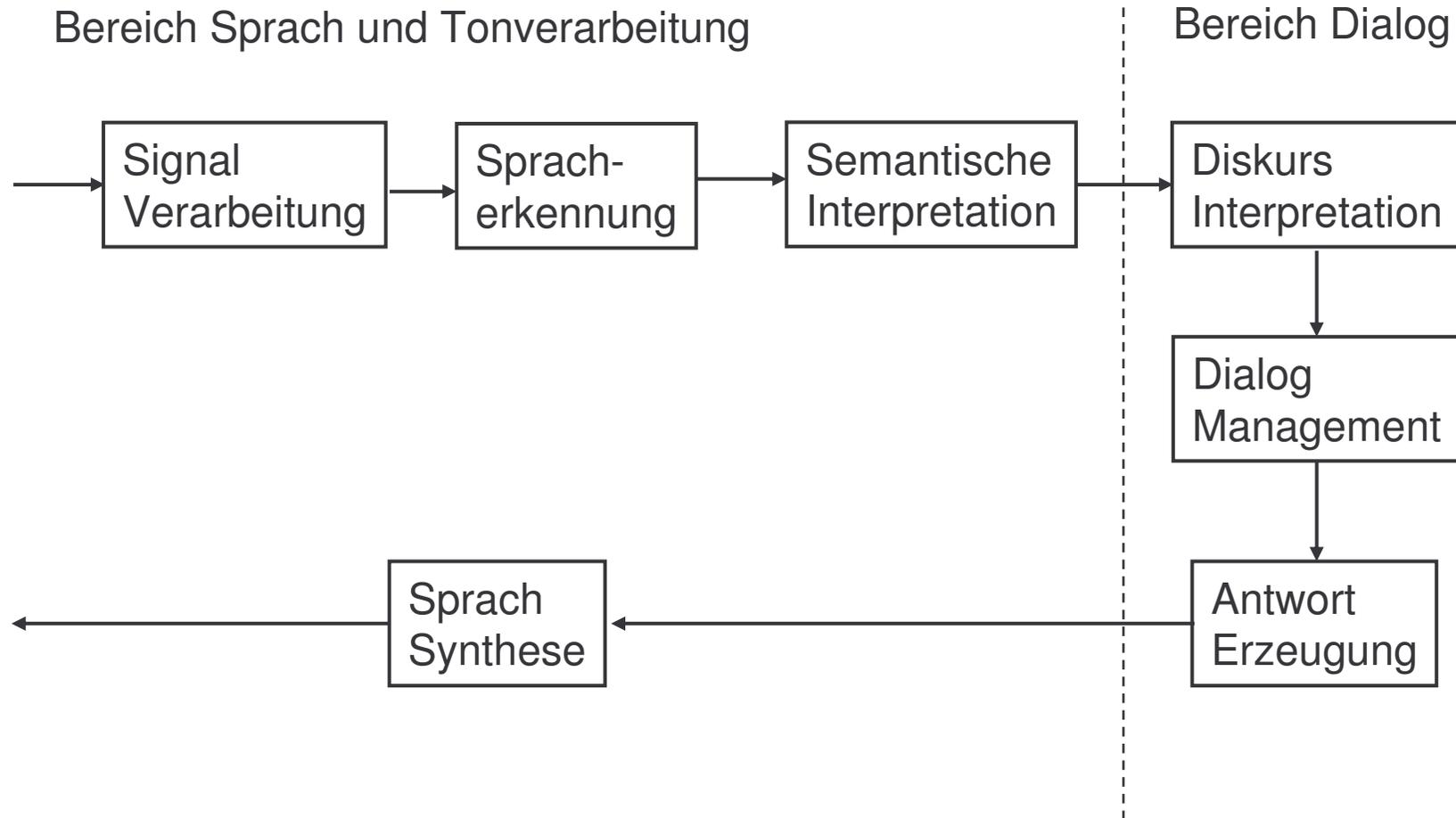


# Dialogsysteme

## Überblick

# Datenfluß



# Sprach- und Tonverarbeitung

- **Signalverarbeitung:**
  - Konvertieren der Audiowelle in einen Featurevektor
- **Spracherkennung:**
  - Die Sequenz der Featurevektoren in eine Sequenz von Wörtern dekodieren
- **Semantische Interpretation:**
  - Die Bedeutung des erkannten Wortes bestimmen
- **Sprachsynthese:**
  - Eine künstliche Sprachausgabe aus einer markierten Wortfolge erzeugen

# Semantische Interpretation: Wortketten

## Beispiele:

- Usereingabe beinhaltet nur relevante Informationen
  - *System:* What is your address?
  - *User:* fourteen eleven main street
- Sonst Schlüsselwortsuche und Konzeptextraktion nötig
  - *User:* My address is **fourteen eleven main street**
- Allgemeine Anwendungen der Dialogsysteme
  - Ausfüllen von Formularen oder Templates
  - Verzeichnisdienste
  - Informationssuche

# Semantische Interpretation: Musterbasiert

- Inhalt wird durch einfache (meist reguläre) Modelle repräsentiert
- Beispiel Reisebuchung:
  - *System*: What are your travel plans?
  - *User*: [On Monday], I'm going [from Boston] [to San Francisco].
  - Inhalt: [DATE=Monday, ORIGIN=Boston, DESTINATION=SFO]
- Kombiniert Inhaltsextraktion mit Sprachmodellierung
  - als Sprachmodell alleine zu restriktiv
- Java Speech API:

```
public <command> = <action> [<object>] [<polite>];  
    <action> = open {OP} | close {CL} | move {MV};  
    <object> = [<this_that_etc>] window | door;  
    <this_that_etc> = a | the | this | that | the current;  
    <polite> = please | kindly;
```

- Generierung und Update zur Laufzeit (z. B. Web Apps)

