Human-Computer Interaction

Session 11
Natural Language & Dialog

Overview: machines as...

tools ➜ operate

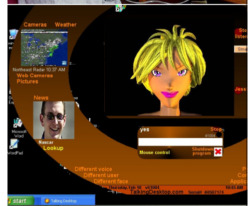
smart tools ➜ instruct

Spoken Language Dialogue Systems

assistants ➜ converse

companions ➜ collaborate






History of user interfaces

Year	Paradigm	Implementation
1950s	None	Switches, punched cards
1970s	Typewriter	Command-line interface
1980s	Desktop	Graphical UI (GUI), direct manipulation
1980s+	Spoken Natural Language	Speech recognition/synthesis, **Natural language processing**, **dialogue systems**
1990s+	Natural interaction	Perceptual, multimodal, interactive, conversational, tangible, adaptive
2000s+	Social interaction	Agent-based, anthropomorphic,social, emotional, affective, collaborative

What is a dialogue?

- multiple participants exchange information
- all participants pursue (ideally) the same goal
- discourse develops over the dialogue
- some conventions and protocols exist

- general structure
  - Dialogue = [episodes]+ (topic changes)
  - Episodes = [turn]+ (speaker changes)
  - Turn = [utterance]+ (function changes)

A lot to be handled...

- in both monologue and dialogue
  - information status: what is given, what is new?
  - coherence: how do the utterances fit together?
  - references: what is being referred to?
  - speech acts: what is the intention of the speaker?
  - implicature: what can be inferred from it?
- +only in dialogue
  - turn-taking: who has the the right to speak?
  - initiative: who is seizing control of the dialogue?
  - grounding: what info is settled between the speakers?
  - repair: how to detect and repair misunderstandings?

- Simplifications and limitations in practical systems
  - controlled language
  - narrow domain
  - explicit, direct meaning
  - system initiative
  - clear turn structure
  - slow interaction cylces

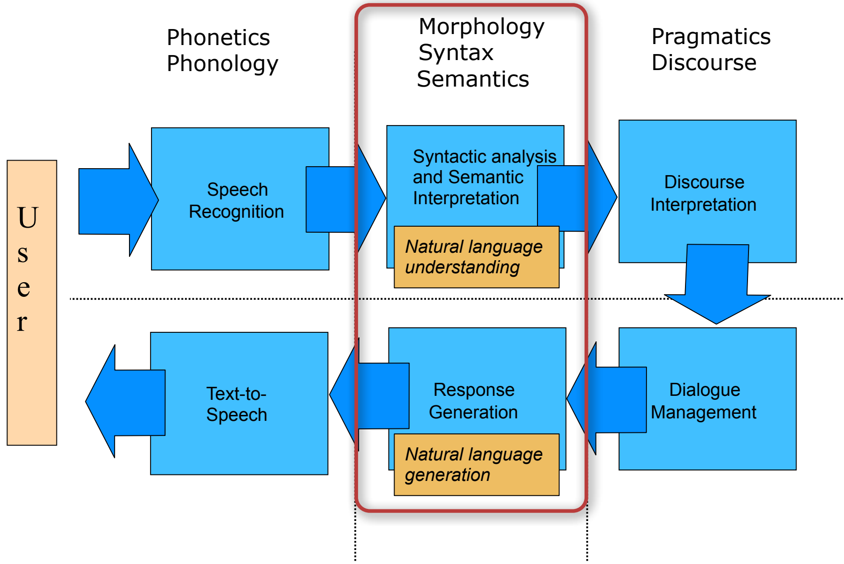
Voice Command

Current automotive speech technology at BMW
- Artikel auf *Spiegel Online* vom 25.6.2009

Voice Command

Automotive voice command (BMW)

Principled SLDS structure



www.chatbots.org

Ohne Syntax und Semantik?

"*keyword-spotting*"

- durchsuchen der Benutzereingabe nach bestimmten Schlüsselworten, z.B. "Wetter", und generieren einer Antwort, die zum Schlüsselwort passt
- Einfach, aber besser skalierbar (grosse Zahl an Regeln)
- Grundlage vieler Chatbots
  - Eliza (Weizenbaum, 1969)
  - ALICE (http://www.alicebot.org/)
  - Jabberwacky.com
  - Anna (www.ikea.de)
- bereits bei einfachen syntaktischen Kniffen überfordert

IKEA

```
Benutzer: "Ich möchte auf keinen Fall über's Wetter reden!"
Bot: "Gern! Hier in Bielefeld regnet es mal wieder."
```

