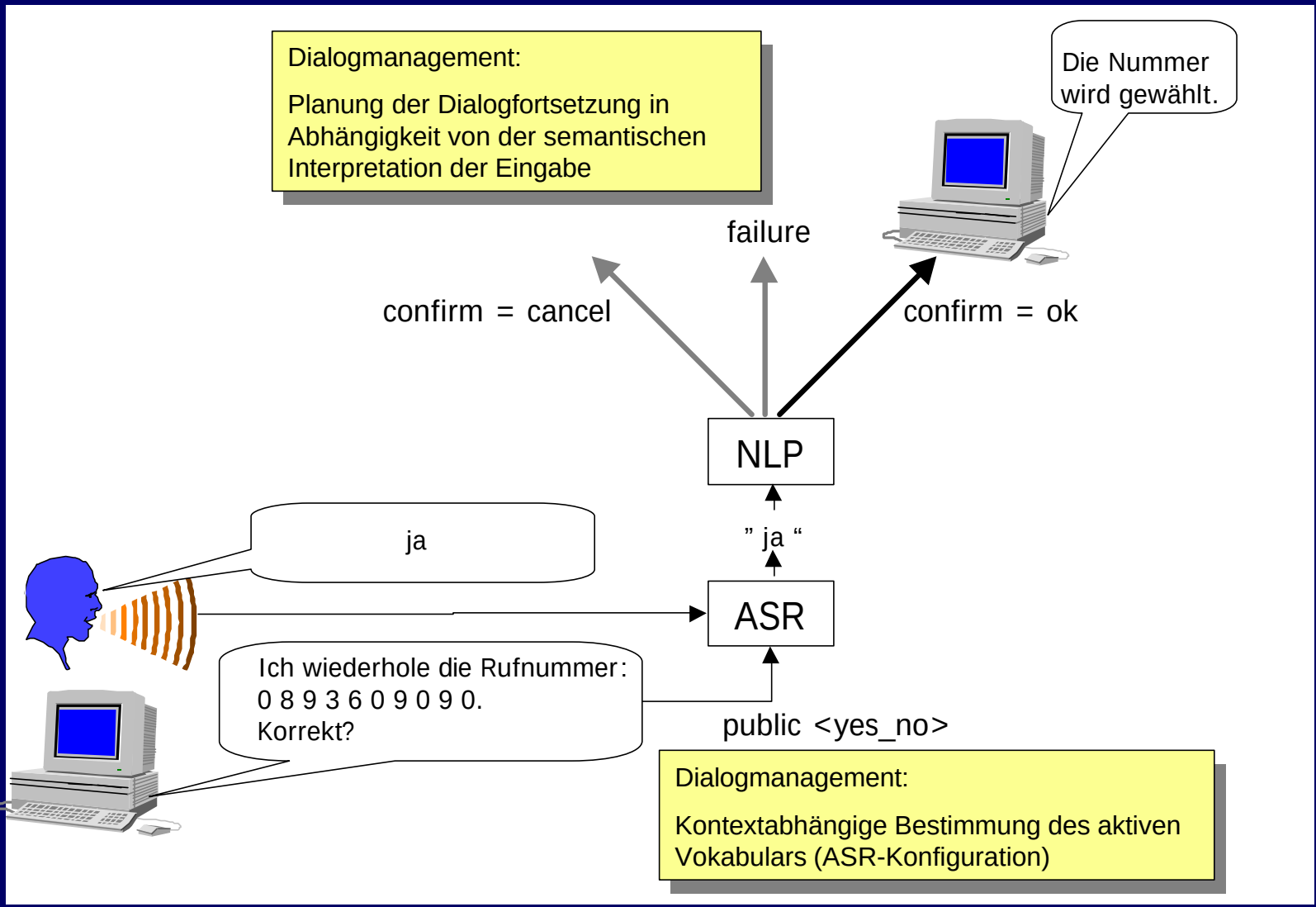
# Komponenten und allg. Architektur von SUIs



# ASR und NLP-Komponenten im Dialogverlauf



# Anforderungen und Voraussetzungen

- Vokabular
  - Größe:
    Umfang nach Leistung des Erkenners bis zu 4000 Wörter (i.d.R. aber nur ca. 1000)
  - Auswahl der Wörter: Festlegung durch vorausgehende Untersuchungen (zwischenmenschl. Dialoge) Tests und Prototypen
  - Syntax: Wortkombinationen, komplexe Phrasen, Grammatik
  - Semantik: Synonymie