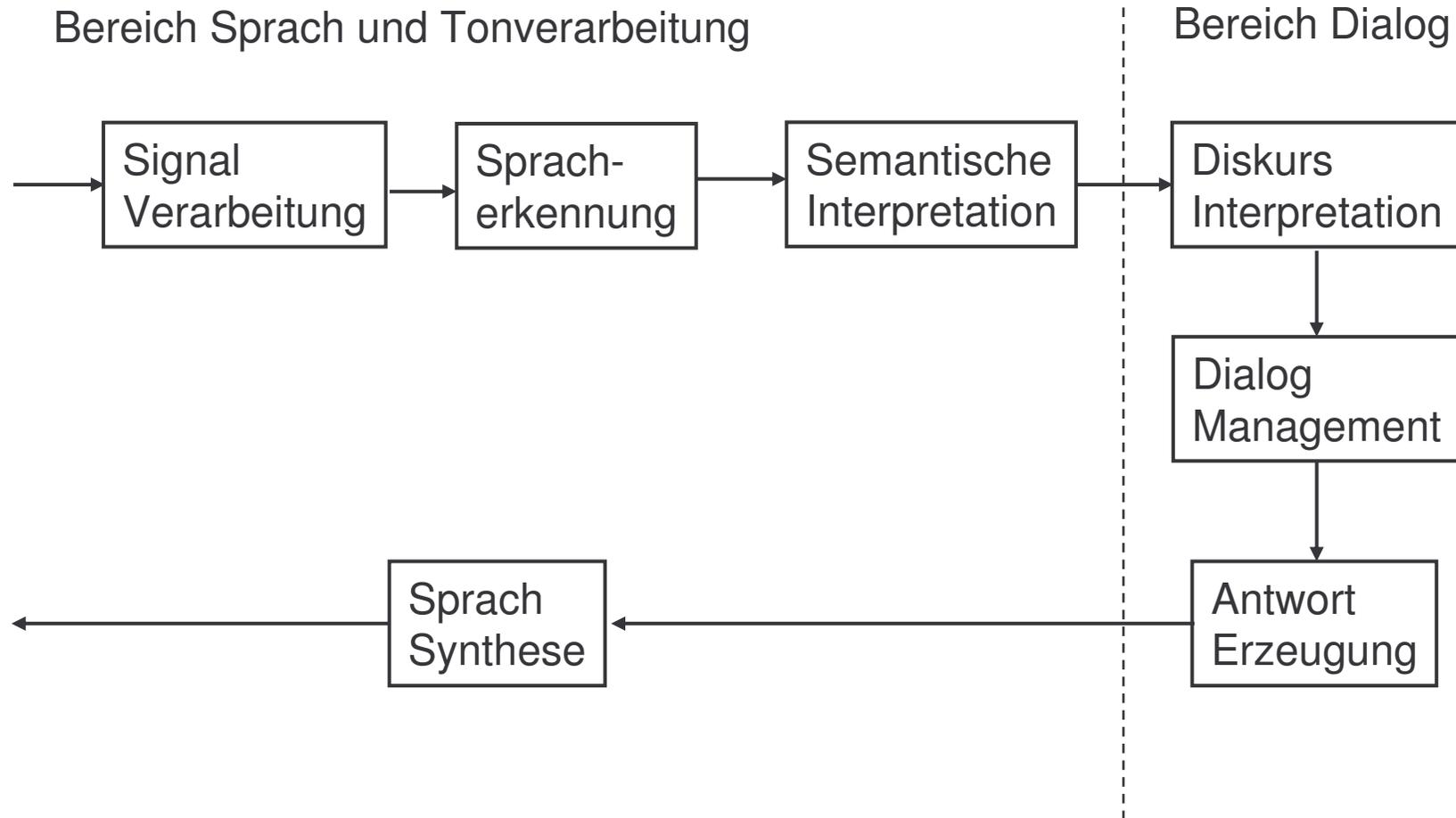
# Semantische Interpretation: Parsing

- Generelle Problemstellung: Wer macht was mit welchem Objekt?:
  - System: What would you like me to do next?
  - User: Put the block in the box on Platform 1. [ambiguous]
  - System: How can I help you?
  - User: Where is A Bug's Life playing in Summit?
- Parsing stellt Beziehungen zwischen den beteiligten Objekten her:
  - Wer macht was mit welchem Objekt?  
User: Where is A Bug's Life playing in Summit?  
S: (where(present(in(Summit,play(BugsLife))))))
  - Methoden aus der maschinellen Übersetzung
- Eleganter ist es die Behandlung kontextbezogen durchzuführen:
  - Aussagentyp: Wo - Frage(Äußerung der Absicht)
  - Film: A Bug's Life
  - Stadt: Summit

# Semantische Interpretation: Probleme

- **Kontrollierte Sprache**
  - limitierter Wortschatz und Grammatik
- **Spontane Sprache**
  - hoher Anteil von Worten die nicht dem Wortschatz entsprechen
  - Beinhaltet: nicht-komplette Wortteile, Auslassungen, Neuanfänge, Fehlerkennungen ...
  - Hohe Varianz in der Grammatik
  - Erzeugt hohe Fehlerquoten im Spracherkenner
- **(oft partielles) Parsing ermöglicht:**
  - Erkennung von Auslassungen
  - Einordnung von Fragmenten

