COMP349
Spoken Language Dialogue Systems
Dialog Design

Rolf Schwitter

schwitt@ics.mq.edu.au
Today’s Program

• Software Development Life Cycle
• The Role of Scenarios
• Building a Conceptual Model
• Goals of Dialog Design
• Dialog Design Complexity
• Form Filling Paradigm
• Steps in Dialog Design
• Wizard of Oz Simulation
• Fill Logic for Error Correction
Software Development Life Cycle

- Software development life cycle of speech applications is similar to the life cycle of traditional applications:
  - feasibility study (=> determine project goals)
  - requirements analysis (=> analyse end-user information need)
  - design (=> describe desired features in detail)
  - implementation (=> write the real code)
  - integration and testing (=> bring the pieces together)
  - maintenance (=> goes on forever)
Software Development Life Cycle

• However, there are some specific tasks to speech applications:
  – dialog design (= call flow layout)
  – specifying the persona
  – coding the application (in VoiceXML)
    – speech grammars
    – error handlers
    – speech synthesis markup.
The Role of Scenarios

- A scenario tells a well-defined story.
- Basically, it is an outline of a hypothesized chain of events.
- A scenario encompasses a context of use of a planned system.
- Example scenario:

  Melinda is a nineteen year old teenager who owns a mobile phone. She is very keen in getting the latest ringtones for her mobile phone weekly. She buys these ringtones via CallRingCharts, a new telephony-based speech service. Melinda calls in this service, navigates through the ringtone list, listens to a number of ringtones, selects a ringtone and buys it with her credit card. The selected ringtone is automatically downloaded to Melinda’s mobile phone and the installation is then tested via CallRingCharts’ server.
Building a Conceptual Model

• The conceptual model describes
  – which objects the caller may manipulate and
  – which operations or commands s/he would use.

• How to build a conceptual model?

  Designer: What are the things you would like to hear?
  Potential caller: Ringtones.
  Designer: What words would you use to browse the ringtone list?
  Potential caller: I would say 'Next' and 'Prior'.
  Designer: What words would you use to purchase a tune.
  Potential caller: I would say 'Buy it'.
Conceptual Model

- The conceptual objects for a call ring application are:
  - Ringtone Application
  - Ringtone List
  - Ringtone
A Simple Call Flow Diagram (without Prompts)

Start

Navigate Ringtone List

Buy

Download

Test

End
State Transition Graph

- Navigate Ringtone List
- Buy
- Ringtone
  - Next
  - Prior
  - Listen
  - Buy
- Verify Tune
  - Yes
  - No
- Enter Credit Card #: Credit Card Number
- Exit
The Goals of Dialog Design

• Our primary goals:
  – to construct systems in such a way that callers get their work done effectively and efficiently
  – to achieve a high task completion rate.

• We should not be distracted by:
  – modelling world knowledge
  – modelling social experience
  – modelling the complexities of human discourse.
Dialog Complexity

• The complexity of a dialog depends on
  – the complexity of the underlying transactions
  – where the initiative lies in the dialog.

• There exist different dialog models:
  – finite-state dialog model (mainly machine-directed)
  – information-state model (mixed-initiative).
The Form Filling Paradigm

• Required caller knowledge:
  – caller must possess data
  – no knowledge of task or interface is required.
• Caller behaviour:
  – responding, answering questions.
• Initiative:
  – machine initiated.
• Easy to learn and to use.
• Little or no advantage to becoming an expert.
Example 1: Credit Card Details

- We need to collect
  - the caller’s credit card details and
  - delivery preference.
- Assume the caller’s purchasing has been handled independently.
Example 1: Credit Card Details

Computer: What kind of credit card will you use? [Beep]
Caller: Mastercard.
Computer: Card number? [Beep]
Caller: 1155-5511-1155-5511
Computer: Expiration date? [Beep]
Caller: Zero six zero eight.
Computer: Normal or special delivery? [Beep]
Caller: Normal.
Computer: Please confirm the order ... 〈machine reads back info〉
Example 1: Credit Card Details

