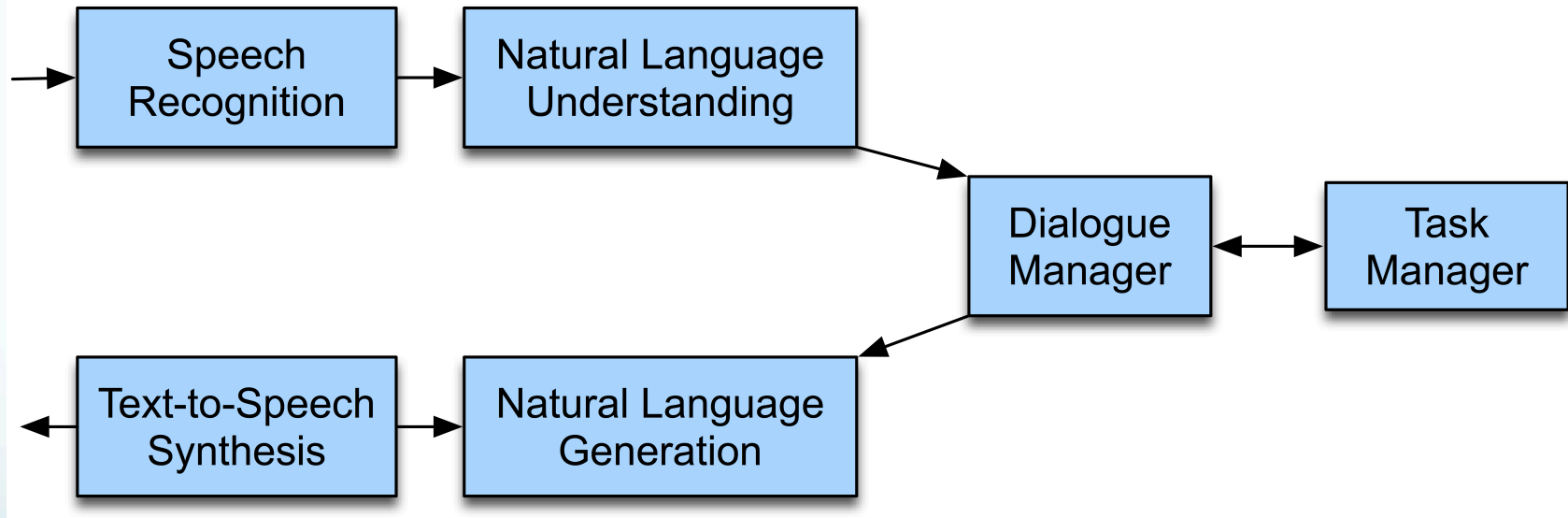


SDS Systems & Components

Ling575
Spoken Dialog Systems
April 9, 2015

Dialogue System Architecture



Speech Recognition

- (aka ASR)
- Input: acoustic waveform
 - Telephone, microphone, and smartphone

Speech Recognition

- (aka ASR)
- Input: acoustic waveform
 - Telephone, microphone, and smartphone
- Output: recognized word string

Speech Recognition

- (aka ASR)
- Input: acoustic waveform
 - Telephone, microphone, and smartphone
- Output: recognized word string
- Requirements:

Speech Recognition

- (aka ASR)
- Input: acoustic waveform
 - Telephone, microphone, and smartphone
- Output: recognized word string
- Requirements:
 - Acoustic models: map acoustics to phone [ae] [k]
 - Pronunciation dictionary: words to phones: cat: [k][ae][t]
 - Grammar: legal word sequences
 - Search procedure: best word sequence given audio

Recognition in SDS

Recognition in SDS

- Create domain specific vocabulary, grammar
 - Typically hand-crafted in most commercial systems
 - Based on human-human interactions
 - Grammars: finite-state, context-free, language model

Recognition in SDS

- Create domain specific vocabulary, grammar
 - Typically hand-crafted in most commercial systems
 - Based on human-human interactions
 - Grammars: finite-state, context-free, language model
- Activate only portion of grammar based on dialog state
 - E.g. Where are you leaving from?

Recognition in SDS

- Create domain specific vocabulary, grammar
 - Typically hand-crafted in most commercial systems
 - Based on human-human interactions
 - Grammars: finite-state, context-free, language model
- Activate only portion of grammar based on dialog state
 - E.g. Where are you leaving from?
 - {I want to (leave|depart) from} CITYNAME {STATENAME}
 - ‘Yes/No’ grammar for confirmations

Natural Language Understanding

- Most systems use frame-slot semantics
Show me morning flights from Boston to SFO on Tuesday
Alternatives:
 - Full parser with semantic attachments
 - Domain-specific analyzers
- SHOW:
- FLIGHTS:
 - ORIGIN:
 - CITY: Boston
 - DATE:
 - DAY-OF-WEEK: Tuesday
 - TIME:
 - PART-OF-DAY: Morning
 - DEST:
 - CITY: San Francisco