# Datenfluß



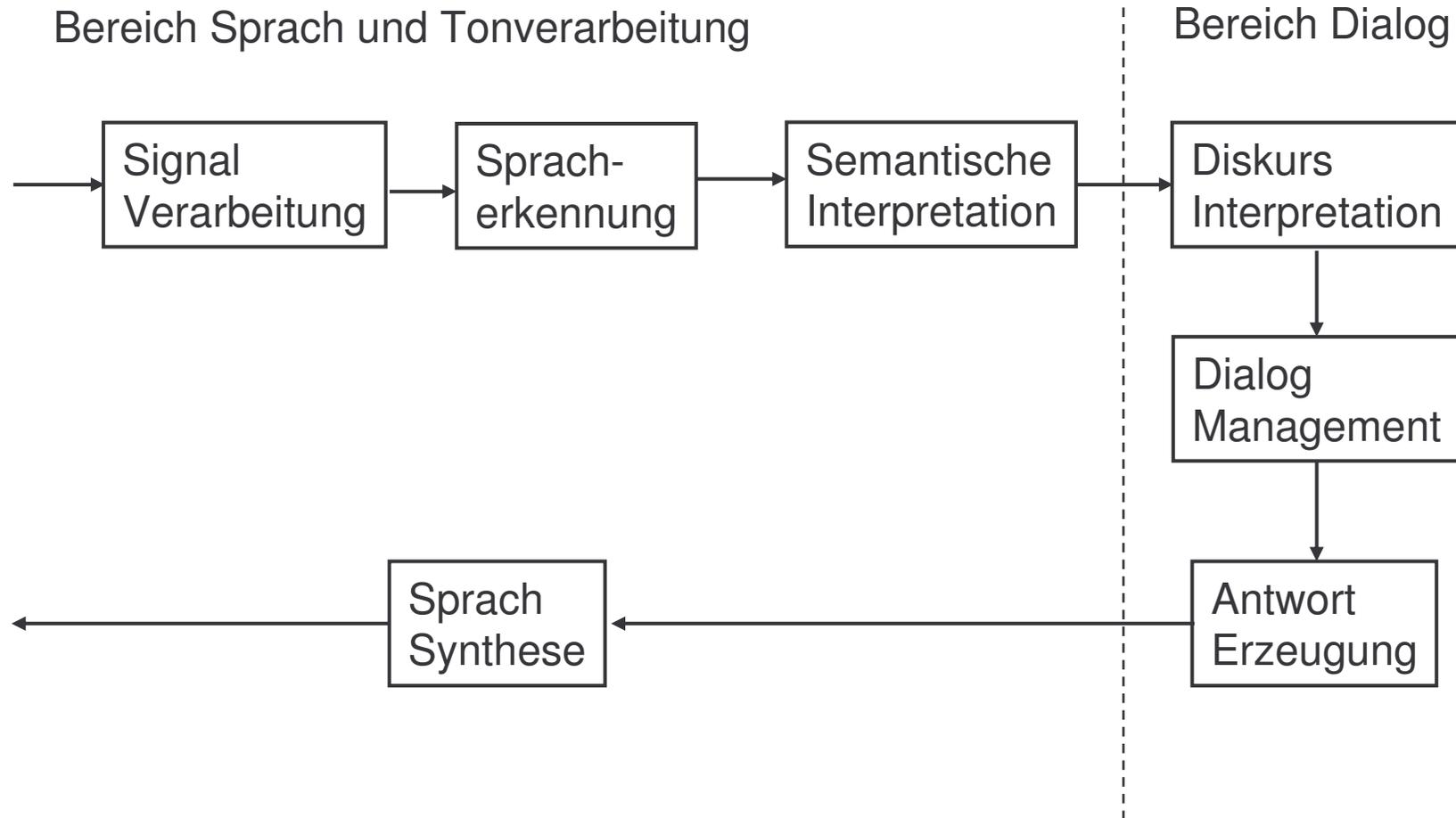
# Sprachsynthese

- Einfachste und verbreitetste Methode:
  - Eingabeaufforderungen von professionellen Sprechern aufnehmen und wiedergeben
- Hohe Qualität
- Limitierte Anzahl von Aufnahmen möglich
- erweiterbar durch Templates und Stückelung von Aufnahmen (verkettende Synthese)

# Sprachsynthese

- regelbasierte Synthese
  - Linguistische Regeln und Training generieren Eigenschaften der Antwort
- verkettende Synthese
  - Aufnahme eines grundlegenden Inventars von Wörtern
  - Bestimmung zutreffender Reihenfolgen zur Laufzeit
  - Verkettung mit Anpassung von Wortlängen und Tonhöhen
  - Generierung der Ausgabedatei

# Datenfluß



## Vom Diskurs zum Dialog

- Diskursinterpretation:
  - Verstehen der Aussage im Kontext
- Dialogmanagement:
  - Festlegen der Systemziele für die Antworten hinsichtlich der Usereingaben
- Antworterzeugung:
  - Natürlichsprachliche Antworten generieren um die gewählten Ziele zu erreichen.

# Diskursinterpretation

- Ziel: Verstehen was der User wirklich will
- Beispiel: Können Sie es bewegen?
  - Was ist mit „es“ gemeint?
  - Ist die Frage mit ja/nein zu beantworten oder eine Aufforderung?
- Relevante Gesichtspunkte:
  - Referenzpunktbestimmung
  - Intentionserkennung
- Interpretation der Useräußerung im Kontext

# Referenzpunktbestimmung

U: Wo wird **Silentium** in **München** gezeigt?

S: Silentium wird im Monopol gezeigt.

U: Wann läuft **er dort**?

S: Er wird um 17, 19 und 22 Uhr 30 gezeigt.

U: Ich möchte zwei **Erwachsene** und ein **Kind** für die **erste Vorstellung**.

Wie viel würde **das** kosten?

- Wissensarten:
  - **Domain Wissen**
  - **Diskurs Wissen**
  - **Weltwissen**

# Theoretische Referenzpunktbestimmung

- Fokusstack:
  - Verwaltung aktueller Objekte in einem Stack
  - Auswahl der Objekte die semantische/pragmatische Richtlinien erfüllen, beginnend mit top() des Stacks
  - Bei Bedarf Rücksichtnahme auf Diskursstruktur
- Fokusverwaltung:
  - Zurückverweisender Fokus (Backward-looking center (Cb)): Objekt verbindet aktuellen Satz mit dem vorhergehenden
  - Vorwärtsgerichteter Fokus (Forward-looking centers (Cf)): Vermerk des potentiellen Cb des nächsten Satzes
  - Regelbasierte Filterung & Ranking der Objekte zur Auflösung von Pronomen

# Referenzpunktbestimmung in der Praxis

- Nicht vorhanden: Rückweisende Referenzen sind nicht erlaubt
- Nur einfache Referenzen möglich:
  - Benutzung des Fokusstacks für Verweisauflösungsmechanismen
  - Kein Bezug auf die Diskursstruktur
- Beispiel:

U: Wo wird Silentium in München gezeigt

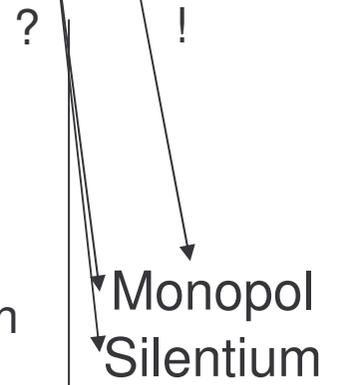
München  
Silentium

# Referenzpunktbestimmung in der Praxis (2)

S: Silentium wird im Monopol gezeigt.