Natural language understanding

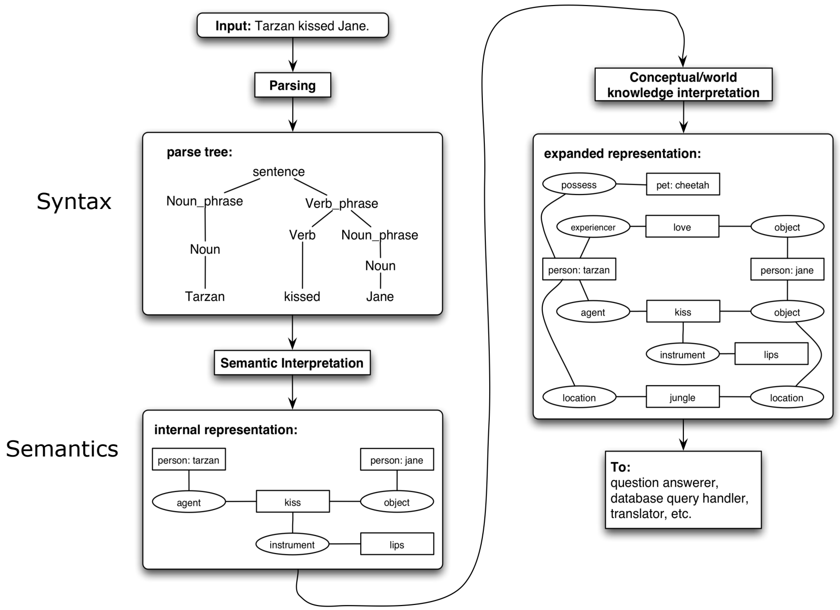
Tree classical steps:

1. Syntax analysis/parsing:
- Determine sentence structure from words

2. Semantic interpretation/understanding:
- Determine word meanings and the overall meaning of their composition in the sentence

3. Discourse interpretation/pragmatic analysis:
- Use context information to complete and disambiguate sentence meaning
- Determine intention behind the sentence

Allen J. (1995)
Natural Language
Understanding.

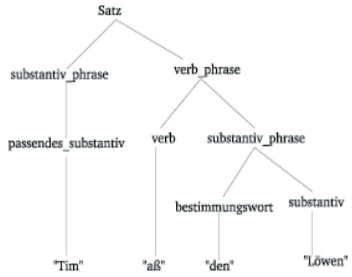


Syntax analysis - parsing

Ziel: Baumartige Zerlegung des sprachlichen Ausdrucks in seine Komponenten gemäß einer Grammatik

```
PARSE ("the dog is dead", G):
      [S: [NP: [Article: the][Noun: dog]]
      [VP: [Verb: is][Adjective: dead]]]
```

- Grammatik: Formale, endliche Beschreibung der *Struktur* aller Elemente einer (oft unendlichen) Sprache
- Parsing = Suchen nach einer möglichen Ableitung eines Satzes in einer Grammatik → Ableitungsbaum
- Beispiel für „Tim aß den Löwen"



Semantic interpretation

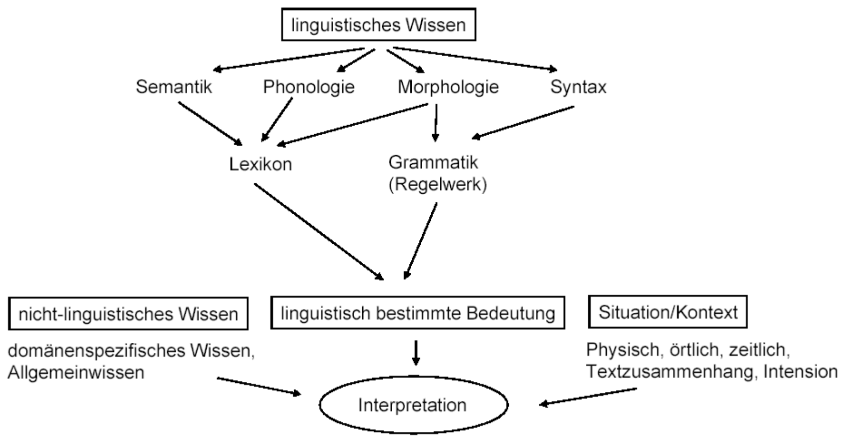
- Aufgabe: *Bedeutungsrekonstruktion*
  - Was ist die *Bedeutung* von *„Er beginnt um zwei im Raum V2-122." ?*
- Unterscheide:
  - **Semantisches Potential**: Linguistisch bestimmte Bedeutung, lässt sich allein mit linguistischem Wissen ermitteln

$$Begin(e,t,l) \wedge Event(e) \wedge Time(t) \wedge Location(l)$$
$$\wedge\ Equal(t,2) \wedge Room(l, V2-122, ?b)$$

  - Aktueller **semantischer Wert**: Volle Interpretation unter Anwendung nicht-linguistischens Wissens (Kontext, Domäne, Welt):

$$Begin(e,t,l) \wedge Event(e) \wedge Time(t) \wedge Location(l)$$
$$\wedge\ Equal(t,2) \wedge Room(l, V2-122, ?b)$$
$$\wedge\ Talk(e,s,l) \wedge Proffessor(s, Cambridge)$$
$$\wedge\ Name(s, Steven-Hawking) \wedge Building(b, Uni-Bielefeld) \wedge \ldots$$