# Anforderungen und Voraussetzungen

- Prompts (Systemausgaben)
  - Aufgaben:
    - Benutzerinteraktion aufrecht erhalten
    - Vorgabecharakter (Beeinflussung des Benutzerverhaltens)
  - Formulierungen durch vorausgehende Untersuchungen (zwischenmenschl. Dialoge) Tests und Prototypen

# Anforderungen und Voraussetzungen

- Interaktion (Dialogflow)
  - Initiative
    - Systemgesteuert
    - Benutzergesteuert
    - Mixed Initiative
  - Echo und Bestätigung
    - Notwendig und unverzichtbar wegen der Möglichkeit des Missverstehens von System(Fehlerkennung auf der Wortebene; falsche Interpretation durch nicht vollständige Syntax/Semantik)) und Benutzer (Missverstehen)
    - Hemmt den freien Dialogfluss
    - Macht den Dialog maschinell

# Anforderungen und Voraussetzungen

- Interaktion (Dialogflow)
  - Korrekturen
    - „back“- Möglichkeiten
  - Holistische Sichtweise des Systems
    - Veränderungen an einer Komponente verändern das Gesamtsystem und machen ein Redesign erforderlich

# Anforderungen an das Dialogdesign (Zusammenfassung)

Kontextabhängige Aufteilung des Sprachmodells
(ASR Grammatik)

Semantische Analyse natürlichsprachlicher Eingaben
(NLP Grammatik)

Promptdesign

Metakommunikation:
Bestätigungen, Korrekturen, Wiederholungen, Klärungsdialoge

Gemischt-initiative Dialogführung

# Dialogmodellierung

## Inkrementelle (Progressive) Prompts

| | |
|---|---|
| S1 | Willkommen beim Nachrichtencenter. |
| S2 | Ihre PIN bitte? |
| U1 | Was fürn Ding? |
| S3 | Wie bitte? Bitte sprechen Sie Ihre PIN in einzelnen Ziffern. |
| U2 | Ich wollt mich ja eigentlich nur erkundigen... |
| S4 | Wie bitte? Dieses spracherkennende System ermöglicht Ihnen Zugang zu Ihren Nachrichten. Dazu müssen Sie sich jetzt durch Eingabe Ihrer PIN anmelden. Bitte sprechen Sie Ihre PIN in einzelnen Ziffern, sagen Sie zum Beispiel 1 2 3. |
| U3 | Ach hmmm |
| S5 | Sie wurden wiederholt nicht richtig verstanden. Möglicherweise ist die Übertragung gestört. Bitte versuchen Sie es später noch einmal. |

# Dialogmodellierung

Implizite Bestätigung (S2/3/4/5), Fehlerkennung (nach U2), Vorgabecharakter (*zwo* statt *zwei*), indirekte Korrektur (U3)

| | |
|---|---|
| S1 | Ihre PIN bitte? |
| U1 | drei acht null |
| S2 | drei acht null und weiter? |
| U2 | fünf zwei vier |
| S3 | fünf drei vier und weiter? |
| U3 | Falsch |
| S4 | drei acht null und weiter? |
| U4 | fünf zwei vier |
| S5 | fünf zwo vier und weiter? |
| U5 | Ende |
| S6 | Willkommen im Hauptmenü Nachrichtencenter. Sie haben keine neuen Nachrichten. |

# Dialogmodellierung

## Mixed-Initiative - Direkte Korrektur (U3)

| | |
|---|---|
| S1 | Ihre PIN bitte? |
| U1 | drei acht null |
| S2 | drei acht null und weiter? |
| U2 | fünf zwei vier |
| S3 | fünf drei vier und weiter? |
| U3 | falsch fünf zwo vier |
| S4 | fünf zwo vier und weiter? |
| U4 | Ende |
| S5 | Willkommen im Hauptmenü Nachrichtencenter. Sie haben keine neuen Nachrichten. |

# Dialogmodellierung

Mixed-Initiative (U2),
Implizite Bestätigung m. Abbruchmöglichkeit durch Barge-In (S3)