Generation and TTS

- Generation:
 - Identify concepts to express
 - Convert to words
 - Assign appropriate prosody, intonation

Generation and TTS

- Generation:
 - Identify concepts to express
 - Convert to words
 - Assign appropriate prosody, intonation
- TTS:
 - Input words, prosodic markup
 - Synthesize acoustic waveform

Generation

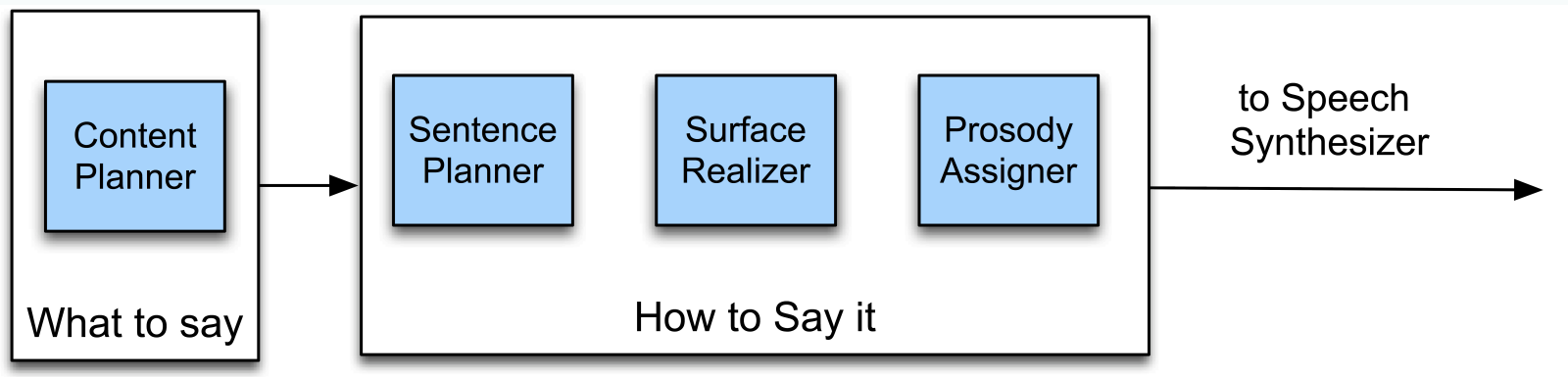
- Content planning:
 - What to say:
 - Question, answer, etc?
 - Often merged with dialog manager

Generation

- Content planning:
 - What to say:
 - Question, answer, etc?
 - Often merged with dialog manager
- Language generation:
 - How to say it
 - Select syntactic structure and words
 - Most common: Template-based generation (prompts)
 - Templates with variable: When do you want to leave CITY?

Full NLG

- Converts representation from dialog manager



Dialogue Manager

- Holds system together: Governs interaction style

Dialogue Manager

- Holds system together: Governs interaction style
 - Takes input from ASR/NLU

Dialogue Manager

- Holds system together: Governs interaction style
 - Takes input from ASR/NLU
- Maintains dialog state, history
 - Incremental frame construction
 - Reference, ellipsis resolution
 - Determines what system does next

Dialogue Manager

- Holds system together: Governs interaction style
 - Takes input from ASR/NLU
 - Maintains dialog state, history
 - Incremental frame construction
 - Reference, ellipsis resolution
 - Determines what system does next
 - Interfaces with task manager/backend app

