6.871 PROJECT DESCRIPTIONS

Please submit one proposal per project group (or two at most, if you are unsure), containing:

- A list of the group members (remember: no more than 2)
- A description of the problem that answers these questions:
  - What will the program do?
  - Indicate, *as precisely and concretely as possible* what the input to the program will be.
  - Indicate, *as precisely and concretely as possible* what the output from the program will be.
  - Indicate the scope of coverage you are aiming for (i.e., what class or family of problems will your program be able to do?)
- A description of the knowledge
  - What will the program have to know? Answer this in two ways:
    - Give one concrete example of the program in simulated operation, by creating one specific example of input and output, then using this to "walk through" a simulated run of the program. This will help you to get a feeling for what kind of and how much knowledge is required.
    - If possible, describe the breadth of knowledge (i.e., the discipline of sub-disciplines that are involved).
  - Give at least two examples of specific chunks of knowledge that you believe would be useful for the program to have in its knowledge base.

Example

What will program do; scope:

The program will help a newcomer to Boston by providing advice on what neighborhood they are likely to enjoy living in.
Scope: It will be limited to Boston, Cambridge, Newton, Brookline and Belmont.

Input:

The input will be information about the newcomer including age, marital status, children, hobbies, interest in sports, interest in restaurants, theater, amount of open space desired, whether they own a car, whether they want to rent or buy, etc.

Output:

The output will be an ordered list of Boston area towns and neighborhoods that would appeal to them.

Concrete example:

Input: Married couple, in their 20's, no children, enjoy theater and restaurants, not avid sports fans, enjoy having things nearby, no car, willing to rent.

Output: Harvard Square, Coolidge Corner, Jamaica Plain.

Example of relevant knowledge: sample rules

```
No Children ⇒ condition of school systems is less important
Desire for theater and restaurants ⇒ suburbs less interesting
No car ⇒ suburbs less interesting
Willing to rent ⇒ city areas with apartments are more likely
```

General Points to Keep in Mind

- You can combine this with thesis or other work, but the project *must* meet the requirements for this class. We will not accept a project done for another reason that has simply been dressed up after the fact to look like a knowledge based system. If you are at all unsure, come talk to us first; don't wait.

  On the other hand, we have no problem if the project for this course also satisfies other obligations you may have. For example, we routinely have people doing course projects that are part of their master's thesis work.

- Watch out for tasks that are essentially table look-up (i.e., the reasoning is one step deep). Look carefully for situations in which the reasoning is legitimately several levels. If you can't easily find any, the problem may not be well suited for this class.

- Watch out for decision trees: if it seems natural to think about the domain in terms of a cut and dried sequence of decisions, that may indeed be the best way, and it may be inappropriate for this class.

- Feel free to solicit our advice.

- Platforms available include: Kappa-PC (PC), M4 (PC), and GRASS (PC), all of which are variations on rule-based system engines. There is also Joshua, which runs on in Common Lisp on the server, for both MAC and PC. Joshua will be introduced later in the course, and is the preferred tool for projects.