Monopol  
Silentium

U: Wann läuft er dort?



Besser: Semantische Verknüpfung zwischen Kinofilm und Laufen

# Intentionserkennung

- Drei mögliche Arten von Reaktionen auf Fragen in einem Dialog:
  - Bestätigung
  - Zurückweisung
  - Informative Antwort

# Intentionserkennung

- Ziel: Erkennung der Intention einer Useräußerung innerhalb eines Dialogs bezüglich des Kontextes
- Beispiele:
  - DAMSL: Dialog Act Markup in Several Layers Sample dialogue actions
  - Verbmobil (Modellierung primär für mobile Endgeräte)
  - Switchboard DAMSL
    - Conventional-closing
    - Statement-(non-)opinion
    - Agree/Accept
    - Acknowledgment
    - Yes-No-Question/Yes-Answer
    - Non-verbal
    - Abandoned
  - Verbmobil
    - Greet/Thank/Bye
    - Suggest
    - Accept/Reject
    - Confirm
    - Clarify-Query/Answer
    - Give-Reason
    - Deliberate
- Standardisierung im Gange (Discourse Resource Initiative)

# Theoretische Intentionserkennung

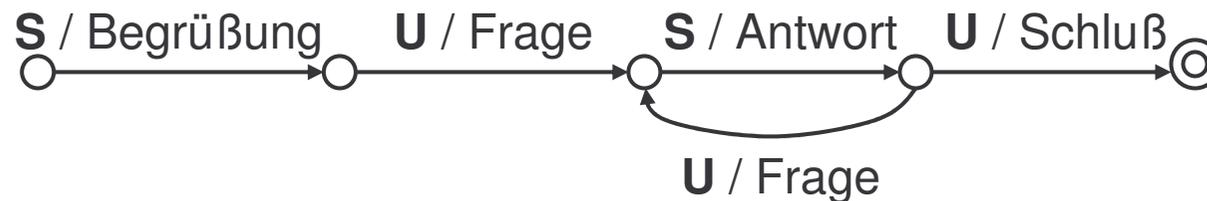
- Wissensquellen:
  - Grundsätzliche Zielsetzung der Dialoge
  - Orthographische Eigenschaften, z.B.:
    - Satzzeichen
    - Hinweisende Wörter und Phrasen: “aber”, “und”, “so” , “wie”
    - Umschreibungen: “Könnten Sie bitte”, “Ich möchte”
  - Dialogverlauf, z. B. bereits abgehandelte Dialogteile
  - Dialogstruktur, z. B.:
    - Begrenzte Unterdialoge
    - Dialog Themawechsel
  - prosodische Merkmale der Eingabe: Akzent, Tonhöhe, Druckstärke, Pausen, Sprechgeschwindigkeit

# Theoretische Intentionserkennung

Drei Grundsätzliche Methoden:

- Endliche Zustandsgrammatik für den gesamten Dialog:

- Beispiel:



- Plan-basiertes Diskursverstehen:
  - “Kochrezepte”: Templates für Aktionen
  - Ableitungen: durch Folgern und Schließen plausible Pläne erzeugen
- Empirische Methoden:
  - Probabilistisch Dialogabläufe klassifizieren: HMMs
  - Regelbasierte Dialogabläuferkennung: Lernen Transformationen

# Intentionserkennung in der Praxis

- Trifft Annahmen zu (höchst) **anforderungs-spezifischen Absichten**:  
z.B.
  - Telefonauskunft
  - Reisebuchung
  - Kinoprogramm
- Erlaubt keine vom User initiierte komplexe Dialogabläufe  
z.B. Bestätigung, Klärung von Sachverhalten oder indirekte Antworten

S1: Wie ist Ihre Kontonummer?

U1: Ist das die auf der EC-Karte?

S2: Wollen Sie 1500€ vom Sparkonto auf das Girokonto überweisen?

U2: Wenn ich genug habe.

