Special Lecture (406)
Spoken Language Dialog Systems
Creating Speech Applications

Rolf Schwitter
{schwitt}@ics.mq.edu.au
Today’s Program

• Speech application development life cycle
• Performance metrics
• Preference metrics
• Choosing the technology
• Developing the application
• Testing
• Appendix: The W3C Speech Interface Framework
The Speech Industry

• VoiceXML platform providers
  – sell VoiceXML browsers, development kits, server hardware (VoiceGenie, IBM, Philips, Motorola)
• VoiceXML network hosting services
  – host a voice portal for a fee (Net2Phone, Qwest, Sprint)
• VoiceXML hosting and application
  – construct and run a voice portal (BeVocal, Tellme, Voxeo)
• Application development (SpeechWorks, Nuance, NetByTel)
Development Life Cycle

- The development life cycle of speech applications consists of the following stages:
  - investigation: identify application
  - design: specify business model, application, and technology
  - development: develop application
  - testing: test application
  - sustaining: deploy application
The Development Team

- There are many roles required to build, test, and deploy a speech-enabled application:
  - caller
  - dialog designer
  - VoiceXML programmer
  - voice talent
  - grammar writer
  - manager
  - etc.
Identify the Application

- Many designers perform four steps to identify the application:
  - conduct ethnographic studies
  - identify candidate applications
  - conduct focus groups
  - select application to implement
Identify the Application

- Ethnographers
  - study small segments of society
  - attempt to understand what people do and how they do it.
- They investigate
  - who are the prospective callers (caller profile)
  - what do the prospective callers (activity log)
  - where do they perform activities (physical environment models)
  - why do callers what they do (context model).
Identify the Application

• Let us assume that ethnographic results suggest the following candidate speech applications for teenagers:
  – voice dialing
  – entertainment activities
  – teen idols
  – astrology information
  – popular music
  – call ring
Identify the Application

- Many developers start with focus groups.
- They discuss possible speech applications with prospective callers.
- For example: call ring application
  Callers listen to, select, download, and install a touchtone melody that is replayed by the mobile phone to announce an incoming call.
- Two critical factors leading to select an application are:
  - ROI (return on investment)
  - constraints regarding time and resources
Ringtones (HTML Solution)

Most Popular Ringtones (Last two weeks)

<table>
<thead>
<tr>
<th>Artist</th>
<th>Song</th>
<th>Play</th>
<th>Send</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lil Kim feat 50 Cent</td>
<td>Magic Stick</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 Cent</td>
<td>P.I.M.P.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jay-Z</td>
<td>Big Pimpin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ludacris</td>
<td>Move Bitch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 Cent</td>
<td>In Da Club</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Featured Artists

Check out ringtones from these hot artists!
Business Model

- A business model explains how the sponsors will earn a ROI.

<table>
<thead>
<tr>
<th>Business Model</th>
<th>Motivation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>Profit from sales of goods or services</td>
<td>Call ring sales</td>
</tr>
<tr>
<td>Advertising</td>
<td>Profit from advertising revenue</td>
<td>Teen idols</td>
</tr>
<tr>
<td>Subscription</td>
<td>Profit from pay per call and per time</td>
<td>Astrology, pop music</td>
</tr>
<tr>
<td>Self-help</td>
<td>Decrease expenses</td>
<td>Call centre automation</td>
</tr>
<tr>
<td>Customer relations</td>
<td>Gaining market share</td>
<td>Automated receptionist</td>
</tr>
</tbody>
</table>
Specify the Application

• To specify the application, developers engage prospective callers to define the application requirement specifications:
  – conceptual model
  – scenarios
  – performance metrics
  – preference metrics
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: Call ring tunes.
  Designer: What words would you use to browse call ring lists?
  Potential caller: I would say 'Next' and 'Prior'.
  Designer: What words would you use to purchase the tune.
  Potential caller: I would say 'Buy it'.