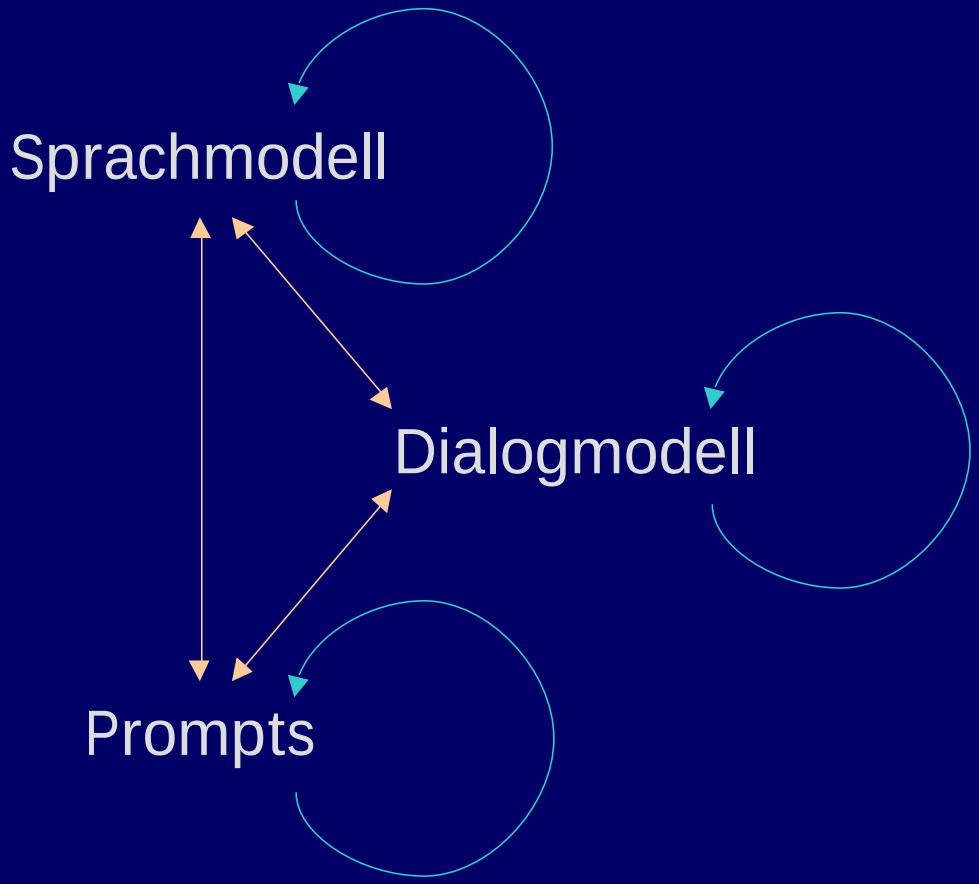
| | |
|---|---|
| S1 | Ihre PIN bitte? |
| U1 | drei acht null |
| S2 | drei acht null und weiter? |
| U2 | fünf zwo vier Ende |
| S3 | fünf zwo vier Eingabe wird überprüft. |
| S4 | Willkommen im Hauptmenü Nachrichtencenter. Sie haben keine neuen Nachrichten. |

# Dialogmodellierung

Mixed-Initiative (U1),
Mehrstufiges Recovery nach Fehlerkennung (U3/4)

| | |
|---|---|
| S1 | Hauptmenü Nachrichtencenter. |
| U1 | Neue e-mail an Hans |
| S2 | e-mail an Franz wird er [ stellt |
| U2 | ] Abbruch |
| S3 | Hauptmenü Nach [ richtencenter. |
| U3 | ] e-mail erstellen |
| S4 | Sagen Sie den Namen des Empfängers |
| U4 | Hans |
| S5 | e-mail an Hans wird erstellt. |

# Designprinzipien




Inhärente und übergreifende Designprinzipien

# Evaluation

Maße (vgl. EAGLES und ACCeSS):

-Anzahl Turns
-Ø Turn Dauer
-Dialogdauer
-Wort-/Satz-/Konzept-Akkuratheit
-Korrekturrate (Benutzer/System)
-Fehlerrate (Benutzer/System)
*-Erfolgsrate bezgl. Teilaufgaben (Subtask-Analyse)*
-Effizienz
(erfolgreich absolvierte Teilaufgaben pro Zeiteinheit)