# Intentionserkennung in der Praxis

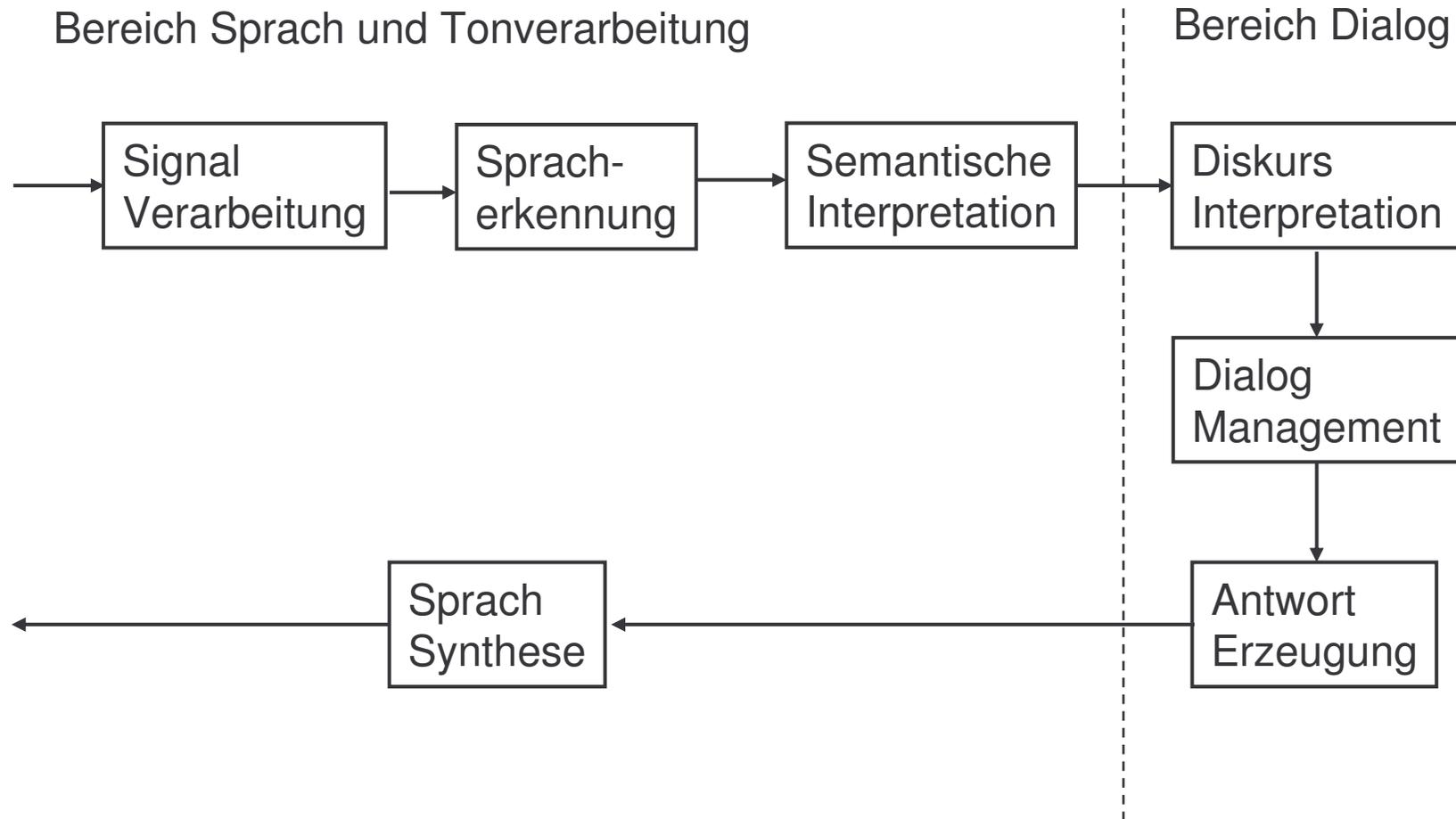
- Usereingabe kann nur eine von zwei Zielen haben:
  - Identifikation eines auszuführenden Dienstes
  - Bereitstellung von Informationen bei der Ausführung eines Dienstes
- Basierend auf Schlüsselwörtern einer Aussage oder der syntaktischen / semantischen Repräsentation
- Zuordnung von Schlüsselwörtern zu Repräsentationen und Intentionen:
  - Templatematching
  - Wahrscheinlichkeitsmodelle (aus Anwendungsstudien)
  - Vektor-basierte Ähnlichkeitsmaße

# Beispiel Intentionserkennung

U: Wann wird **Silentium** im **Monopol** gezeigt?

- Schlüsselwort und Wahrscheinlichkeitsmaximierung ergibt:
  - Intention: **Fragestellung: Zeit**
  - Film: **Silentium**
  - Kino: **Monopol**

# Datenfluß



# Dialog Management: Motivating Examples

- Dialogue 1:

S: Would you like movie showtime or theater playlist information?

U: Movie showtime.

S: What movie do you want showtime information about?

U: Saving Private Ryan.

S: At what theater do you want to see Saving Private Ryan?

U: Paramount theater.

S: Saving Private Ryan is not playing at the Paramount theater.

# Dialog Management: Motivating Examples (2)

- Dialogue 2:

S: How may I help you?

U: When is Saving Private Ryan playing?

S: For what theater?

U: The Paramount theater.

S: Saving Private Ryan is not playing at the Paramount theater, but it's playing at the Madison theater at 3:00, 5:30, 8:00, and 10:30.

## Dialog Management: Motivating Examples (3)

- Dialogue 3:

S: How may I help you?

U: When is Saving Private Ryan playing?

S: For what theater?

U: The Paramount theater.

S: Saving Private Ryan is playing at the Fairmont theater at 6:00 and 8:30.

U: I wanted to know about the Paramount theater, not the Fairmont theater.

S: Saving Private Ryan is not playing at the Paramount theater, but it's playing at the Madison theater at 3:00, 5:30, 8:00, and 10:30.

# Vergleich der Dialoge

- Dialogue 1:
  - System-initiative
  - Implicit confirmation
  - Merely informs user of failed query
  - Mechanical
  - Least efficient
- Dialogue 2:
  - Mixed-initiative
  - No confirmation
  - Suggests alternative when query fails
  - More natural
  - Most efficient
- Dialogue 3:
  - Mixed-initiative
  - No confirmation
  - Suggests alternative when query fails
  - More natural
  - Moderately efficient

# Dialog Management

- Ziel: Feststellung was auf Grund der Useräußerung bewerkstelligt werden muss:
  - Fragen beantworten
  - Weitere Informationen sammeln
  - Bestätigung / Klärung der Useraussage
  - Benachrichtigung bei ungültiger Anfrage
  - Benachrichtigung bei ungültiger Anfrage und alternativen Vorschlag unterbreiten
- Schnittstelle zwischen User / Sprachverarbeitenden Komponenten und der Knowledge Base des Systems

## Dialog Management (2)

- Hauptgesichtspunkte beim Entwurf des DM:
  - Funktionalität: Was und Wieviel soll das System machen?
  - Prozess: Wie soll das im System ablaufen?
- Unterliegt dem Einfluß von:
  - Komplexität der Anwendung:
  - Komplexität des Dialogs: Welche / Wieviele Dialogabläufe sind erlaubt?
- Beeinflusst:
  - Robustheit
  - Natürlichkeit
  - Wahrgenommene Intelligenz

# Komplexität der Anwendung

- Abhängig von der Anwendung
- Beispiele:



- Betrifft direkt:
  - Art und Menge der im System gehaltenen Informationen
  - Komplexität der Schlußfolgernden Algorithmen des Systems