Semantic interpretation



Semantic interpretation

Ziel: Bestimmung des semantischen Potenzials

- Umformung des *Parse*-Baumes in eine *interne Repräsentation* (z.B. Prädikatenlogik, Frames, ...)
- Zwei wesentliche Schritte:
  1. **Lexikalische Semantik**: Bestimmung der Bedeutung einzelner Wörter
     - Probleme: Homonymie, Polysemie (bank/bank), Synonyme (big/large), Antonyme (boy/girl, hot/cold)
     - Resourcen, z.B. *WordNet* (http://wordnet.princeton.edu/)
  2. **Satzsemantik**: Konstruktion der Gesamtbedeutung aus den Einzelbedeutungen (*kompositionelle* Semantik),
     - häufig anhand des *Parse*-Baums, erweitert mit sem. Kategorien (Name, Aktionsbeschreibung, etc.) *syntaktisch-semantisches Parsing*

Discourse interpretation

Ziel: Von Satzsemantik zu Text-/Diskurssemantik/sem. Wert

- Nötige Wissensquellen (über ling. Wissen hinaus):
  - Domänenwissen (banking transaction)
  - Diskurswissen (satzübergreifend)
  - Weltwissen (*Common-sense knowledge*, Situationswissen)
- Beispiel:

U: I would like to open a fixed deposit account.

S: For what amount?

U: Make it for 8000 Rupees.

S: For what duration?

U: What is the interest rate for 3 months?

S: Six percent.

U: Oh good then make it for that duration.

Discourse/pragmatic interpretation

- Referenzauflösung*:* Worauf wird Bezug genommen?
  - Ellipsen: ausgelassene Wörtern oder Phrasen
  - Anaphern: "John likes that blue car. He buys it."
- Intentionserkennung: Was will der Sprecher?
  - "Do you have the time?" → will die Zeit wissen
  - "When is the last train to London?" → will nach London
- Informationsstruktur: Was ist bekannt, was neu?
- Rhetorische und narrative Struktur: Wie ist der Bezug zum vorher Gesagten?

⇨ Vielfach unterspezifierte Fragen, benötigen „Ppagmatische Inferenzen“ unter Berücksichtigung des Diskurskontext; siehe später

Natural Language generation (NLG)

- **Goal**:
  - produce understandable and appropriate output in natural language, along with prosodic information
- **Input**:
  - some underlying non-linguistic representation of information
- **Result**:
  - text to speak, prosodic information
- Knowledge sources required:
  - linguistic knowledge (of language)
  - domain and world knowledge

E. Reiter & R. Dale (2000) *Building Natural Language Generation Systems*. Cambridge University Press.

Natural Language Generation

- Simplest generation method is using templates, mapping representation straight to text template (with variables/ slots to fill in).
  - loves(X, Y) → X "loves" Y
  - gives(X, Y, Z) → X "gives the" Y "to" Z
- Templates are very rigid, much more to NLG in general..
  - Consider "John eats the cheese. John eats the apple. John sneezes. John laughs."
  - Better: "John eats the cheese and apple, then sneezes. He then laughs."
- Getting good *style* involves working out how to map many facts to one sentence, when to use pronouns, when to use connectives like "then" etc.

Tasks in NLG

Content Planning
- Content Determination
- Discourse planning
- (Document Structuring)
- what to say, how to order and structure it

⇩

Micro-planning
- Aggregation
- Lexicalisation
- Referring Expression Generation
- how to break it up into sentences and words, how to refer to objects

⇩

Surface Realisation
- Linguistic Realisation
- Structure Realisation
- How to express things in terms of grammatically correct sentences

1. Content Planning

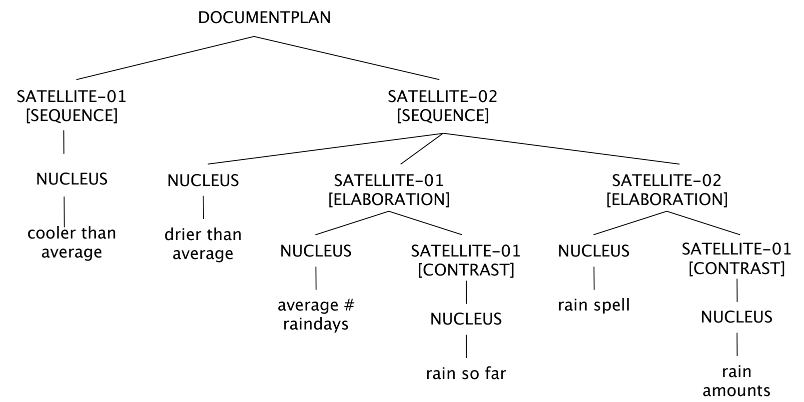
Goals:
- determine *what* information to communicate (content)
- determine *structure* of this information to make a coherent text/discourse

Results: *messages*, predefined data structures that…
- correspond to informational elements (units)
- collect underlying data in ways convenient for ling. expression