# Dialogmanagement-Software

- generische, “programmierbare” Lösung für o.g. Anforderungen

- Anbindung an übrige Komponenten

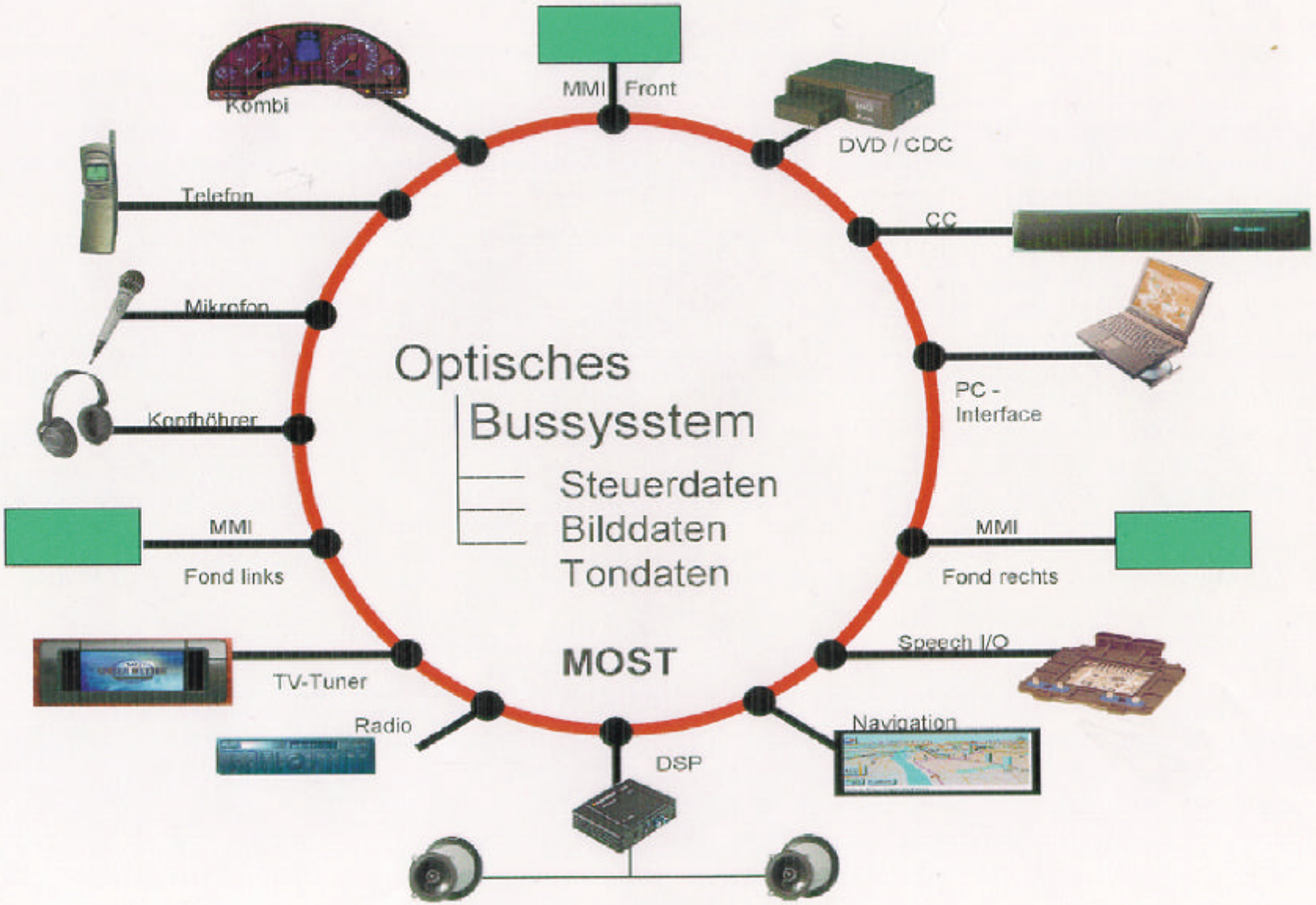
Lösung:
Dialogmodellierungssprache (DML) + Interpreter