- What kind of credit card will you use?
  - recognised
  - not recognised

- Card number?
  - recognised
  - not recognised

- Expiration date?
  - recognised
  - not recognised

- Normal or special delivery?
  - recognised
  - not recognised

- Please confirm your order is as follows ...
Basic Notational Conventions

• A call flow diagram shows the flow of the conversation.
• Different types of notations exist (depending on the granularity).
• We will use the following conventions for call flow diagrams:
  – explicit prompts (or abbreviations) label nodes
  – system actions appear on arcs.
• Distinguish between recognition states and non-recognition states.
• Error recovery:
  – you can assume each state has a re-prompt capability built in
  – there’s no need to include it in your call flow diagram.
Example 1: Credit Card Details

What kind of credit card will you use?
- STORE CARD TYPE

Card number?
- STORE CARD NUMBER

Expiration date?
- STORE EXPIRATION DATE

Normal or special delivery?
- STORE DELIVERY PREFERENCE

Please confirm your order is as follows ...
Example 2: The Nuance Travel Planner

Computer: Welcome to Travel Plan, a service of Nuance Communications.
          What city would you like to leave from?
Caller:  From San Francisco.
Computer: What city are you flying to?
Caller:  To San Diego.
Computer: Please say the destination city; for example, Atlanta Georgia.
Caller:  San Diego.
Computer: On what date are you travelling?
Caller:  Uh, on May the 28th.
Example 2: The Nuance Travel Planner

Computer: Around what time would you like to leave?
Caller: I’ve got to be in San Diego before 10 am.
Computer: Please say your approximate departure time.
           For example, 10 am.
Caller: Oh, probably 8 am or so?
Computer: Please say your approximate departure time.
           For example, 10 am.
Caller: Uh, 8 am.
Computer: You are travelling from San Francisco to San Diego on
           Tuesday May 28th at 8 am. Is this correct?
Caller: Yes, that’s fine.
Example 2: The Nuance Travel Planner

What city would you like to leave from?
STORE ORIGIN

What city are you flying to?
STORE DESTINATION

On what date are you travelling?
STORE DATE

Around when would you like to leave?
STORE TIME

You are travelling from ... to ... at ... on ... Is that correct?
Example 3: Pizza Ordering

Computer: Welcome to the CSLU Pizza Parlour
Computer: Would you like a small, medium, or large pizza?
Caller: A medium one.
Computer: What kind of toppings: cheese, hawaiin, pepperoni, or vegetarian?
Caller: Pepperoni, please.
Computer: Would you like a salad with that?
Caller: Oh, no.
Computer: So, you want to buy a large pizza with pepperoni, right?
Example 3: Pizza Ordering

Welcome to the CSLU Pizza Parlour.

Would you like a small, medium or large pizza?

What kind of topping: cheese, hawaiian, pepperoni or vegetarian?

Would you like a salad with that?

So you want a ..., right?

Okay, your order will be ready shortly.

Let's start again then.
Example 4: A More Complex Enquiry System

- Things get more interesting when we add conditional branching.
- Suppose we have to build a speech-enabled front end that is similar to the Qantas’ general DTMF enquiry system on 13 13 13.
Example 4: A More Complex Enquiry System

Computer: Welcome to Qantas Telephone Sales.
If you are a Frequent Flyer member, please press 1 now.
If you have an existing booking, press 2.
For new bookings, press 3.
For flight information, press 4.
If you want to hear these options again, press 0.

Caller: [4]

For international flights, press 2 ...
Example 4: A More Complex Enquiry System

- **Welcome Message**
- **Are you a Frequent Flyer member?**
  - Yes: Please say your Frequent Flyer membership number.
  - No: Please choose flight information or reservations.
- **COLLECT FF NUMBER**
- **Reservations**
  - Do you already have a reservation or do you wish to make a new reservation?
- **Information**
  - Are you enquiring about a domestic flight or an international flight?
Steps in Dialog Design

1. Make sure you understand what you are trying to achieve (use scenarios and build a conceptual model).
2. See if you can decompose the task into smaller meaningful subtasks.
3. Identify the information tokens you need for each task or subtask.
4. Decide how you will obtain this information from the caller.
5. Sketch a call flow diagram with appropriate prompts that captures this information.
6. Test your call flow diagram in a Wizard of Oz simulation.
7. Revise your call flow diagram and repeat Step 6 …
Wizard of Oz Simulations