- Essentially, a domain-dependent expert-system task
- Common approaches:
  1. based on observations about common utterance structures
  2. based on reasoning about discourse coherence and the purpose of the utterance

Content plan (aka. document plan)

- Tree structure with messages at its leaf nodes
- Rhetorical Structure Theory (RST): distinction between *nucleus,* the central segment, and the *satellite*, the more peripheral one, and relations between them (e.g. elaboration, contrast, …)
- Example from *WeatherReporter* system:



2. Microplanning

Goal:

- convert a content plan into a sequence of sentence or phrase specifications

Tasks:

- **Aggregation** via *conjunction, ellipsis, or embedding*
  - Heavy rain fell on the 27th and [] on the 28th.
- **Lexicalisation***:* choosing word lemmas
- **Reference**: how to refer to entities
  - initially: full name, relate to salient object, specify location
  - subsequently: Pronouns, definite NPs, proper names, possibly abbreviated

3. Surface realisation

Goal:

convert text specifications into actual text

Purpose:

hide peculiarities of English (or whatever the target language is) from the rest of the NLG system

Tasks:

- *Structure realisation*
  - Choose markup to convey document structure
- *Linguistic realisation* using specialized grammars
  - Insert function words
  - Choose correct inflection of content words
  - Order words within a sentence
  - Apply orthographic rules

Remarks

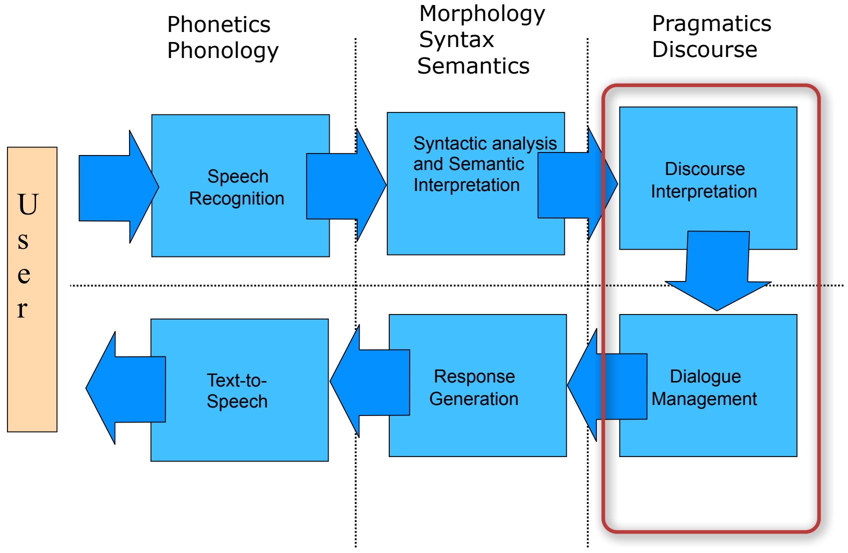
- problems like NLU and NLG are still challenges and not generally solved (compared to TTS)
  - in practice, often circumvented by design
  - SLDS successful where this is possible (phone services, call center, ticketing, etc.)
- several toolkits & standards for directly scripting spoken dialgue behavior exist
  - VoiceXML (Voice Extensible Markup Language)
  - SALT (Speech Application Language Tags)
  - X+V (XHTML+Voice)

"Speech is the bicycle of user-interface design, it is great fun to use [...], but it can carry only a light load. Sober advocates know that it will be tough to replace the automobile: graphic user-interfaces", Ben Shneiderman, 1998

Main problems in today's systems

- Lack of understanding
  - only little of what is said or communicated can be sensed and recognized by computers
  - only little of what is really important is said explicitly
- Lack of knowledge
  - about the world (commonsense), situation, discourse, communicative system (language, other modalities)
- Lack of expressivity
  - only limited ways to communicate information
- Lack of interactivity
  - slow responses, long latencies
  - no adaptation, recipient design, alignment

Classical SLDS structure



Resolve references

- Ellipsis
  - People often utter partial phrases to avoid repetition
    A: At what time is "Titanic" playing?
    B: 8pm
    A: And "The 5th Element"?
  - Necessary to keep track of the conversation to complete such phrases
- Some words are only interpretable in conext
  - Anaphora: "I'll take it", he said.
  - Temporal/spatial: "The man behind me will be dead tomorrow."

Handle information structure

Distinguish two parts of one utterance

- Theme:
  Part of a proposition that repeats known information to create cohesive connection to <u>previous</u> propositions („discourse cohesion")
- Rheme:
  Part of a proposition that contributes <u>new</u> information

Example: Who is he? He is a student.

Theme (He is) — Rheme (a student.)