© Macquarie University 2004
Conceptual Model

• The conceptual objects for a call ring application are:
  – Call Ring Application
  – Call Ring List
  – Call Ring
Scenarios

- Scenarios describe how callers manipulate conceptual objects.
- Dialog designer and prospective caller identify possible scenarios.
- Useful technique is to role-play:
  - designer plays the role of the application
  - prospective caller plays the role of the caller.
- First, use the conceptual model, then hide the model.
- Scenarios suggest new objects and relationships in the conceptual model.
- Refine the conceptual model and repeat the role-playing exercise.
## Example Scenario

<table>
<thead>
<tr>
<th>Verbal Command</th>
<th>Resulting Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke call rings</td>
<td>Welcome to call ring application. Say ‘next’ or ‘prior’ to review call ring titles.</td>
</tr>
<tr>
<td>Next</td>
<td>Happy Birthday.</td>
</tr>
<tr>
<td>Next</td>
<td>Three Blind Mice.</td>
</tr>
<tr>
<td>Listen</td>
<td>Plays theme from ‘Three Blind Mice’.</td>
</tr>
<tr>
<td>Purchase</td>
<td>You want to purchase ‘Three Blind Mice’, is that correct?</td>
</tr>
<tr>
<td>Yes</td>
<td>The cost is one dollar. Please key in your credit card number.</td>
</tr>
</tbody>
</table>
# Example Scenario

<table>
<thead>
<tr>
<th>Verbal Command</th>
<th>Resulting Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>Hello, did you hear ‘Three Blind Mice’.</td>
</tr>
<tr>
<td>Yes</td>
<td>Thank you for using call rings. Please call back when you want to purchase another call ring tune. (Hangs up).</td>
</tr>
</tbody>
</table>
Performance Metrics

- Performance metrics measure the caller’s success or failure to perform a specific task.
- Performance metrics
  - should be objective
  - are generated by measurable tasks performed by a caller
  - are used to develop user tests for development phase.
- If the caller fails to achieve the specified success indicators, then the application is not ready for release.
Example: Performance Metrics

<table>
<thead>
<tr>
<th>Caller Task</th>
<th>Metrics</th>
<th>Success Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caller speaks command</td>
<td>Word error rate</td>
<td>Less than 3%</td>
</tr>
<tr>
<td>Caller understands a prompt</td>
<td>Caller performs an appropriate action after hearing the prompt</td>
<td>Greater than 97%</td>
</tr>
<tr>
<td>Caller navigates a list</td>
<td>Caller successfully selects the specified option</td>
<td>Greater than 95%</td>
</tr>
<tr>
<td>Caller purchases a ring tone</td>
<td>Caller successfully completes the purchase option</td>
<td>Greater than 93%</td>
</tr>
</tbody>
</table>
Preference Metrics

- Preference metrics measure the caller’s likes and dislikes.
- These metrics are subjective.
- They are measured by asking callers questions.
## Example: Preference Metrics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Success Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>On a scale of 1 to 10, rate the help facility.</td>
<td>Average caller score is greater than 7.</td>
</tr>
<tr>
<td>On a scale of 1 to 10, rate the understandability of the synthesized voice.</td>
<td>Average caller score is greater than 7.</td>
</tr>
<tr>
<td>Would you recommend using this voice portal to a friend?</td>
<td>Over 75% of caller respond by saying &quot;yes&quot;</td>
</tr>
<tr>
<td>What would you be willing to pay to use this voice-enabled application?</td>
<td>Over 75% of callers indicate that they are willing to pay $1.00 or more per use.</td>
</tr>
</tbody>
</table>
Choose the Right Technology

• What technology to use in developing the ring call application?
• The three candidate technologies are:
  – using DTMF as input
  – using voice as input
  – using small displays on newer mobile phones.
• Determine if the technology
  – is the best available
  – works correctly
  – can be integrated onto the planned platform(s).
Which Speech Recognition Engine?

• Collect the words and phrases uttered by the callers in the scenarios.
• Ask prospective callers to speak these words into the recogniser.
• Determine
  – number of spoken words that were not recognised
  – number of words recognized that were not spoken
  – number of words substituted for a spoken word.
• These three measurements define the word error rate.
• The smaller the word error rate, the better is the recogniser.
Which Speech Recognition Engine?

• In addition to word error rate, other performance factors are:
  – required storage space
  – footprint (size of executing engine and data within memory)
  – latency (time from speaking to time the text is available)
  – cost and licensing requirements.
Which Speech Synthesis Engine?

