Today’s Program

• All about Assignment 1
• Introduction
• Example Dialog
• Dialog Design
• Databases
• Documentation
• Marking Schema
• Submission
Introduction

• Your task is to build an automated real estate agent in VoiceXML.
• The caller should be able to find properties such as
  – houses, townhouses, and units
  that are
  – for sale and for rent
in three suburbs of Sydney.
• The speech-enabled agent should suggest suitable properties.
Introduction

• Additionally, the caller should be able to ask
  – for suburb-specific background information
    about a property.
• This might be background information about:
  – train station, post office, chemist, park, restaurant, school or doctor.
Example Dialog

Computer: Welcome to AMAZING HOME's automated real estate agent. This speech-enabled service will help you to find a suitable property by asking you a number of questions. If you are not sure about how to answer a question, simply say 'help'.

© Macquarie University 2007
Example Dialog

Computer: Would you like to buy or rent a property?
Caller: I would like to rent a property.
Computer: Okay.
   In which suburb would you like to rent a property?
   Please name the suburb or indicate the suburb's postcode.
Caller: Marsfield.
Example Dialog

Computer: Would you like to rent a house, a townhouse or a unit in Marsfield?
Caller: I would like to rent a townhouse.
Computer: Would you like to rent a 2 or 3 bedroom townhouse?
Caller: A 2 bedroom one.
Example Dialog

Computer: What price range are you looking for: 250 to 300 dollars, 300 to 350 dollars or above 350 dollars?

Caller: Between 300 and 350 dollars.

Computer: I have a 2 bedroom townhouse in 29/8 Taranto Road for $315 per week. Would you like more information about this townhouse?

Caller: Oh, yes, please.
Example Dialog

**Computer:** This modern 2 bedroom townhouse has a separate lounge and dining, built-ins, internal laundry, courtyard, ensuite, a single lock up garage, and is situated near to Macquarie University and shops. The townhouse is available now. Would you like to hear this information again?

**Caller:** No, thanks.
Example Dialog

Computer: Would you like to inspect the townhouse in 29/8 Taranto Road?
Caller: Yes.
Computer: The townhouse is open for inspection next Saturday, between 11:15 and 11:40.
Example Dialog

Computer: Would you like some background information about nearby facilities?
Caller: Yes, please.
Computer: About which facility would you like more information: Train Station, Post Office, Chemist, Park, Restaurant, School or Doctor?
Caller: Train Station.
Computer: The nearest train station is in Epping.
Example Dialog

Computer: Would you like more background information?
Caller: No, thank you.
Computer: Okay, thank you for using AMAZING HOME's speech-enabled service.
Dialog Design

- You can construct this application using a form-filling paradigm.
- However, this is not a very satisfactory solution.
- You might want to allow for some degree of “mixed-initiative”.
- For example:
  
  **Computer:** About which **facilities** would you like more information: Train Station, Post Office, Chemist, Park, Restaurant, School, Doctor?

  **Caller:** Train Station and Restaurant.
Dialog Design

• How far you take this approach is up to you.
• Whatever strategy you chose, make sure that
  – the user can ask any time for help
  – the system recovers gracefully from errors.
Databases

• You will have to build two simple databases:
  – a property database
  – a database with information about “nearby” facilities.
Property Database

• The property database should contain entries about **buyable and rentable** properties.

• Both types of properties should contain
  – at least two different entries for houses, townhouses, and units that are available in **three** different suburbs.

• These are at least 36 entries in total:
  – 12 houses, 12 townhouses, and 12 units.
Property Database

• You can build up your property database as a simple CSV file.
• Each record of the file should have the following schema:
  Suburb, PostCode, BuyOrRent, PropertyType, NumberOfBedrooms, Price, Road, PropertyDetails, InspectionDay, InspectionDate, StartInspectionTime, EndInspectionTime
Example: Property Database

• Here is a record with (more or less) realistic data:
  Marsfield, 2122, rent, townhouse, 2, 315, 29/8 Taranto Road, This modern 2 bedroom townhouse has a separate lounge and dining; built-ins; internal laundry; courtyard; ensuite; a single lock up garage; and is situated near to Macquarie University and shops. The townhouse is available now., Saturday, 25:08:2007, 11:15, 11:40

EPPING $390 p.w. NEW!

Huge 3 bedroom first floor
1/14-16 ESSEX STREET
Huge 3 bedroom first flr. apartment in well maintained garden block. 3-ina, large light living room & cnr. balcony, sunny eat-in kitchen, int.kdry, LGU. Area of unit 136.3sqm, garage 24.4sqm. Short walk to train & bus...