• A human experimenter (the Wizard) simulates an automated system.
• The dialog specifications (for example, call flow diagrams with associated prompts) are spread out in front of the experimenter.
• The experimenter reads the appropriate prompts from the specs, waits for a response from the subject (or no response), checks the specs on how to proceed, and then speaks the next prompt.
• Very effective in uncovering problems with logic, navigation, awkward sequences of prompts, omissions, and so on.
Subdialogs

- Subdialogs are like subroutines in programming languages.
- They provide a way of managing complexity.
- They provide a way of decomposing a problem into subproblems.
- They provide a way of reusing common code.
Example 5: A More Complex Enquiry System

〈Welcome Message〉

Are you a Frequent Flyer member?

Yes

Please say your Frequent Flyer membership number.

No

Please choose flight information or reservations.

Reservations

Information

Reservations Subdialog

Information Subdialog

Thanks for calling. Goodbye.

Do you want to make another enquiry?

Yes

No
Some Tips

• A typical use case should be painless to use.
• Assume the caller does not know the active vocabulary.
• Guide the caller towards responses that
  – maximize clarity and
  – unambiguousness.
• Allow for the caller
  – not knowing the answer to a question or
  – not understanding a question.
Some Tips

- Supply confirmation messages frequently, especially when the cost or likelihood of a recognition error is high.
- Assume errors are the fault of the recognizer, not the caller.
- Design graceful recovery when the recognizer makes an error.
- Assume a frequent caller will have a rapid learning curve.
Universal Help

• Allow the caller to access help at any state.
• Use a help subdialog that provides information on using the system.
• If your help is universal there’s no need to show it explicitly in your call flow diagram.
Prompts

• Guide callers toward natural ‘in vocabulary’ responses.
  Computer 1: Welcome to ABC Bank. What would you like to do?
  versus
  Computer 2: Welcome to ABC Bank. You can check an account balance, transfer funds, or pay a bill. What would you like to do?

• Do not give too many options at once.

• Keep syntax and semantics consistent across all prompts in the system.

• Keep prompts brief to encourage the caller to be brief.

• Avoid prompts that are too similar.
Fill Logic

- Observation: So far, it was not possible to correct a specific item.
- We need more control over the way fields are filled than is provided by a standard form-filling dialog.
- A piece of code is needed that maintains an agenda of what information has been provided so far, and what remains to be provided.
- Fill logic can be used to provide more natural interaction.
Example: Pizza Ordering

Welcome to the CSLU Pizza Parlour.

Would you like a small, medium or large pizza?

STORE SIZE

What kind of topping: cheese, hawaiian, pepperoni or vegetarian?

STORE TOPPING

Would you like a salad with that?

STORE SALAD

So you want a ..., right?

Yes

Okay, your order will be ready shortly.

No

Ok. What did I get wrong? Please choose size, topping or salad.
Fill Logic: Agenda

• Maintain an ordered agenda of information tokens to be acquired:

<table>
<thead>
<tr>
<th>Item</th>
<th>Acquired?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>no</td>
</tr>
<tr>
<td>Topping</td>
<td>no</td>
</tr>
<tr>
<td>Salad</td>
<td>no</td>
</tr>
</tbody>
</table>
Fill Logic: Algorithm

- Algorithm:
  
  for each item in agenda
  
  if item not yet acquired
  
  then invoke subdialog to acquire it
  
  next item

- The effect of a negative confirmation is to reset the acquired status of the relevant item or items and then to restart this loop.
Welcome to the CSLU Pizza Parlour.

Fill Logic

Get Size  Get Topping  Get Salad

size  topping  salad

So you want a ..., right?

No

Ok. What did I get wrong? Please choose size, topping or salad.

Yes

Okay, your order will be ready shortly.
Take-Home Messages

- Good dialog design allows callers to get their work done efficiently.
- In a form filling dialog design model
  - user must possess data
  - no knowledge of task or interface is required.
- A call flow diagram shows the flow of the conversation.
- Fill logic can be used to provide more natural interaction.