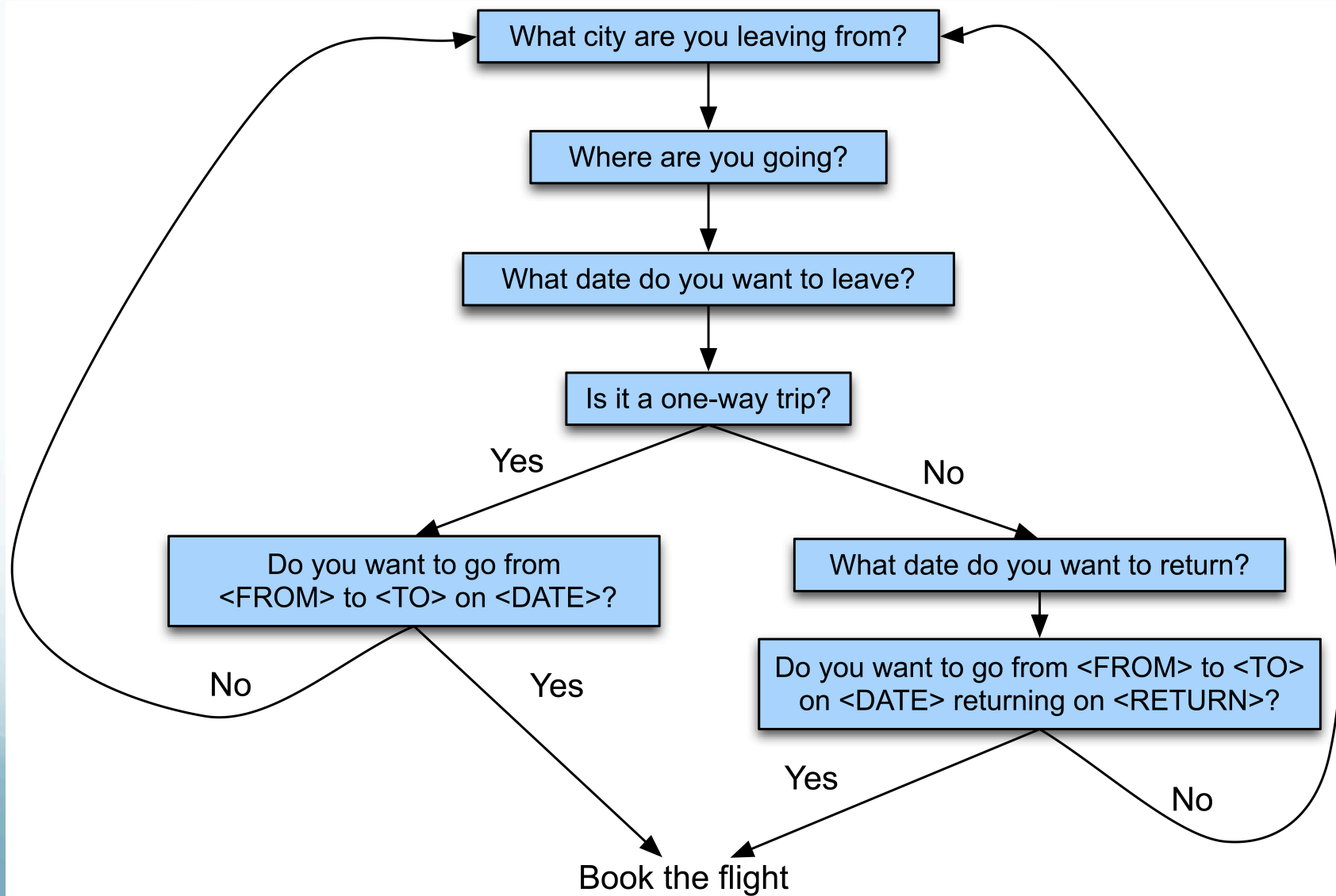
Dialogue Manager

- Holds system together: Governs interaction style
 - Takes input from ASR/NLU
 - Maintains dialog state, history
 - Incremental frame construction
 - Reference, ellipsis resolution
 - Determines what system does next
 - Interfaces with task manager/backend app
 - Formulates basic response, passes to NLG,TTS

Dialog Management Types

- Finite-State Dialog Management
- Frame-based Dialog Management
- Information State Manager
- Statistical Dialog Management

Finite-State Management



Finite-State Dialogue Management

- Simplest type of dialogue management
 - States:
 - Questions system asks user
 - Arcs:
 - User responses

Finite-State Dialogue Management

- Simplest type of dialogue management
 - States:
 - Questions system asks user
 - Arcs:
 - User responses
- System controls interactions:
 - Interprets all input based on current state
 - Assumes any user input is response to last question

Finite-State Dialogue Management

- Initiative:
 - Control of the interaction
- Who's in control here?

Finite-State Dialogue Management

- Initiative:
 - Control of the interaction
- Who's in control here?
 - System!
 - “system initiative”/”single initiative”
 - Natural?

Finite-State Dialogue Management

- Initiative:
 - Control of the interaction
- Who's in control here?
 - System!
 - “system initiative” / “single initiative”
 - Natural? No!
 - Human conversation goes back and forth
- Deploy targeted vocabulary / grammar for state
 - Add ‘universals’ – accessible anywhere in dialog
 - ‘Help’, ‘Start over’

Pros and Cons

- Advantages

Pros and Cons

- Advantages
 - Straightforward to encode
 - Clear mapping of interaction to model
 - Well-suited to simple information access
 - System initiative
- Disadvantages

Pros and Cons

- Advantages
 - Straightforward to encode
 - Clear mapping of interaction to model
 - Well-suited to simple information access
 - System initiative
- Disadvantages
 - Limited flexibility of interaction
 - Constrained input – single item
 - Fully system controlled
 - Restrictive dialogue structure, order
 - Ill-suited to complex problem-solving

Frame-based Dialogue Management

- Essentially form-filling
 - User can include any/all of the pieces of form
 - System must determine which entered, remain
 - Rules determine next action, question, information presentation

Frame-based Dialogue Management

- Essentially form-filling
 - User can include any/all of the pieces of form
 - System must determine which entered, remain
 - Rules determine next action, question, information presentation

Slot	Question
ORIGIN CITY	“From what city are you leaving?”
DESTINATION CITY	“Where are you going?”
DEPARTURE TIME	“When would you like to leave?”
ARRIVAL TIME	“When do you want to arrive?”