- There can be purely rhematic/thematic utterances

(Bolinger; Halliday, 1960's)

Understand speech acts

- Every utterance is an action performed by the speaker in a real speech situation
- Obvious in *performative* sentences: „I <u>name</u> this ship titanic.", „I <u>bet</u> you 5 bugs."
- Any sentence in a speech situation constitutes three kinds of acts:
  - Locutionary act: the utterance of the sentence „I'm cold."
  - Illocutionary act: the action in uttering it (asking, answering, commanding, ...) → informing that I'm cold.
  - Perlocutionary act: the production of effects upon the addressee and ultimately the world → get window closed
- speech act explicates the illocutionary act

Austin (1962), Searle (1975)

Understand indirect meaning

S: „*What day in May do you want to travel?*"

U: „*I have a meeting from the 12th the 15th*."

U does not answer directly, expects hearer to draw certain inferences

Cooperative Principle: hearer can draw inferences because they assume conversants are cooperative and follow four maxims (Paul Grice, 1975):

- Maxim of Quantity: Be exactly as informative as required
- Maxim of Quality: Make your contribution one that is true
- Maxim of Relevance: Be relevant.
- Maxim of Manner: Be understandable, unambiguous, brief, and orderly

→ Maxim of Relevance allows S to know that U wants to travel by the 12th.

Understand grounding

Allwood, 1976;
Clark & Shaefer, 1989

- Interlocutors are trying to establish common ground, a set of mutual beliefs
- Listener must ground a speaker's contribution by acknowledging it, signaling understanding or agreement
- Various ways to do this:

  S: „I can upgrade you to an SUV at that rate."

  - Continued attention/permission to proceed - U gazes appreciatively at S
  - Relevant next contribution - U: „Do you have an Explorer available?"
  - Acknowledgement, "backchanneling" - U: „Ok/Mhm/Great!"
  - Display/repetition - U: „You can upgrade me to an SUV at the same rate?"
  - Request for repair- U: „Huh?"

Manage initiative

Control - the ability/license to bring up new topics, to start tasks, to pose questions, etc.

- System-initiative:
  system always has control, user only responds to system questions

- User-initiative:
  user always has control, system passively answers user questions
- Mixed-initiative:
  control switches between system and user either using fixed rules or dynamically based on participant roles, dialogue history, etc.

Initiative strategies

- System initiative (spoken "form filling")
  S: Please give me your arrival city name.
  U: Baltimore.
  S: Please give me your departure city name
  U: Boston
  S:…

Rigid, restricted vocabulary, rigid, NLP easy and more accurat,

- User initiative
  U: When do flights to Boston leave?
  S: At 8:30 AM and 3:45 PM.
  U: How much are they?
  S:…

requires good NLP, users must be aware of possible words

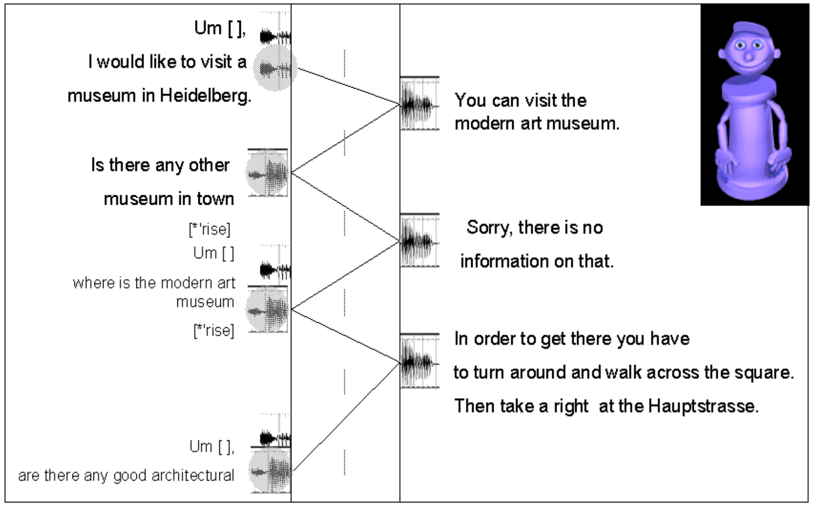
- Mixed initiative
  S: Where are you traveling to?
  U: I want to go to Boston.
  S: At time do you want to fly?
  U: Are there any cheap flights?

natural, open, unpredictable, hard to model, requires NLP and complex dialogue manag.

Manage turn-taking

- People know well when they can take the turn
  - Only little speaker overlap (~5% in English)
  - But little silence between turns either, a few of 1/10 s
    - Less than needed to plan motor routines for speaking
    - Speakers usually start motor planning before previous speaker has finished talking !!
- How do we know?
  - Schegloff (1968): *Adjacency pairs* set up speaker expectations and give rise to discourse obligations
    - QUESTION → ANSWER, REQUEST → GRANT, ...
    - Silence inbetween is dispreferred → pauses disturb users!
  - Sacks et al. (1974): *transition-relevance places* and rules that govern turn-taking, e.g.
    - If current speaker does not select next speaker, any other speaker may take next turn