# Dialogentwurf und Komplexität

- Festlegung über was geredet werden kann:
  - Ausschließlich die Aufgabe
  - Unterdialoge: z.B. zur Klärung und Bestätigung von Eingaben
  - Rückverweise: Metadialoge
    - Können Sie kurz warten?
    - Haben Sie ein Störgeräusch gehört?
- Festlegung welcher Dialogpartner über welchen Inhalt reden kann:
  - System only
  - User only
  - Beide

# Dialog Management: Theoretische Hintergründe der Funktionalität

- Festlegung von Anwendungsfällen / Ergebnissen die je Durchlauf vom System erreicht werden können
- Auf Anwendungsebene durch die Komplexität der Anwendung vorgegeben
- Auf Dialogebene durch den Systementwickler vorgegeben durch die erlaubte Komplexität der Dialoge:
  - Sind Subdialoge erlaubt?
  - Sind Meta-Dialoge erlaubt?
  - Only by the system, by the user, or by both agents?
- Weitere Faktoren zur Komplexität der Dialoge :
  - System/user-initiierte komplexe Subdialoge
    - Eingebettete Unterhandlungen
    - Ausdruck von Zweifel
  - Multiple Dialogfäden

# Dialog Management: Funktionalität in der Praxis

- Komplexität der Anwendung: bisher nur simple bis moderate Probleme implementiert
  - Anrufweiterleitung / Mailboxen
  - Wetterinformationsdienst
  - Fahrplan und Ticketing
  - Augmented Systems (Kombination mit Tasteneingabe)
- Komplexität der Dialoge:
  - Nur Anwendungsspezifisch
  - Begrenzte System-initiierte Subdialoge

# Dialog Management: Prozess

- Legt das Systemverhalten innerhalb der einzelnen Anwendungsfälle / Aufgaben fest
- Auf Dialogebene wird vom Systementwickler die Entscheidungsdirektive festgelegt:
  - System-initiative: das System hat stets die Kontrolle über den Dialogablauf, der User beantwortet lediglich Fragen
  - User-initiative: der User hat stets die Kontrolle über den Dialogablauf, das System beantwortet lediglich Fragen
  - Mixed-initiative: Kontrolle über den Ablauf wechselt zwischen System und User nach festen Regeln
  - Variable-initiative: Kontrolle über den Ablauf wechselt zwischen System und User, gesteuert nach dynamisch erzeugten Regeln und Dialoghistorie, Uservariablen...

# Dialog Management: Theoretischer Prozess

- Kontrollfluß (Entscheidungs-) Strategien:
  - Mixed-initiative
  - Variable-initiative
- Mechanismen zur Kontrollflußmodellierung:
  - Planen und Folgern
  - Theorembeweise
  - Modellierung von Annahmen
- Wissenquellen zur Kontrollflußmodellierung:
  - Informationsquellen des System, Wissen des Users, Allgemeinwissen(Weltwissen) und beiderseitiges Wissen
  - Domainwissen des Systems
  - Dialoghistorie
  - User Präferenzen (aus vorherigen Dialogen / “Userdaten”)

# Dialog Management: Prozess in der Praxis

- Kontrollfluß (Entscheidungs-) Strategien:
  - User-initiative
  - System-initiative
  - Mixed-initiative
  - Variable-initiative
- Mechanismen zur Kontrollflußmodellierung:
  - System und mixed-initiative: Endliche Automaten
  - Variable-initiative: Modell der Berechneten Kontrolldirektive
    - Wissensquellen:
      - Dialoghistorie: z.B. Kein Fortschritt bei der Bearbeitung der Aufgabe
      - Rollen der Dialogpartner: advisor/advisee vs. collaborators
      - Eigenschaften der aktuellen Spracheingabe z.B.: nicht eindeutige Aussagen, zu geringer Informationsgehalt der Aussage

# Dialog Management: Zusammenfassung

- **Theorie:**
  - Fokus liegt auf der Entwicklung von Algorithmen die schwierige Dialogeigenschaften in komplexen Anwendungen realisieren
- **Praxis:**
  - Fokus liegt auf der Entwicklung von robusten Echt-Zeit Dialogstrategien zur Behandlung häufiger Dialogeigenschaften in einfachen Anwendungen

## Dialog Management: Beispiel

- System for providing movie showtime and theater playlist information
- Functionality:
  - Task: provide simple movie information
  - Dialogue: allow system-initiated subdialogues
- Process:
  - Variable-initiative

## Dialog Management: Beispiel

<U has task & dialogue initiative>

U: When is Antz playing at the Fairmont theater?

<S has dialogue initiative (low score for theater)>

<goal: confirm Paramount theater>

S: Did you say the Paramount theater?

U: No, the Fairmont theater.

<S has no initiative>

<goal: answer user question>

S: Antz is playing at the Fairmont theater at 7pm.