Frames and Initiative

- Mixed initiative systems:
 - A) User/System can shift control arbitrarily, any time
 - Difficult to achieve
 - B) Mix of control based on prompt type

Frames and Initiative

- Mixed initiative systems:
 - A) User/System can shift control arbitrarily, any time
 - Difficult to achieve
 - B) Mix of control based on prompt type
- Prompts:
 - Open prompt: 'How may I help you?'

Frames and Initiative

- Mixed initiative systems:
 - A) User/System can shift control arbitrarily, any time
 - Difficult to achieve
 - B) Mix of control based on prompt type
- Prompts:
 - Open prompt: ‘How may I help you?’
 - Open-ended, user can respond in any way
 - Directive prompt: ‘Say yes to accept call, or no o.w.’

Frames and Initiative

- Mixed initiative systems:
 - A) User/System can shift control arbitrarily, any time
 - Difficult to achieve
 - B) Mix of control based on prompt type
- Prompts:
 - Open prompt: ‘How may I help you?’
 - Open-ended, user can respond in any way
 - Directive prompt: ‘Say yes to accept call, or no o.w.’
 - Stipulates user response type, form

Dialogue Management: Confirmation

- Miscommunication common in SDS
 - “Error spirals” of sequential errors
 - Highly problematic
 - Recognition, recovery crucial
- Confirmation strategies can detect, mitigate
 - Explicit confirmation:

Dialog Example

C₁: ...I need to travel in May.

A₁: And, what day in May did you want to travel?

C₂: OK uh I need to be there for a meeting that's from the 12th to the 15th.

A₂: And you're flying into what city?

C₃: Seattle.

A₃: And what time would you like to leave Pittsburgh?

C₄: Uh hmm I don't think there's many options for non-stop.

A₄: Right. There's three non-stops today.

C₅: What are they?

A₅: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.

C₆: OK I'll take the 5ish flight on the night before on the 11th.

A₆: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.

C₇: OK.

Travel Planning

S1: Hello. You've reached the [Communicator]. Tell me your full name
U2: *hi i'd like to fly to seattle tuesday morning*
S3: Traveling to Seattle on Tuesday, August eleventh in the morning. Your full name?
U4: *[John Doe]*
S5: Hello, [John], I'm glad to hear from you again. You can now ask me for help.
S6: OK. Let's see I have a United flight departing Pittsburgh at ten forty five a.m. arriving into Seattle at one fifty seven p.m. Is that OK?
U7: *Yes*
S8: Will you return to Pittsburgh from Seattle?
U9: *no i will go to Los Angeles*
S10: Traveling to Los Angeles. on what date
U11: *Sorry can we go back to talk about the previous leg again*
S12: For the first leg, you are on a United flight 1446 departing Pittsburgh at ten forty five A.M. tomorrow, arriving in Seattle at one fifty seven P.M. Do you want to take that?
U13: *actually i want to leave on wednesday*
S14: a flight on wednesday.
S15: I have a flight departing Pittsburgh at ten forty five a.m., arrives Seattle at one fifty seven p.m. Is that OK?
U16: *Yes*

Figure 19.1 The travel domain: a fragment from a successful conversation between a user (U) and the Communicator system (S) of Xu and Rudnicky (2000).

Dialogue Management: Confirmation

- Miscommunication common in SDS
 - “Error spirals” of sequential errors
 - Highly problematic
 - Recognition, recovery crucial
- Confirmation strategies can detect, mitigate
 - Explicit confirmation:
 - Ask for verification of each input
 - Implicit confirmation:

Dialogue Management: Confirmation

- Miscommunication common in SDS
 - “Error spirals” of sequential errors
 - Highly problematic
 - Recognition, recovery crucial
- Confirmation strategies can detect, mitigate
 - Explicit confirmation:
 - Ask for verification of each input
 - Implicit confirmation:
 - Include input information in subsequent prompt

Confirmation Strategies

- Explicit:

S: Which city do you want to leave from?

U: Baltimore.

S: Do you want to leave from Baltimore?

U: Yes.

U: I'd like to fly from Denver Colorado to New York City on September twenty first in the morning on United Airlines

S: Let's see then. I have you going from Denver Colorado to New York on September twenty first. Is that correct?

U: Yes

Confirmation Strategy

- Implicit:

U: I want to travel to Berlin

S: When do you want to travel to Berlin?

U2: Hi I'd like to fly to Seattle Tuesday Morning

A3: Traveling to Seattle on Tuesday, August eleventh in the morning.
Your full name?