MORE INFO »

EPPING $25,000PA

"PROFESSIONAL PERSON REQUIRED"
15/74 RAWSON ST
Commercial/Retail space available in new genesis building. Ideally suited for a professional person. 57sqm. Freshly painted, suspended ceiling with downlight. Outgoings paid by landlord. Centrally located...

MORE INFO »

EPPING $450 p.w. NEW!

Executive style regency apartment!
402/36 VICTORIA STREET
Two bedroom apartment on the 4th floor of a modern apartment block, located on the quiet end of the street [no through road], adjoining Baronia Park. Bathroom, plus generous ensuite to spacious main bedroom...

MORE INFO »
Facilities Database

• This database should contain information about nearby facilities.
• This means you need a least three records in your database.
• For each suburb one record:
  Suburb, TrainStation, PostOffice, Chemist, Park, Restaurant, School, Doctor
• You can populate this database with information from realestate.com or with more specific information of your choice.
<table>
<thead>
<tr>
<th>Facility</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train Station</td>
<td>2</td>
</tr>
<tr>
<td>Hospital</td>
<td>1</td>
</tr>
<tr>
<td>Post Office</td>
<td>2</td>
</tr>
<tr>
<td>Cafe</td>
<td>6</td>
</tr>
<tr>
<td>Doctor</td>
<td>26</td>
</tr>
<tr>
<td>Restaurant</td>
<td>17</td>
</tr>
<tr>
<td>School/College</td>
<td>10</td>
</tr>
<tr>
<td>Fast Food</td>
<td>9</td>
</tr>
<tr>
<td>Chemist</td>
<td>7</td>
</tr>
<tr>
<td>Park</td>
<td>33</td>
</tr>
</tbody>
</table>
Preparing the Database

• You can translate the CSV file into a JavaScript array file.
• You can use the resulting JavaScript in your VoiceXML application.
• You can use the following CSV to JavaScript converter:
  http://www.creativyst.com/Prod/17/
CSV to JavaScript Array Converter

Input (CSV file):
- Marsfield, 2122, rent, townhouse, 2, 315, 29/8
- Taranto Road, This modern 2 bedroom townhouse has a separate lounge and dining; built-ins; internal laundry; courtyard; ensuite; a single lock up garage; and is situated near to Macquarie University and shops. The townhouse is available now. , Saturday, 25:08:2007, 11:15, 11:40

Output (JavaScript Arrays):
- Col0[0] = "Marsfield";
- Col1[0] = "2122";
- Col2[0] = "rent";
- Col3[0] = "townhouse";
- Col4[0] = "2";
- Col5[0] = "315";
- Col6[0] = "29/8 Taranto Road";
Documentation

• Your final submission should be supported by a written report (not more than 8 pages) that contains:
  • A top level description of the system's purpose and capabilities, drawing attention to any neat features you have implemented;
  • A call flow diagram showing the flow of the conversation;
  • A state description for each state of one use case (= one run of the system), including at least the following items:
Documentation

- A state description for each state of one use case in the dialog, including:
  - short description of the purpose of the state;
  - the wording of the prompt to be played in this state;
  - some examples of expected responses to the prompt;
  - a grammar description: for each state, you should provide a specification of the pieces of information that will be collected at that state, and some sample utterances that will provide this information;
  - any event handler that you feel is required at this state.
Marking Schema

- This assignment contributes 15% of the total marks for this unit.
- The 15 marks are distributed as follows:

<table>
<thead>
<tr>
<th>Aspect</th>
<th># of Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report</strong></td>
<td></td>
</tr>
<tr>
<td>Top-level description</td>
<td>1</td>
</tr>
<tr>
<td>Call flow diagram</td>
<td>1</td>
</tr>
<tr>
<td>State description</td>
<td>3</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td></td>
</tr>
<tr>
<td>System architecture</td>
<td>2</td>
</tr>
<tr>
<td>Quality of prompts</td>
<td>2</td>
</tr>
<tr>
<td>Quality of grammars</td>
<td>2</td>
</tr>
<tr>
<td>Usability</td>
<td>4</td>
</tr>
</tbody>
</table>
Submission

- Your final submission should consist of following three elements:
  - a written report in Word or PDF as outlined above;
  - VoiceXML document(s), grammar files (if you use external grammars), Javascripts;
  - two databases in CSV format.
Deadline

- Please submit all files via WebCT.
- The assignment is due at 2pm on Friday 14th September 2007.
Take-Home Messages

• Use OptimTalk to implement the real estate agent in VoiceXML.
• Focus is on dialog design and prompt design.
• Focus is not on database backend integration.
• Make your application as easy as possible to use.
• And as always: start with this assignment as early as possible.