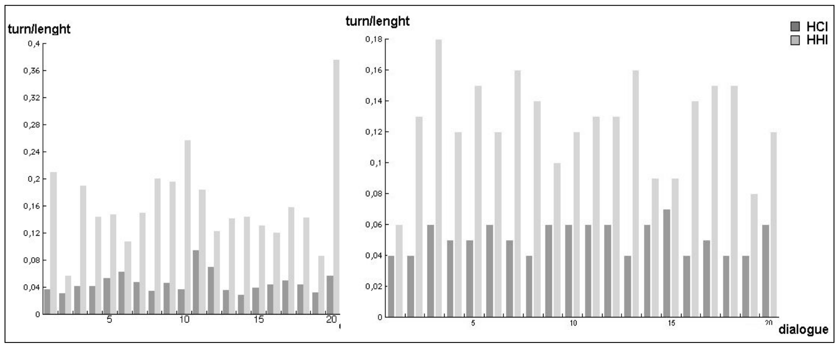
Usual structure of HCI dialogues

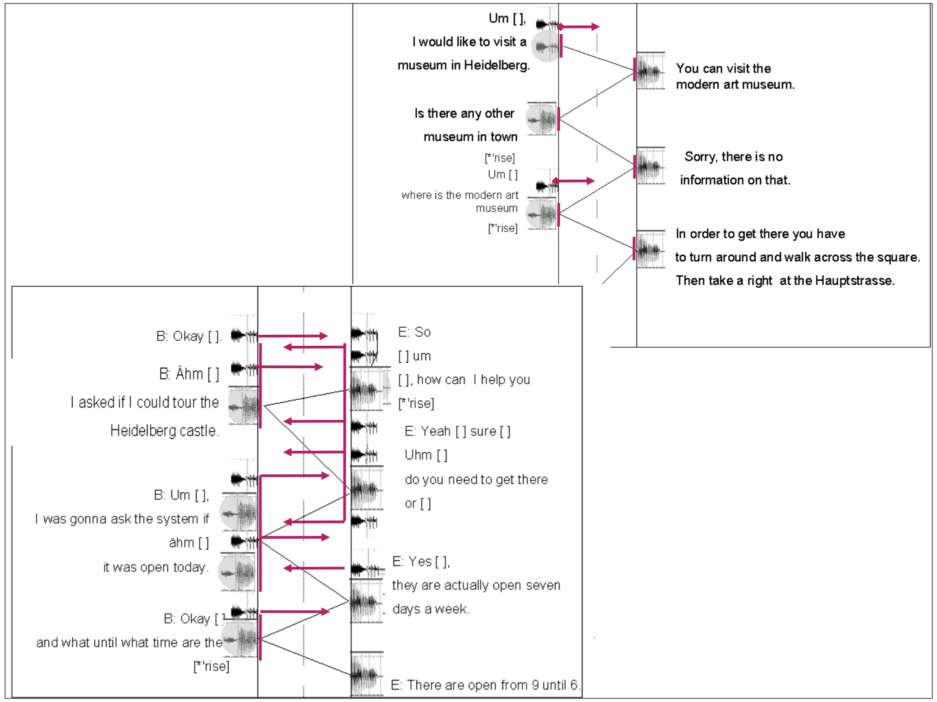


Measuring dialogue efficiency

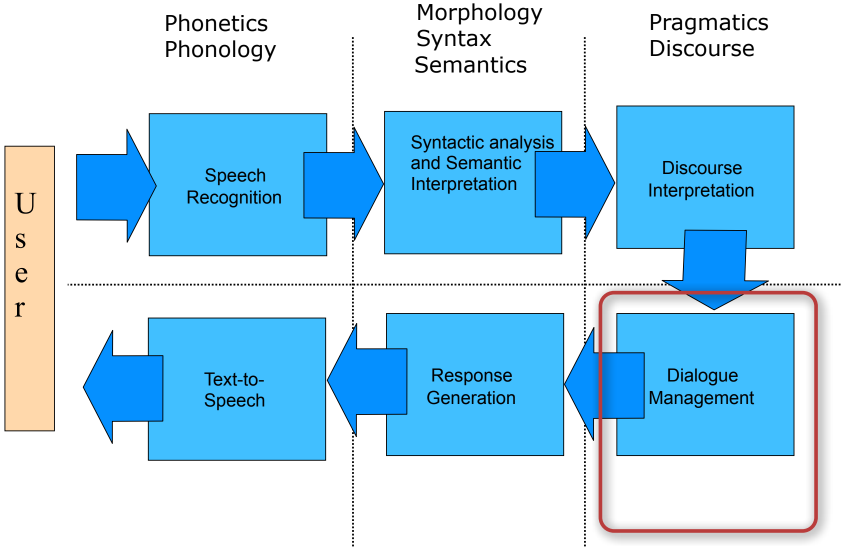
Highly significant loss of dialogical efficiency in HCI vs. HHI using the PARADISE metric: Walker et al (2001) - dialogue turns / dialogue length