Pros and Cons

- Grounding of user input
 - Weakest grounding
 - I.e. continued att'n, next relevant contribution

Pros and Cons

- Grounding of user input
 - Weakest grounding insufficient
 - I.e. continued att'n, next relevant contribution
 - Explicit:

Pros and Cons

- Grounding of user input
 - Weakest grounding insufficient
 - I.e. continued att'n, next relevant contribution
 - Explicit: highest: repetition
 - Implicit:

Pros and Cons

- Grounding of user input
 - Weakest grounding insufficient
 - I.e. continued att'n, next relevant contribution
 - Explicit: highest: repetition
 - Implicit: demonstration, display
- Explicit;

Pros and Cons

- Grounding of user input
 - Weakest grounding insufficient
 - I.e. continued att'n, next relevant contribution
 - Explicit: highest: repetition
 - Implicit: demonstration, display
- Explicit;
 - Pro: easier to correct; Con: verbose, awkward, non-human
- Implicit:

Pros and Cons

- Grounding of user input
 - Weakest grounding insufficient
 - I.e. continued att'n, next relevant contribution
 - Explicit: highest: repetition
 - Implicit: demonstration, display
- Explicit;
 - Pro: easier to correct; Con: verbose, awkward, non-human
- Implicit:
 - Pro: more natural, efficient; Con: less easy to correct

Frame-based Systems: Pros and Cons

- Advantages
 - Relatively flexible input – multiple inputs, orders
 - Well-suited to complex information access (air)
 - Supports different types of initiative
- Disadvantages
 - Ill-suited to more complex problem-solving
 - Form-filling applications

Richer Dialog Management

- Alternative Dialog Management approaches
 - More flexible interaction, motivated by human-human

Richer Dialog Management

- Alternative Dialog Management approaches
 - More flexible interaction, motivated by human-human
- Information State
 - General interpretation of speech in terms of dialog acts
 - Similar to “speech acts”, e.g. statement, wh-q, yn-q, check,..
 - Model of knowledge, belief state of current dialog

Richer Dialog Management

- Alternative Dialog Management approaches
 - More flexible interaction, motivated by human-human
 - Information State
 - General interpretation of speech in terms of dialog acts
 - Similar to “speech acts”, e.g. statement, wh-q, yn-q, check,..
 - Model of knowledge, belief state of current dialog
 - Statistical dialog management
 - Builds on reinforcement learning approaches (planning)
 - Aims to automatically learn best sequence of actions
 - Models uncertainty in system understanding of user

Designing Dialog

- Apply user-centered design

Designing Dialog

- Apply user-centered design
 - Study user and task: How?

Designing Dialog

- Apply user-centered design
 - Study user and task: How?
 - Interview potential users, record human-human tasks
 - Study how the user interacts with the system

Designing Dialog

- Apply user-centered design
 - Study user and task: How?
 - Interview potential users, recorded human-human tasks
 - Study how the user interacts with the system
 - But it's not built yet....

Designing Dialog

- Apply user-centered design
 - Study user and task: How?
 - Interview potential users, recorded human-human tasks
 - Study how the user interacts with the system
 - But it's not built yet....
 - Wizard-of-Oz systems: Simulations
 - User thinks they're interacting with a system, but it's driven by a human
 - Prototypes

Designing Dialog

- Apply user-centered design
 - Study user and task: How?
 - Interview potential users, recorded human-human tasks
 - Study how the user interacts with the system
 - But it's not built yet....
 - Wizard-of-Oz systems: Simulations
 - User thinks they're interacting with a system, but it's driven by a human
 - Prototypes
 - Iterative redesign:
 - Test system: see how users really react, what problems occur, correct, repeat

SDS Evaluation

- Goal: Determine overall user satisfaction
 - Highlight systems problems; help tune

SDS Evaluation

- Goal: Determine overall user satisfaction
 - Highlight systems problems; help tune
- Classically: Conduct user surveys

SDS Evaluation

- Goal: Determine overall user satisfaction
 - Highlight systems problems; help tune
- Classically: Conduct user surveys

TTS Performance	Was the system easy to understand ?
ASR Performance	Did the system understand what you said?
Task Ease	Was it easy to find the message/flight/train you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
User Expertise	Did you know what you could say at each point?
System Response	How often was the system sluggish and slow to reply to you?
Expected Behavior	Did the system work the way you expected it to?
Future Use	Do you think you'd use the system in the future?

Figure 24.14 User satisfaction survey, adapted from Walker et al. (2001).