- Collect the words and phrases spoken to the callers in the scenarios.
- Ask prospective callers to listen and write down each expression.
- Perform the same calculations as for the speech recogniser tests.
- Major factor in selecting a synthesiser are:
  - word error rate
  - caller preference
  - required storage space
  - footprint
  - latency
Develop the Application

• Only after all design activities are completed, a go/no go decision should be made about entering the development stage.

• Building the application consists of:
  – specifying the application’s persona
  – specifying the dialog structure
  – writing the dialog VoiceXML document
  – refining the speech application.
Specify the Application’s Persona

• The application’s persona describes its "sound and character".
• It is important to specify the application’s persona early.
• The persona should reflect the needs and desires of the audience.
• For example, the call ring application should have a persona that appeals to teenagers.
Specify the Dialog Structure

• Dialog designer specifies the call flow of the application.
• A call flow is a high level flow diagram of the conversation.
• A state transition graph represents
  – a (system-driven) dialog with
  – nodes containing prompts to the caller and
  – arcs indicating caller responses.
• Validate the dialog structure by a Wizard of Oz test.
Call Flow Diagram

Start

Navigate Call Ring

Buy

Download

Test

End
Wizard of Oz Test

- To validate the dialog structure, conduct Wizard of Oz experiments.
- The developer pretends to be speech application by listening and speaking to the caller while constraining responses to be consistent with the conceptual model.
- The conversation is recorded and later analysed.
- If necessary the conceptual model needs to be modified.
Write the Dialog Document

• Write the dialog document in VoiceXML:
  – conceptual model provides names for forms, fields, and menus
  – scenarios are sources for basic prompts and responses.
• Write the grammars that specify the word and phrases.
• Specify event handlers for each field and menu.
Refine the Speech Application

• Test, test, test, and check the log file for errors.
• Modify the prompts.
• Refine the grammars.
• Add any new words that are not in the lexicon.
• Caution:
  If a voice file needs to be changed, the original voice talent might not be available anymore.
Test the Application

• Several types of testing are necessary before deploying the application:
  – usability test
  – qualification test
  – stress test
  – field test
Usability Test

• The objective of usability testing is to refine the user interface.
• Common speech recognition errors include:
  – out of grammar utterances
  – substitution errors
  – truncation errors.
• Do the following:
  – tune the phonetic representation
  – tune the grammar
  – tune parameters (confidence level, end pointing, noise filtering).
Qualification Test

• The objective of a qualification test is to determine if the application is ready for deployment.
• Qualification tests measure the performance against a benchmark.
• A benchmark consists of
  – typical scenarios
  – performance metrics
  – preference metrics.
Stress Test

• The objective of a stress test is to determine
  – how robust the telephony-enabled application is
  – how it will perform while several callers try to use it.
• Stress tests can be performed by a call generator.
• Additional background noise may be added to the recordings.
• Two distinct components in a VoiceXML application to test:
  – voice infrastructure
  – web infrastructure.
Field Test

• The objective of a field test is to validate the application under real-world environments.
• This is done with a small number of users.
• Developers look for unexpected performance problems.
• User preferences are collected and evaluated.
• User problems are logged and analysed.
Deploy and Monitor the Application

- "Build it and they will come" does not work.
- Publicize and advertise the new application.
- Emphasise the benefits but do not oversell.
- Ramp up the volume of callers gradually.
- Monitor the application:
  - use log files created by the speech application
  - use log files created by call centre help personnel.
Take-Home Message

• The life cycle for speech applications has many components that are similar to traditional software applications.

• However there are differences:
  – specifying the application’s persona
  – coding the application in VoiceXML
  – prompts and grammars (and event handlers).

• Iterative approach is useful.

• Prospective callers must be heavily involved during each stage.
Appendix: The W3C Speech Interface Framework

- The Voice Browser Working Group (VBWG) develops specifications for markup languages for speech-enabled Web applications.
- This framework of specification languages for speech processing systems is called W3C Speech Interface Framework.
Components of the Speech Interface Framework
A Speech Application Fragment

```xml
<form id="travel">
  <field name="destination">
    <prompt>
      Destination. Do you want to fly to
      <emphasis>New York</emphasis> or
      <emphasis>Washington</emphasis>
    </prompt>
    <grammar mod="voice" xml:lang="en-US" version="1.0"
      root="destination_city">
      <rule id="destination_city">
        <one-of>
          <item>New York</item>
          <item>Big Apple <tag>‘New York’</tag></item>
          <item>Washington</item>
          <item>The Capital <tag>‘Washington’</tag></item>
        </one-of>
      </rule>
    </grammar>
  </field>
</form>
```
The Voice Extensible Markup Language (VoiceXML 2.0) - as we know -

- is a dialog markup language for specifying conversational dialogs
- is based on VoiceXML 1.0 which was contributed to the W3C by members of the VoiceXML forum.