Robert Porzel, Uni Bremen





Classical SLDS structure



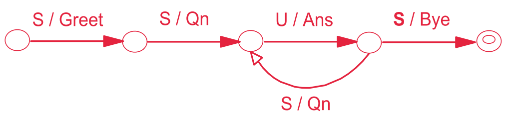
Dialog management

four general approaches

- no dialogue management (turn-to-turn)
  ➜ chatbots
- dialogue grammars (fixed structure)
  ➜ state-based, finite state machines/automata
- form-filling (fixed content)
  ➜ frame-based
- agent-based, plan/intention-based

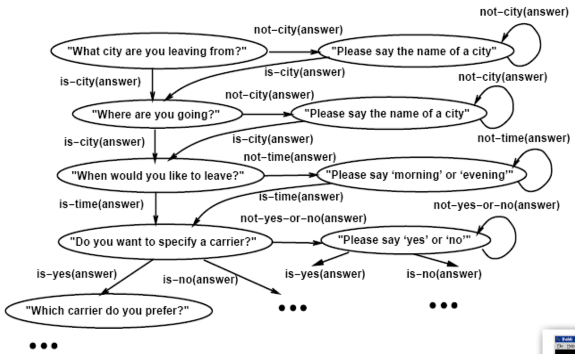
Finite state machine DM

Finite State Dialogue Grammar

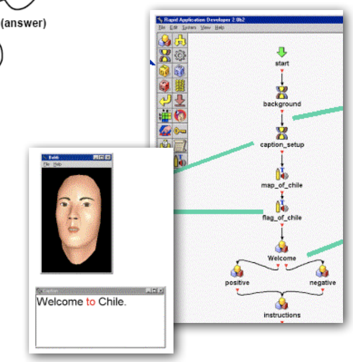


- Graph specifies all legal dialogues (dialogue grammar)
  - Nodes: system's questions
  - Transitions: possible paths through the network
  - Each state represents a stage in the dialogue ("now"), rarely with complete dialogue history
- System has initiative
- Context is fixed by the question being asked
- Used widely in commercial applications

Finite state machine DM



(Jurafsky & Martin, 2000)



Do-it-yourself example: CSLU Toolkit
http://cslu.cse.ogi.edu/toolkit/

Frame-based DM

Prompt: Where and when do you want to travel?
Grammar: <input of departure and arrival city, date and time>
Help: Please specify the departure and arrival city, date and time

FROM
Prompt: From which city are you leaving?
Grammar: <input of a city>
Help: Tell me the name of the city you want to leave from

TO
Prompt: To which city do you want to travel?
Grammar: <input of a city>
Help: Tell me the name of the city you want to travel to

WHEN
Prompt: When do you want to travel?
Grammar: <input of date and time>
Help: Please specify date and time of your journey

Filled: SELECT * FROM connections WHERE departure like 'FROM' AND destination like 'TO' *AND* time like *'WHEN'*

Frame-based DM

- frame: template containing slots to be filled
  - destination: London, date: unknown, time of departure: 9
- questions to fill slots, conditions at which they can be asked
  - condition: unknown(origin) & unknown(destination)
    question: "Which route do you want to travel?"
  - condition: unknown(destination)
    question: "Where do you want to travel to?"
- decision on next question based on filled/empty slots
- system initiative, more flexible, dialogue reflects current state of the system (transparent)
- bad for negotiation, planning, mixed-initiative

Frame-based + FSM-based DM

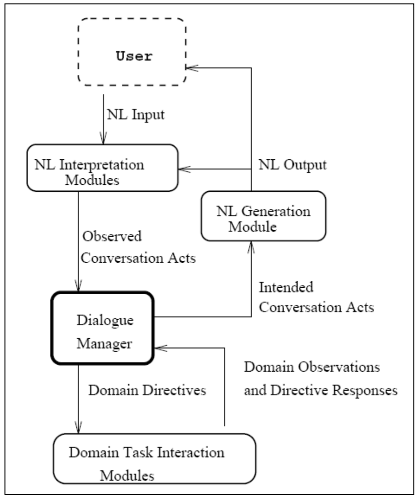
- Commercial standards, in bundles with ASR/TTS
  - VoiceXML
  - SALT
- Frame-based DM, combined with FSMs for single fields/slots
  - structured input patterns
  - parsing and assigning to values
  - clarification subdialogues

```
<form id="start">
  <field name="answer">
    <noinput> Hey, don't sleep! </noinput>
    <nomatch> say 'yes' or 'no' </nomatch>

    <prompt> Are you sleepy? </prompt>

    <grammar root="main" tag-format="semantics/1.0-literals">
      <rule id="main" scope="public">
        <one-of>
          <item><ruleref uri="#yes"/><tag>yes</tag></item>
          <item><ruleref uri="#no"/><tag>no</tag></item>
        </one-of>
      </rule>
      <rule id="yes">
        <one-of>
          <item>yes</item>
          <item>yeah</item>
          <item>yep</item>
          <item>sure</item>
        </one-of>
      </rule>
      <rule id="no">
        <one-of>
          <item>no</item>
          <item>not</item>
          <item>nope</item>
        </one-of>
      </rule>
    </grammar>

    <filled>
      <if cond="answer=='yes'">
        So you are sleepy. Me too.
      <else/>
        So you are not sleepy. But I am.
      </if>
    </filled>
  </field>
</form>
```