SDS Evaluation

- User evaluation issues:

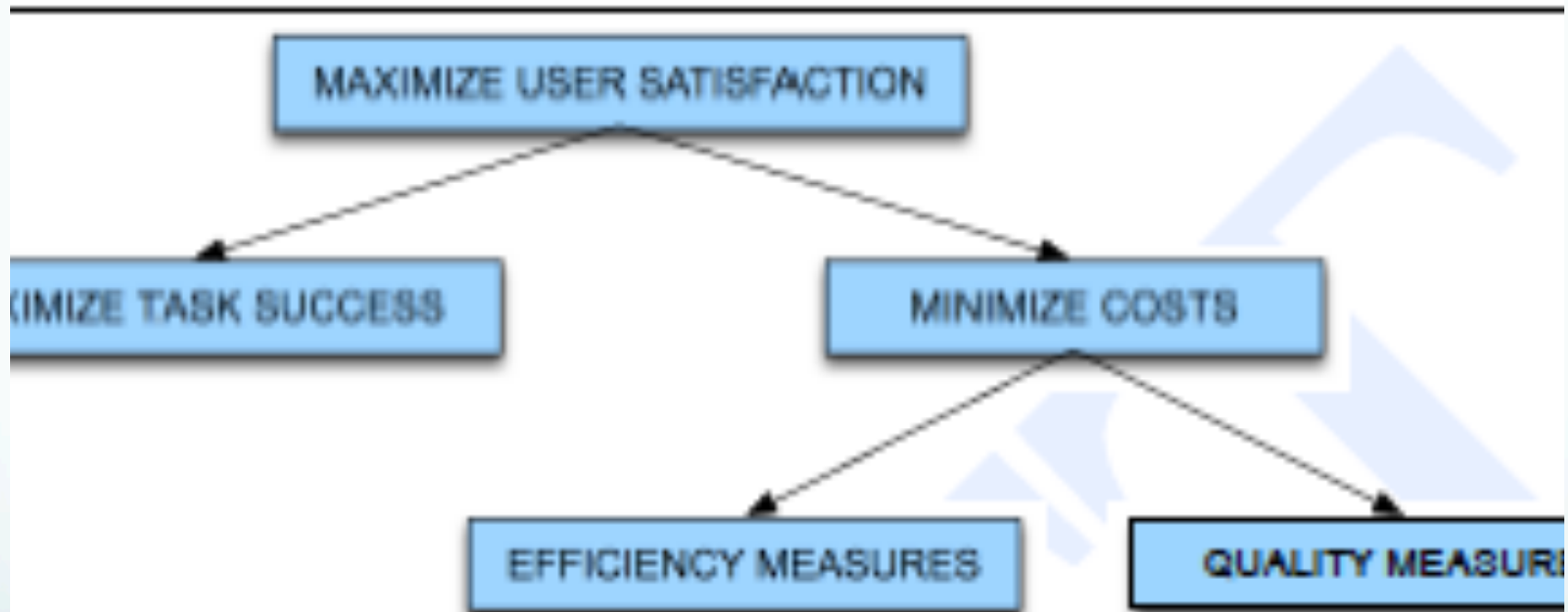
SDS Evaluation

- User evaluation issues:
 - Expensive; often unrealistic; hard to get real user to do
- Create model correlated with human satisfaction
- Criteria:

SDS Evaluation

- User evaluation issues:
 - Expensive; often unrealistic; hard to get real user to do
- Create model correlated with human satisfaction
- Criteria:
 - Maximize task success
 - Measure task completion: % subgoals; Kappa of frame values
 - Minimize task costs
 - Efficiency costs: time elapsed; # turns; # error correction turns
 - Quality costs: # rejections; # barge-in; concept error rate

PARADISE Model



PARADISE's structure of objectives for spoken dialogue performance (1997).

PARADISE Model

- Compute user satisfaction with questionnaires
- Extract task success and costs measures from corresponding dialogs
 - Automatically or manually
- Perform multiple regression:
 - Assign weights to all factors of contribution to Usat
 - Task success, Concept accuracy key
- Allows prediction of accuracy on new dialog

Summary

- Spoken Dialogue Systems:
 - Build on existing text-based NLP techniques, but
 - Incorporate dialogue specific factors:
 - Turn-taking, grounding, dialogue acts
 - Affected by computational and modal constraints
 - Recognition errors, processing speed, etc.
 - Speech transience, slowness
 - Becoming more widespread and more flexible

Semantic Grammars

- Alternatives:
 - Full parser with semantic attachments
 - Domain-specific analyzers
- CFG in which the LHS of rules is a semantic category:
 - LIST -> show me | I want | can I see|...
 - DEPARTTIME -> (after|around|before) HOUR| morning | afternoon | evening
 - HOUR -> one|two|three...|twelve (am|pm)
 - FLIGHTS -> (a) flight|flights
 - ORIGIN -> from CITY
 - DESTINATION -> to CITY
 - CITY -> Boston | San Francisco | Denver | Washington

Result

- **SHOW FLIGHT ORIGIN DEST DEP_DATE DEP_TIME**
- *Show me flights from Boston to SFO on Tuesday morning*

