198:336 Spring 2005
Homework One
Due at the Start of Class
Monday, February 28

This homework explores three uses of conceptual modeling: in designing computer systems that provide interactive decision-support for people; in organizing collections of documents for information retrieval and search; and in designing database applications. It has three problems.

Conceptual modeling is an open-ended process, so there is no single right answer to these questions. That does not mean, however, that every answer is equally good. It is actually easy to make substantial mistakes! The challenge of conceptual modeling is to think and write carefully about the distinctions you need to make in representing the world.

It's OK to talk in groups to exchange ideas and come to an understanding of what the assignment is, but limit your collaboration to this. Don't work out paper solutions in groups; don't share notes; and write up your own work yourself. This homework might look easy, but don't think that you can do a good job on it in just an hour or two. Being careful takes time.

Problem One

Develop a classification of intersections on New Jersey roadways based on the rules that govern left turns there.

Motivation: If you've use online direction giving services, you've probably seen that these systems know a lot about geography but not very much about driving. For a long time, they had no understanding of what an exit ramp was; they'd give directions like "turn right onto unnamed road" that no person would ever give. Online direction systems still don't give you the kinds of heads-up about traffic patterns that people often use to make following directions easier. Good directions will say things like "get over into the left hand lane because you're going to have to turn soon". To give better directions, a system would have to know about the different kinds of turns that are possible, and about the different traffic rules that govern the different kinds of turns. This is a conceptual model. Of course, the system would also need a lot of data—it would need to know the intersection type for each intersection that it has information about. (If you want to know why online directions are so limited in the kinds of directions they can provide, it's because of the difficulty of gathering better information, and not because of the inherent difficulty of supporting more effective interaction.) You probably know quite a bit about driving already, so this exercise also has the advantage of pointing out some of the complexity of the information you use to act effectively in the world.

Details. Your answer should take the form of a network of concepts for types of intersections organized into a network connected by subtype or *isa* links. Note that you do not have to create concepts for all of the entities that might be involved in conceptualizing driving—cars, roads, signs, lights, and so forth; you can focus exclusively on the intersections. This is why only *isa* links are required.

For each concept, you should briefly describe the real-world category the concept is intended to correspond to. You should also briefly specify the distinctive considerations that govern left turns at such intersections. There are obviously a lot of ways that intersections differ that don't have anything to do with left turns—you should ignore these features, because otherwise you will probably encounter an overwhelming array of possibilities! So you should consider things like whether a left turn is possible at all; how the traffic signal, if any, governs the left turn; and where you go to make the left turn.

Problem Two

Develop a dictionary entry for indexing the noun *stroke* based on the WordSketch described for it at

`http://www.itri.brighton.ac.uk/~Adam.Kilgarriff/WORDSKETCHES/stroke_n.html`

Remember that a WordSketch is explicitly designed to show off all the different ways of using a word in English. By grouping these uses together you can be confident that you've organized all the meanings that are required.

Motivation. You will see that the word *stroke* is highly ambiguous. Not only does its meaning vary, but even the kind of meaning it has varies. Sometimes it contributes to the key content that's presented in a document, for example when it refers to brain injury. In other cases, it is an emphatic word that is simply used to emphasize the suddenness with which something happens—as in the expressions *stroke of luck* or *stroke of midnight*. If we want to index documents using controlled collections of concepts (like the UMLS or the Semantic Web), we need to distinguish between different senses of key terms.

If you think about using specific concepts in a document retrieval or search system, what's important is to organize concepts into a coherent structure that highlights differences that are likely to be important for peoples' information needs. For example, it might be useful to distinguish the different kinds of *stroke* that are used in swimming and golf. If you want to know about one you're not necessarily very likely to want to know about the other. This is different to the kind of dictionary entries you might write for a natural language processing system in artificial intelligence or for a printed dictionary for learning English. For these other applications perhaps it would be enough to say that a *stroke* as in swimming or golf is just any swinging motion, involving the application of force to something stationary, which can be repeated.

Interestingly, if you reflect on the word sketches, you can see some of why search engines like google don't use word senses, and why clever searches can often get around the difficulty. With today's technology it would require a lot of time and effort to model the concepts that are used in any document. On the other hand, the cues you would use to draw distinctions are often directly searchable. They are just a little counterintuitive. For example, if you want to find the medical sense of stroke, you can just search for *stroke* and the word *after*!

Details. Your answer should take the form of a list of more specific concepts that a writer may have in mind when they use the noun *stroke*. For each one, you should give a brief statement of the meaning. Then select a good illustrative sentence from the data presented in the word sketch to illustrate this sense of the word. You may want to include several sentences if you think that a common general meaning applies across a range of contexts that might superficially seem different, but say why you think the differences wouldn't matter for document retrieval. Finally, put together a bunch of patterns (head entries of the word sketch) to give an overall of some of the ways this sense is commonly used.

Remember, this is different from the last problem. No brainstorming is required. All the data you need to do the problem is available as part of the sketch. In this problem, the focus is on analysis: grouping ideas together and drawing distinctions that are likely to be useful to an application or to particular users.

Problem Three

Draw an entity-relationship diagram and build an SQL database schema for information about collections of music.

Motivation. The typical interfaces that you find in MP3 software like Musicmatch or even iTunes is pretty limited when it comes to representing the relationships among tracks. In many cases, the software pretty much assumes a pop-album model where tracks are grouped together into albums by a single artist and the tracks can pretty much be selected individually and shuffled in any order. This gets confused with DJ mixes or classical pieces. You could definitely do this better if you took advantage of general database technology instead of trying to store information about tracks in a particular predefined format.

Details. Your answer should take the form of an ER diagram of your database, and a database schema that corresponds to it. You should include a brief statement explaining your design.

Your design should accommodate at least three kinds of collections. There are pop albums with a uniform collection of tracks. There are mix albums where tracks have different performers and titles and are organized in sequence by another artist (the DJ). And there classical albums that are made up of pieces by particular composers and played by particular performers; each piece may have many tracks because it contains several movements.

You may have to design new abstractions and entities in order to organize the data naturally and avoid storing information redundantly. As you design the schema, think about making it easy to answer questions like: Who is playing the violin on the current track? What pieces in the collection were composed by Beethoven? On what albums does David Holmes mix three or more soul tracks?