CSCI 3813/780 – Spoken Language Processing
Prof. Rosenberg

Homework 4
Due November 23, 2011 @ 3:00pm

Note: This assignment is **all** group work. There is only one individual work component, see #4 in the Submission section.

Discourse Management

In this homework, you will design and build the discourse management and sentence generation component of your spoken dialog system

This assignment allows you to develop the discourse management component of your system without yet attaching the speech recognition and synthesis components.

For this assignment the input and output to your system will be text. The input text should be expected to follow the grammar of your speech recognizer. You will parse the concepts and values out of the input text using your program from Homework 3. These concepts will then be used to store, retrieve or manipulate information in your system.

For this assignment: Describe in a flow chart the possible dialogs that are possible in your system. An example appears on the following page. The flow chart need not include specific utterances that that will be spoken. Rather, broad concepts about the discourse flow can be used.

Your implementation will then execute this discourse flow. Your goal should be to allow for as natural a discourse as possible without making your grammar overly complicated. A successful system will have multiple paths to a complete dialog. For example, a system that allows a user to provide information in any order or multiple pieces of information in a single utterance would be preferred over one which prompts the user for one piece of information at a time.

It is important to consider how your system will handle errors and corrections. It is important that a user be able to say something to the effect of "you misunderstood me", or "no that's not right", or confirm a choice. You may assume that all of the input utterances are valid in your grammar, but not that they are correct.

You need not write your program in java, but remember that you will eventually be attaching the sphinx speech recognition input. If you decide to use a language other than java, keep this in mind as you design your program. The input may come from the user typing into a keyboard, or from the output of the ASR engine.

Submission

You should submit the following:

1. A readme file that
   a. Explains how to run your program with a command line example. Do not build a GUI interface to your program
2. The discourse flow chart
3. The program that runs the discourse.
4. A file prepared **individually** by each team member, describing how each team member (including yourself) contributed to the submitted project. Note: **do not write one description for the whole team and submit it N times, one for each team member.** This need not take very long to write. However, do not collaborate with your teammates when preparing this write up.

start → **Greeting** → **Cross street or place name?**

Cross street or place name? --street--> **Exact Cross street?**

Cross street or place name? --name--> **place name?**

Exact Cross street? → **Cross street confirmation**

place name? → **name confirmation**

name confirmation --no--> **place name?**

Cross street confirmation --no--> **Ave?**

Cross street confirmation --ok--> **Results / Want another time of day?**

name confirmation --yes--> **Results / Want another time of day?**

Ave? → **Street?**

Street? → **Cross street confirmation**

Results / Want another time of day? --yes--> **when?**

when? → **Results / Want another time of day?**

Results / Want another time of day? --no--> **want another query?**

want another query? --yes--> **Cross street or place name?**

want another query? --no--> **finish**