Verbmobil DA

- 18 high level tags

Tag	Example
THANK	<i>Thanks</i>
GREET	<i>Hello Dan</i>
INTRODUCE	<i>It's me again</i>
BYE	<i>Allright bye</i>
REQUEST-COMMENT	<i>How does that look?</i>
SUGGEST	<i>from thirteenth through seventeenth June</i>
REJECT	<i>No Friday I'm booked all day</i>
ACCEPT	<i>Saturday sounds fine,</i>
REQUEST-SUGGEST	<i>What is a good day of the week for you?</i>
INIT	<i>I wanted to make an appointment with you</i>
GIVE_REASON	<i>Because I have meetings all afternoon</i>
FEEDBACK	<i>Okay</i>
DELIBERATE	<i>Let me check my calendar here</i>
CONFIRM	<i>Okay, that would be wonderful</i>
CLARIFY	<i>Okay, do you mean Tuesday the 23rd?</i>
DIGRESS	<i>[we could meet for lunch] and eat lots of ice cream</i>
MOTIVATE	<i>We should go to visit our subsidiary in Munich</i>
GARBAGE	<i>Oops, I-</i>

Figure 24.17 The 18 high-level dialogue acts used in Verbmobil-1, abstracted over a total of 43 more specific dialogue acts. Examples are from Jekat et al. (1995).

Dialogue Act Ambiguity

- Indirect speech acts

A	OPEN-OPTION	I was wanting to make some arrangements for a trip that I'm going to be taking uh to LA uh beginning of the week after next.
B	HOLD	OK uh let me pull up your profile and I'll be right with you here. [pause]
B	CHECK	And you said you wanted to travel next week?
A	ACCEPT	Uh yes.

Performance Functions for 3 Systems

- ELVIS User Sat.= $.21 * \text{COMP} + .47 * \text{MRS} - .15 * \text{ET}$
 - TOOT User Sat.= $.35 * \text{COMP} + .45 * \text{MRS} - .14 * \text{ET}$
 - ANNIE User Sat.= $.33 * \text{COMP} + .25 * \text{MRS} + .33 * \text{Help}$
-
- COMP: User perception of task completion (task success)
 - MRS: Mean (concept) recognition accuracy (cost)
 - ET: Elapsed time (cost)
 - Help: Help requests (cost)

VoiceXML

- W3C standard for simple frame-based dialogues
 - Fairly common in commercial settings
- Construct forms, menus
 - Forms get field data
 - Using attached prompts
 - With specified grammar (CFG)
 - With simple semantic attachments

Simple VoiceXML Example

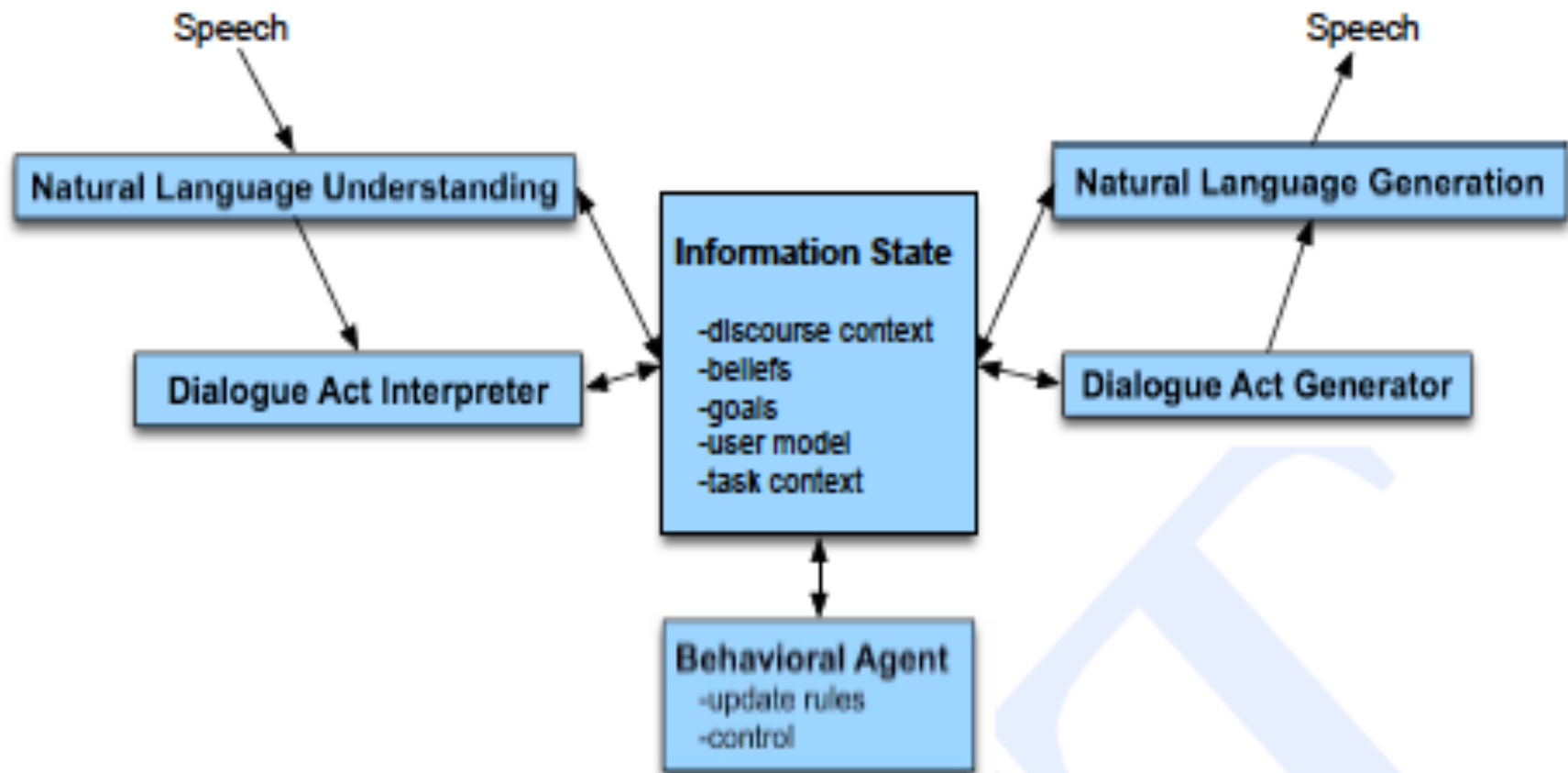
```
<form>
  <field name="transporttype">
    <prompt>
      Please choose airline, hotel, or rental car.
    </prompt>
    <grammar type="application/x-nuance-gsl">
      [airline hotel "rental car"]
    </grammar>
  </field>
  <block>
    <prompt>
      You have chosen <value expr="transporttype">.
    </prompt>
  </block>
</form>
```