Beispiele:
HDDL, VoiceXML, GDML, SDML

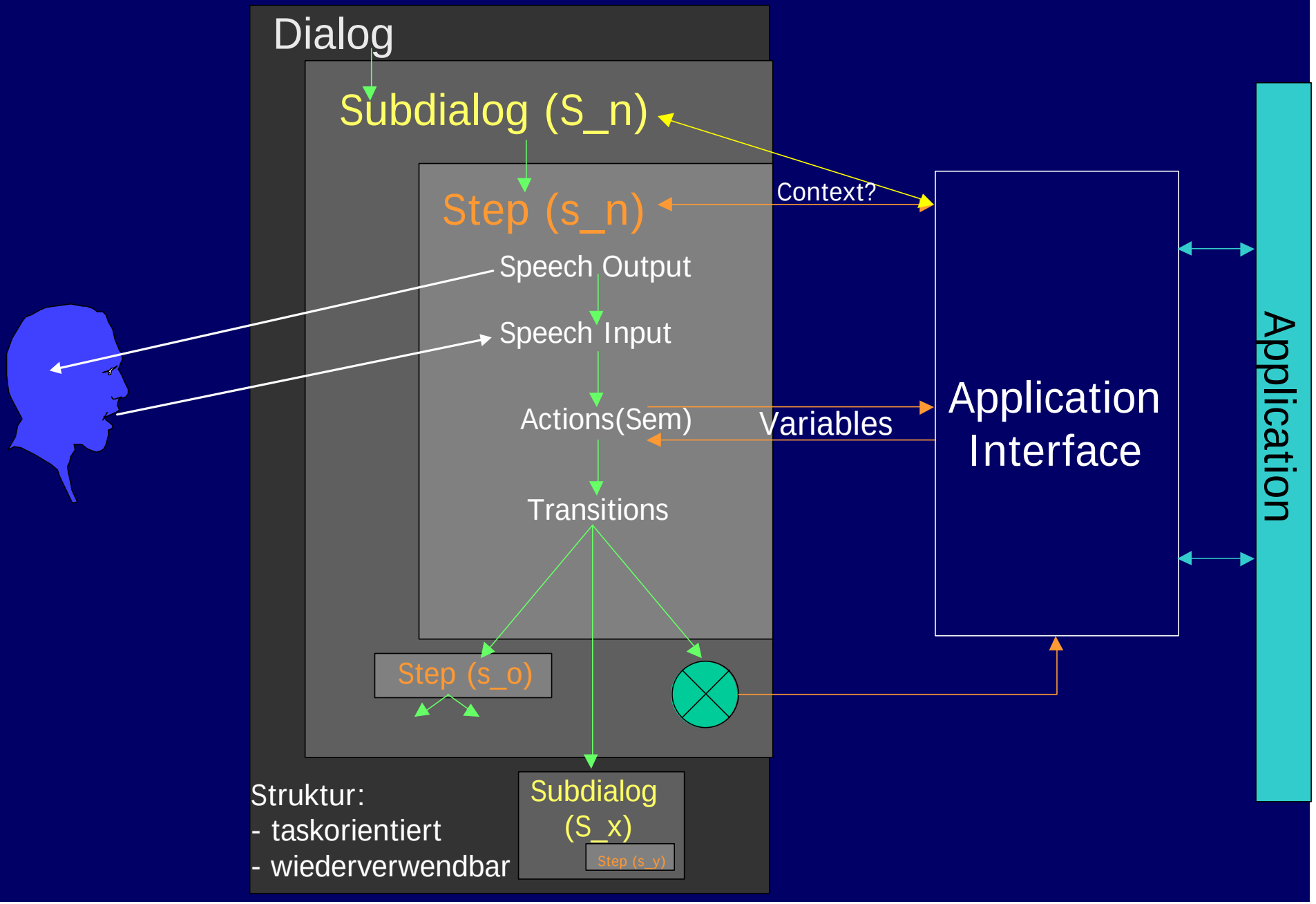
Kommerzielle Systeme/Anbieter:
Philips (SpeechMania), Temic, Sympalog, Crealog, Clarity