Intention-/plan-based DM

- Idea: dialogue = collaboration of intentional agents on solving a task
  - there are goals to be reached
  - plans are made to reach those goals
  - the goals and plans of the other participants must be inferred or predicted
  - goals may involve changing the beliefs of others
  - models of the mental state of participants are used
- draws on methods from Artificial Intelligence
- permits more complex interaction between user, system, and underlying application
- allows for mixed-initiative dialogue

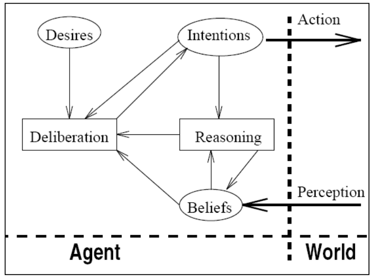
Example: TRAINS (Traum, Allen, 1996)



- Design system as agent with own mental states (Bratman, 1987)
  - Beliefs: world model
  - Desires: goals
  - Intentions: plans to pursue

  Reasoning: derive new beliefs

  Deliberation: decide actions



TRAINS dialogue manager

- Reactive: system will deliberate as little as possible until it can act, running in cycles
- No long-range plans, one step at a time

- Prioritized list of sources for deliberations
  1. Discourse obligations
  2. Weak obligation: don't interrupt user's turn
  3. Intended speech act (→ NLG + state update)
  4. Weak obligation: grounding (acknowledge, repair)
  5. Discourse goals: proposal negotiation
  6. High-level discourse goals (domain reasoning)

Dialog management

```
DIALOGUE_MANAGER
  while conversation is not finished
   if user has completed a turn
   then interpret user's utterance
   if system has obligations
   then address obligations
   else if system has turn
   then if system has intended conversation acts
      then call generator to produce NL utterances
      else if some material is ungrounded
      then address grounding situation
      else if high-level goals are unsatisfied
      then address goals
      else release turn or attempt to end conversation
   else if no one has turn
   then take turn
   else if long pause
   then take turn
```

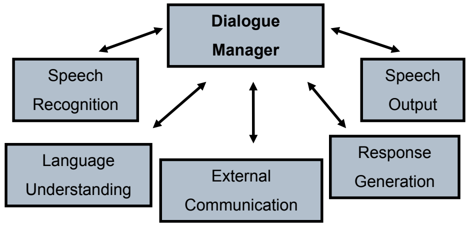
Jurafsky & Martin, 2000

Summary

Features/ dialogue control	State-based	Frame-based	Intention-based
Input	Single words or phrases	NL with concept spotting	Unrestricted NL
Verification	Explicit confirmation of each turn or at end	Explicit & implicit confirmation	Grounding
Dialogue Context	Implicitly in dialogue states	Explicitly represented Control represented with algorithm	Model of System's BDI + dialogue history
User Model	Simple model of user characteristics / preferences	Simple model of user characteristics / preferences	Model of User's BDI

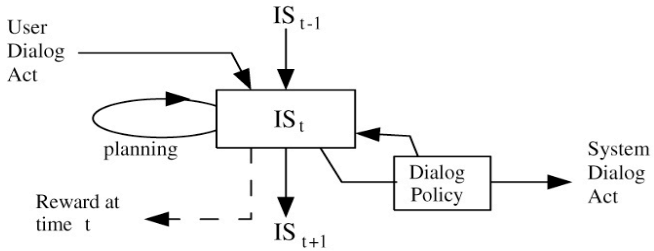
SLDS architectures

- Pipeline structure with message passing
  - classical (see above), but with problems
  - strictly sequential processing
  - only local context for single processing stages
- Blackboard
  - distributed, collaborating agents; no strict process protocol
  - dialogue manager hosts central data structures
  - accounts for importance of context/discourse for all stages



Information State approach

- Central data structure(s) to define conversational state
  - employed in deciding on next actions
  - updated in effect of dialogue acts by either speaker
- operational semantics of plans stated as update rules
- dialogue manager = definition of the contents of the IS + description of update processes



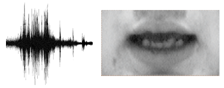
(Traum & Larsson, 2003)

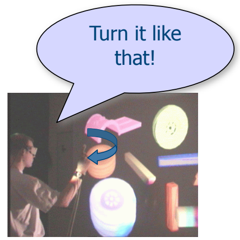
Incremental processing

- Mitigate lack of interactivity
  - Modules process input as it comes in
  - pass on preliminary output for further modules to start processing
  - augment or change it when necessary
  - commit to it once done and certain about it
- Different frameworks being developed
  - Jindigo (KTH Stockholm)
  - InPro (Uni Potsdam/Uni Bielefeld)
  - IPAACA (Uni Bielefeld)

Nonverbal behavior & Multimodality

speech recognition & lip reading





Next session