Information State Dialogue Management

- Problem: Not every task is equivalent to form-filling
- Real tasks require:
 - Proposing ideas, refinement, rejection, grounding, clarification, elaboration, etc
- Information state models include:
 - Information state
 - Dialogue act interpreter
 - Dialogue act generator
 - Update rules
 - Control structure

Information State Architecture

- Simple ideas, complex execution



Dialogue Acts

- Extension of speech acts
 - Adds structure related to conversational phenomena
 - Grounding, adjacency pairs, etc
- Many proposed tagsets

Dialogue Act Interpretation

- Automatically tag utterances in dialogue
- Some simple cases:
 - **YES-NO-Q:** Will breakfast be served on USAir 1557?
 - I don't care about lunch.
 - Show be flights from L.A. to Orlando

Dialogue Act Interpretation

- Automatically tag utterances in dialogue
- Some simple cases:
 - **YES-NO-Q:** Will breakfast be served on USAir 1557?
 - **Statement:** I don't care about lunch.
 - Show be flights from L.A. to Orlando

Dialogue Act Interpretation

- Automatically tag utterances in dialogue
- Some simple cases:
 - **YES-NO-Q:** Will breakfast be served on USAir 1557?
 - **Statement:** I don't care about lunch.
 - **Command:** Show be flights from L.A. to Orlando
- Is it always that easy?
 - Can you give me the flights from Atlanta to Boston?
 - Yeah.

Dialogue Act Interpretation

- Automatically tag utterances in dialogue
- Some simple cases:
 - **YES-NO-Q:** Will breakfast be served on USAir 1557?
 - **Statement:** I don't care about lunch.
 - **Command:** Show me flights from L.A. to Orlando
- Is it always that easy?
 - Can you give me the flights from Atlanta to Boston?
 - Yeah.
 - Depends on context: Y/N answer; agreement; back-channel

Detecting Correction Acts

- Miscommunication is common in SDS
 - Utterances after errors misrecognized >2x as often
 - Frequently repetition or paraphrase of original input

Detecting Correction Acts

- Miscommunication is common in SDS
 - Utterances after errors misrecognized >2x as often
 - Frequently repetition or paraphrase of original input
- Systems need to detect, correct

Detecting Correction Acts

- Miscommunication is common in SDS
 - Utterances after errors misrecognized >2x as often
 - Frequently repetition or paraphrase of original input
- Systems need to detect, correct
- Corrections are spoken differently:
 - Hyperarticulated (slower, clearer) -> lower ASR conf.
 - Some word cues: 'No', 'I meant', swearing..

Detecting Correction Acts

- Miscommunication is common in SDS
 - Utterances after errors misrecognized >2x as often
 - Frequently repetition or paraphrase of original input
- Systems need to detect, correct
- Corrections are spoken differently:
 - Hyperarticulated (slower, clearer) -> lower ASR conf.
 - Some word cues: 'No', 'I meant', swearing..
- Can train classifiers to recognize with good acc.