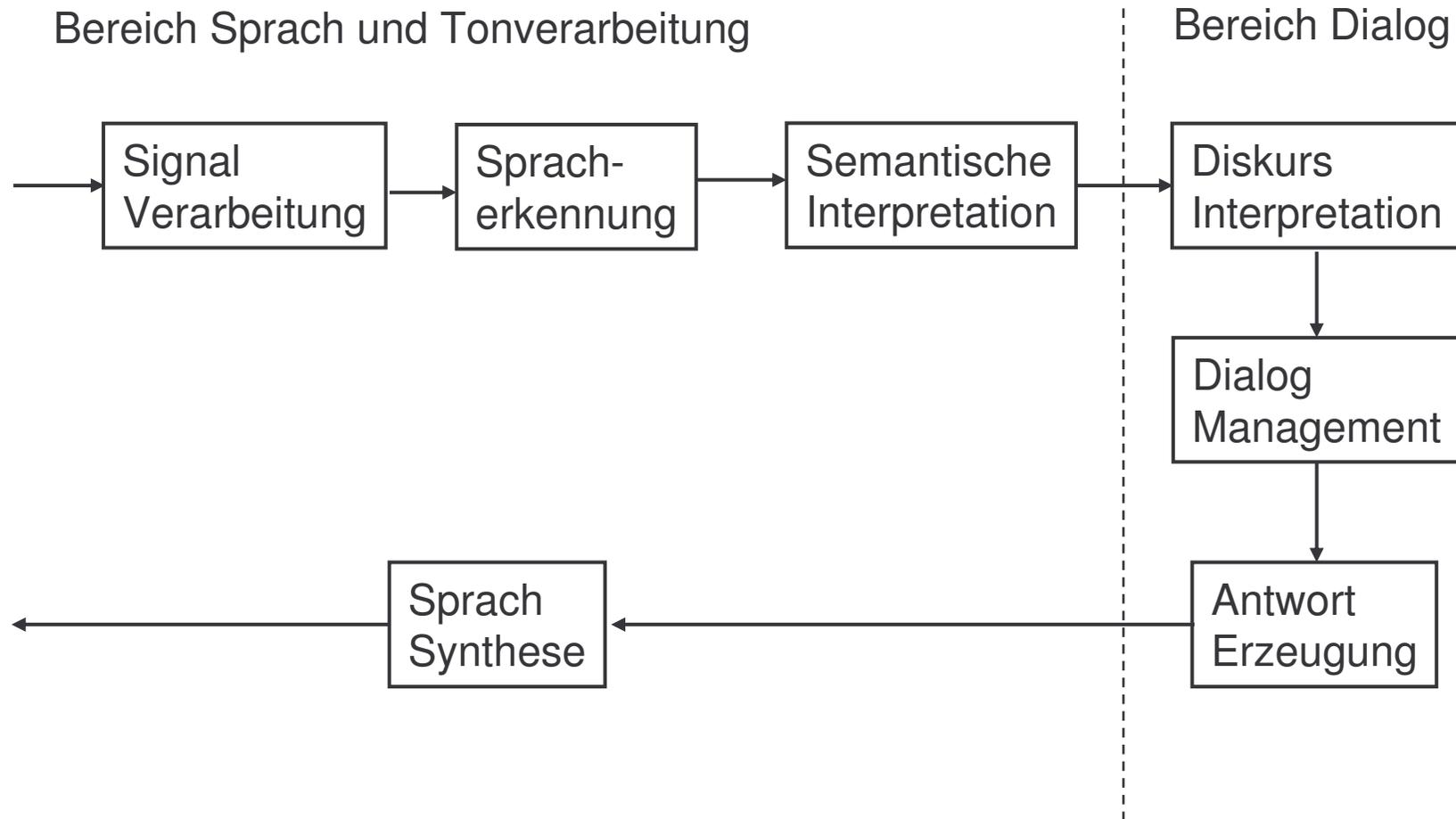
U: When is it playing at the Summit theater?

<S has task & dialogue initiative>

<goal: notify failure & suggest alternative>

S: Antz is not playing at the Summit theater, but it's playing at the AMC theater in Summit at 8pm.

# Datenfluß



# Antworterzeugung

- Ziel: Erzeugung natürlichsprachlicher Aussagen zum Erreichen der vom Dialogmanager ausgewählten Ziele
- Fragestellungen:
  - Auswahl des Inhalts: Was ist zu sagen?
  - Realisierung der Ausgabe (Surface generator):  
Wie ist es zu sagen?
- **Generation gap:**  
Diskrepanz:
  - Auswahl des Inhalts liegt im Bereich der Diskursplanung
  - Realisierung der Ausgabe ist Teil des Sprach und Tonverarbeitung

# Antwörterzeugung: Auswahl des Inhalts

U: Wird Silentium im Royal Filmpalast gezeigt

- Gestaltung der Antwort

- Minimal (wortkarg)

S: Nein

- Vollständige Antwort (mit Absicherung der Eingabe)

S: Nein, Silentium wird nicht im Royal gezeigt.

- Vollständige Antwort mit hilfreichem Hinweis

S: Nein, Silentium wird nicht im Royal gezeigt, das Kino wird gerade renoviert.

→ Erweiterung um Informationen die Dialoge vereinfachen:

- Hinweise
- Hintergründe für einen Vorschlag (Motivation)
- Befähigung des Users Entscheidungen zu treffen. (enablement)

# Antworterzeugung: Auswahl des Inhalts

- Wissensquellen:
  - Domainwissen
  - Annahmen des Users
  - Usermodell: Charakteristika, Präferenzen, etc.
  - Dialoghistorie (Kontext)
- Auswahlverfahren:
  - Schemas: Prädikatenmuster
  - Regelbasierte Generierung
  - Geplante Generierung:
    - Rezepte: Template für Einzelaktionen
    - Planer: Erzeugung von Vorgehensweisen zu gegebenen Zielen
  - Fallbasiertes Schließen (vgl. Zustandsgrammatiken)

# Antworterzeugung: Realisierung der Ausgabe

- Ziel: Festlegen wie der ausgewählte Inhalt in natürliche Sprache verpackt wird
- Beispiele:
  - Silentium wird im Monopol gezeigt.
  - Das Monopol zeigt Silentium.
  - Silentium läuft im Monopol.
  - Im Monopol läuft Silentium.
  - Monopol heist das Kino in dem Silentium läuft.
- Fragestellung:
  - Konstruktion einer Struktur von Abschnitten
  - Lexikalische Auswahl