Check

- W3C: http://www.w3.org/TR/voicexml20/
- VoiceXML forum: http://www.voicexml.org/index.html
SSML

• The Speech Synthesis Markup Language (SSML)
  – describes how text is presented as audio to the user.
• Developers use SSML to specify speech formatting, including
  – word emphasis,
  – prosody,
  – speed,
  – volume,
  – pitch and other voice characteristics.
• Check: http://www.w3.org/TR/speech-synthesis/
The Speech Recognition Grammar Specification (SRGS)
• specifies the words and phrases
  • which a user may speak in response to a prompt.
• The syntax of the grammar format can be presented in two forms:
  • as Augmented BNF Form and
  • as XML Form.
• The specification makes the two representations mappable to allow automatic transformations between the two forms.
• Check: http://www.w3.org/TR/speech-grammar/
Example

ABNF

$destination =
  to boston {"boston"} |
  to new york {"new-york"}

XML

<rule id = "destination">
  <one-of>
    <item>
      to boston
      <tag> "boston" </tag>
    </item>
    <item>
      to new york
      <tag> "new-york" </tag>
    </item>
  </one-of>
</rule>
Semantic Interpretation

• The Semantic Interpretation Language (SI)
  – extracts words and phrases that have been recognized by the speech recognition engine and
  – translates them to semantically meaningful tokens for processing by the speech application.

• For example, "Big Apple" is translated to "New York" and "The Capital" is translated to "Washington".

• Check: http://www.w3.org/TR/semantic-interpretation/
• The Call Control Extensible Markup Language (CCXML)
  – is used for controlling how phone calls are placed, answered, transfered, conferenced.

• Traditionally, call control has required interaction with the telephony API's which often change from one platform to another.

• Basically, CCXML provides support for VoiceXML to move calls around and connect them to dialog resources.

• For example, being able to conditionally answer a call is one of the features that CCXML brings to VoiceXML applications.

• Check: http://www.w3.org/TR/2004/WD-ccxml-20040430/


```xml
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0">
    <eventhandler>
        <!-- Lets handle the incoming call -->
        <transition event="connection.CONNECTION_ALERTING" name="evt">
            <log expr="'The caller ID is ' + evt.callerid + '.'"/>
            <if cond="evt.callerid == '8315551234'">
                <accept/>
            </if>
            <else/>
                <reject/>
            </if>
        </transition>
        <!-- Lets handle the call being answered -->
        <transition event="connection.CONNECTION_CONNECTED">
        </transition>
    </eventhandler>
</ccxml>
```
EMMA

- Extensible Multimodal Markup Annotation (EMMA)\(^1\)
  - intended use is to represent the semantics
  - for information entered via various input modalities and
  - the resulting integrated information.
- Each of the modality-specific recognizers/interpreters convert the user supplied information into an EMMA representation.
- The EMMA representation is then processed by a dialog manager.

\(^1\) The W3C Multimodal Interaction working group specifies EMMA (http://www.w3.org/TR/emma/).
Figure 1: W3C Multimodal Framework
Example

• The speech utterance "Zoom in here" is represented as

  <emma:interpretation>
    <command>
      <zoom><location/></zoom>
    </command>
  </emma:interpretation>
Example

- The area circled by a pen is represented as a series of points:

```xml
<emma:interpretation>
  <area>
    (200,200), (200,400), (400,400), (400,200)
  </area>
</emma:interpretation>
```
Example

• The integration of information from both the speech and pen modalities is represented as:

```xml
<emma:interpretation>
  <command>
    <zoom>
      <area>
        (200,200), (200,400), (400,400), (400,200)
      </area>
    </zoom>
  </command>
</emma:interpretation>
```