Problem:
Prozedurale Anteile in der Dialogdefinition

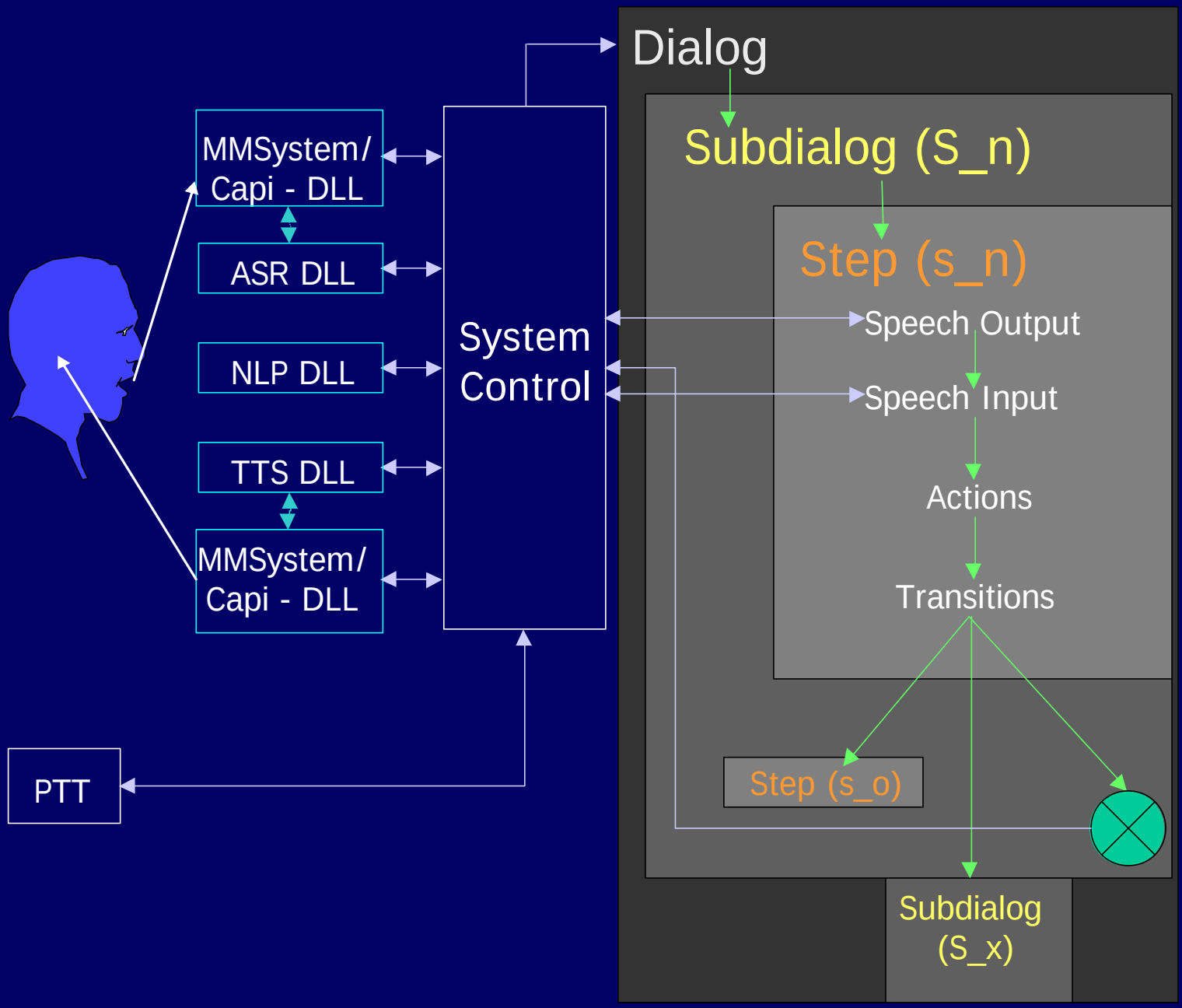
# Verknüpfungen über Bussystem



# Dialogmanagement und Anwendung



# Dialogmanagement und andere Komponenten



# Dialogmodell
# XML-basierte Dialogmodellierungssprache (DML)

```
<Subdialog Name="S_VoiceDialing">
    <Step Name="s_vd_num_main_1">
        <SpeechOutput>
            <PlayList Prompts="p_vd_req_no_1"/>
            <PlayList Prompts="p_sorry p_main_opt p_vd_req_no_2"/>
            <PlayList Prompts="p_sorry  p_hlp_vd_no_0 p....."/>
        </SpeechOutput>
        <SpeechInput Subgrammars="digit_entry"/>
        <ActionSequence Condition="digits = ANY">
            <Action String="store lastBlock"/>
            <Action String="request number"/>
        </ActionSequence>
        <Transition Condition="digits = ANY" Target="s_vd_num_continue"/>
        <Transition Condition="command = back_main" Target="S_VP_Main"/>
        <Transition Condition="command = help" Target="s_vd_num_help"/>
    </Step>
```

# Sprachmodell - Java Speech Grammar Format

Grammatik zur Nummerneingabe:

```
public <digit_entry> =
            (
            (<digits_0_9> {$Tag} <finished> {$Tag}) |
            <digits_0_9> {$Tag} |
            <finished> {$Tag} |
            <delete> {$Tag} |
            <correct> {$Tag}
            ) {$Tag};

<digits_0_9> = ( ( (1 | 2| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 ) + )) {digits=%tokens};

<finished> = (aus | Ende | fertig | Eingabe beendet) {command = finish};

<delete> = (löschen | [nochmal] von vorne) {command = delete};

<correct> = ((falsch | Korrektur | korrigieren | zurück) +) {command = correct};
```

„2 0 9 Ende“ --> digits = „2 0 9“; command = finish