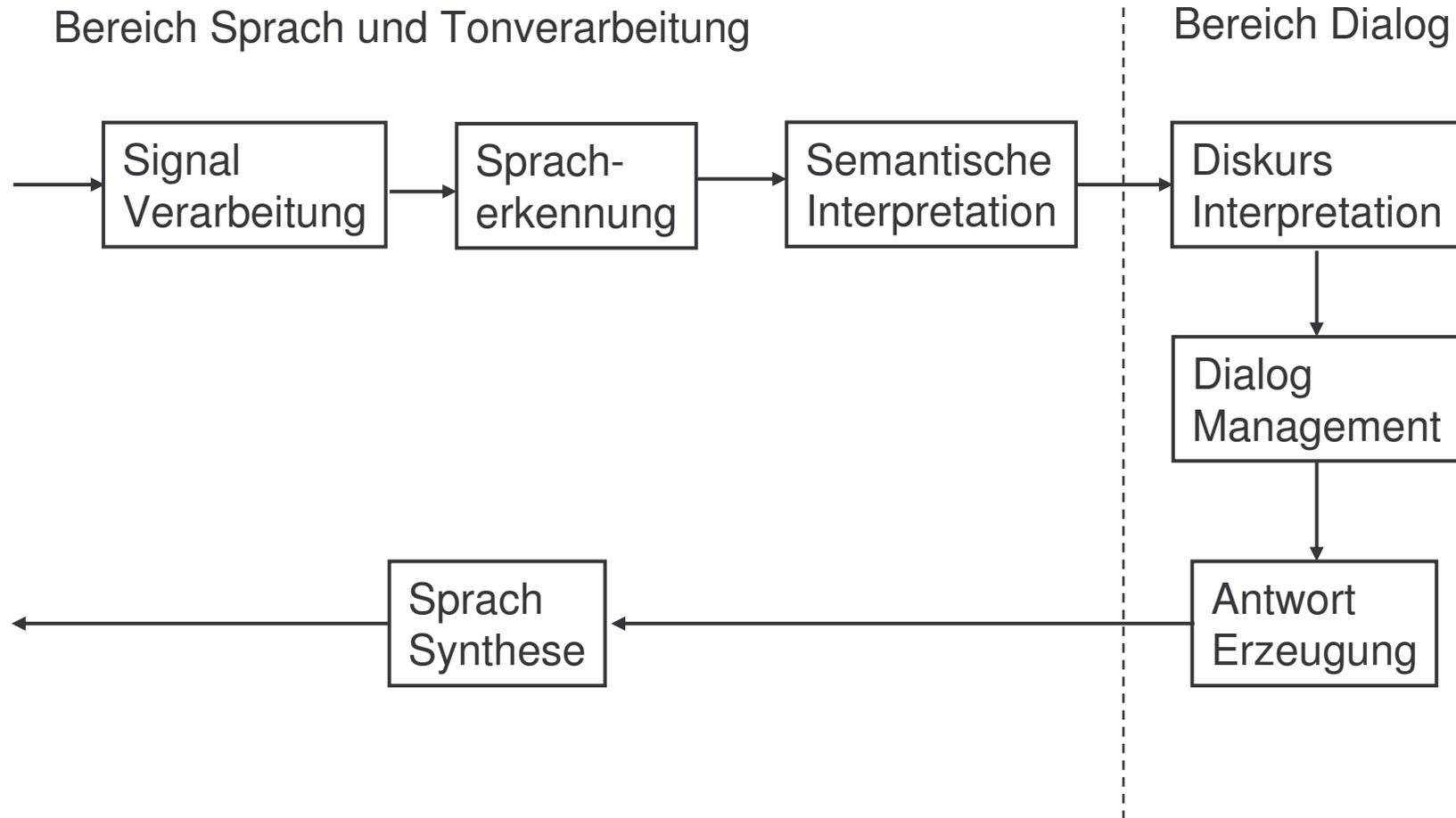
# Antworterzeugung: Realisierung der Ausgabe (Theorie)

- Typische Surface Generatoren arbeiten mit folgendem Input:
  - Umzusetzende Semantische Repräsentation (Wortlaut)
  - Struktur von Abschnitten für die generierte Aussage
- Basierend auf einer Grammatik wird die semantische Repräsentation in eine Aussage überführt

# Antworterzeugung: Realisierung der Ausgabe in der Praxis

- gespeicherte Aussagen:
  - Fix vorgegebene Aussagen:
    - Begrüßungstexte
    - Generics: “Bitte wiederholen Sie Ihre Eingabe”
  - Ermöglicht aufgezeichnete Eingabeaufforderungen als Sprachausgabe zu verwenden
- Generierung mittels Templates:
  - Templates nach Anwendungsfall
    - Mitteilungen: “Your call is being transferred to X.”
    - Informieren: A,B,C,D, and E are playing at the F theater.
    - Klärung: Did you say X or Y?
  - Benötigt das Zusammensetzen aus vorher aufgenommenen Soundsegmenten oder ein komplettes TTS-System

# Datenfluß



# Spracherzeugung / Synthese

→ Zum Ende des Semesters

# Qualitätskriterien

- Ziel: Messung von Leistung und Güte des Dialogsystems
- Zentrale Probleme:
  - Abgrenzung richtig / falsch
  - Was sind Faktoren die gute von schlechten Systemen abgrenzen?
  - Schwierige Definition eine Metrik für “Güte” einzuführen
  - Zusammenwirken und Kompensation einzelner Metriken
  - Zeitaufwendig für dynamische Systeme Daten zu sammeln

## Qualitätskriterien (2)

- Meßbare Größen
  - objektiv:
    - Erfolgsquote
    - Wortfehlerquote
    - Dialoglänge
    - Anzahl korrigierender Subdialoge
    - Performance
  - subjektiv:
    - Natürlichkeit der Sprache
    - Natürlichkeit des Dialogs

## Standard Produkte - Dialogsysteme

- Nuance <http://www.nuance.com/demo/index.html>
- SpeechWorks <http://www.scansoft.com/speechworks/vocon/>
- MIT Spoken Language Systems Laboratory  
<http://groups.csail.mit.edu/sls/sls-blue-noflash.shtml>
- IBM <http://www.scansoft.com/viavoice/>

# OpenSource – Verfügbare Software

- Sphinx  
<http://sphinx.sourceforge.net/>
- FreeTTS  
<http://sourceforge.net/projects/freetts/>
- Java Speech API  
<http://java.sun.com/products/java-media/speech/>
- (optional) VoiceXML  
Tutorials: <http://www.vxml.org/>

# Quellen

- <http://www.colloquial.com/carp/Publications/acl99.ppt>
- <http://www.aliasi.com/lingpipe>
- <http://www.vxml.org/>
- <http://java.sun.com/products/java-media/speech/>
- [http://www.ling.gu.se/~sl/dialogue\\_links.html](http://www.ling.gu.se/~sl/dialogue_links.html)
- <http://communicator.colorado.edu/>