CMSI 682 KNOWLEDGE-BASED SYSTEMS

Spring 2008
Dr. Stephanie E. August

Term Project

Task	Due Date
T#1 Describe and motivate the project	6 February 2008
T#2 Describe the knowledge *Knowledge Identification Phase*	27 February 2008
T#3 Organize the knowledge *Knowledge Conceptualization Phase*	27 February 2008
T#4 Document *Knowledge Formalization Phase*	27 February 2008
T#5 Progress report #1, written and oral	27 February 2008
T#6 Add *blackboards* or *uncertainty factors* or *??*	2 April 2008
T#7 Progress report #2, written and oral	2 April 2008
T#8 Complete implementation *Knowledge Implementation Phase* *Knowledge Testing Phase*	30 April 2008
T#9 Demo	30 April 2008
T#10 Evaluate	30 April 2008
T#11 Lessons learned	30 April 2008
T#12 Final report and demonstration	30 April 2008

Introduction

The goal of the class project is to learn first-hand about issues involved in implementing a knowledge-based system. It includes four components: design, implementation, documentation, and research. The balance among these four components will vary according to each student's interests and the domain selected. The project constitutes 45% of your grade in the class.

The project will be implemented using the CLIPS expert system language, which will be introduced in class. To successfully complete the project, each student will need access to CLIPS itself, as well as the *CLIPS Basic Programming Guide*, and *CLIPS User's Guide*. These materials are available on the CD that accompanies the course text and can also be found at http://www.ghg.net/clips. They will be made available in the Keck Lab as needed.

The project deliverables include an oral report, a written report, and code. The latter includes all of the rules, facts, procedures, and test drivers developed during the course of the project, as well as a `readme` file which explains how to run the code. The code can be posted to your LionShare account for review during the term, but the final version of the project should be submitted on a CD secured in the project notebook. All final results are due by start of class on 30 April 2008. Late projects will be docked three points for each day they are late. Demonstrations should be integrated into the final oral report. Grades will take into consideration the difficulty of the chosen problem, the student's mastery of the subject, and the amount of work completed, both in terms of implementation and writeup, and the overall quality of the project paper, demo, and oral project report. The remainder of this writeup describes the required components of the project and due dates for each.

All project reports must be kept in a three-ring notebook. The entire notebook should be submitted on each due date, and will be available no later than the following class meeting. Dividers should be used to organize information in the notebook. There should be a way of securing the CD in the notebook.

1. **Describe and motivate the project (The Introduction) (due 6 February 2008)**
   This section introduces the project to the reader (your instructor at this point!), explains why it is a worthwhile project, identifies the class of problems the system is expected to solve, and generally serves to interest the reader in the project. It should clearly state the domain of your knowledge-based system, such as fish disease diagnosis, diamond appraising, or an automated travel agent; explain why there is a need to develop a KB system for this domain; identify who the targeted end-user will be (or would be, if the system were completed and fielded); and specify the target platform for your system (what hardware, operating system, and version of CLIPS it will run on). Be as specific as possible about what you expect to accomplish. Explain why developing a knowledge-based system is an appropriate approach to solving the problem, and give some insight into the knowledge that will be encoded in the system, what the user will provide as input, and the kind of output the system will provide.

   A written project proposal including the description of and motivation for the project is due on 6 February. Students will be informed via email by end of business on 9 February 2008, if any adjustments are needed to the project proposals as submitted.

   (See pages 326-328 in the Gonzalez and Dankel text for a brief description of information that might be included.)

2. **Describe the knowledge (due 27 February 2008)**
   During this phase of the project, you take on the role of knowledge engineer. Your task is to interview the knowledge expert(s) and record their responses using a combination of natural language narrative and relevant special notations drawn from fields such as mathematics and physical science. Identify problem characteristics and the class of problems the system will be expected to solve. Identify the resources available for the project in terms of expertise, man-power, time constraints, computing facilities, and money.

   At this stage, don't be concerned about your representational schema. Do try to separate knowledge about objects or structures (object descriptions) from knowledge about procedures or rules.

   This is the *knowledge identification phase* of the project and produces the project *requirements*.

3. **Organize the Knowledge (due 27 February 2008)**

   See section 12.2 pages 326-328 in the Gonzalez and Dankel text, and problems 12-3, 12-4, and 12-6 for an idea of how to approach this section.

This is the *knowledge conceptualization* phase of the project and produces the key *concepts* to be incorporated into the system. The goal is to develop a formal description of knowledge in terms of primitive concepts and conceptual relations. This task entails probing more deeply into the content of the interview protocols. Our objective is to eliminate the inevitable redundancy and ambiguity of the natural language description.

During this phase, seek out concepts to represent the knowledge acquired from your expert and uncover the key concepts and the relationships between these concepts. Characterize the various kinds of data involved, the flow of information within the system, and the underlying structure of the domain in terms of causal, spatio-temporal, and part-whole relationships.

Identify the answers(s) or solution(s) to the problem being tackled. These constitute the output of the system, and represent the goal(s) that you, in the role of expert, and the knowledge-based system are expected to reach when searching for an answer.

Identify the source(s) of information that you will use to deduce the solution. These constitute the inputs to the system. Explain the source and significance of these sources.

Determine how to map from the input to the output. This mapping represents the core of the expert knowledge within the system. It can be represented as a set of rules or objects, stated informally in English. What you need to represent will depend upon the requirements of your system.

See knowledge organization, p.354-356 in the Gonzalez and Dankel text, for an example.

4. Documentation (due 27 February 2008)

This is the *knowledge formalization* phase of the project. During this phase you design the structures in which to organize the expert knowledge contained in the system. The structure must reflect an understanding of the nature of the underlying search space, the character of the search to be conducted, and the c*ertainty* and *completeness* of the information contained in the system, as well as other constraints on the logical interpretation of the data such as time dependencies and the *reliability* and *consistency* of various data sources. It must also reflect the paradigm to be used for system implementation.

Identify the specific knowledge structures and procedures to be used. If you are using an inference-network based system, this is where you will include a diagram of the network. For each rule, you should specify what the input parameters of the rule are, whether the premise of the rule forms a conjunct, a disjunct, or a combination of the two, whether a particular premise can lead to multiple conclusions, and what the output parameters are, if any. For each object, describe the type of each object and its attributes.

See the discussion of knowledge documentation in Gonzalez and Dankel section 13.3, p.356-360; section 13.7.

5. **Progress report #1** **(due 27 February 2008)**
Prepare a written report incorporating the information developed in tasks one through five above. The report should include the previously submitted description of the project (including any recommended changes), the motivation and justification for the project, and a plan of attack. The plan should include risk analysis and clearly indicate due dates for other academic and job-related projects, which could lead to time conflicts later in the semester. Prepare a brief (approximately five minute) oral status report to be given in class. The *oral* report should list tasks, achievements, and risks ahead, *without* going into detail about the knowledge in your system. This is in contrast to the written report, which should be comprehensive.

   Your project notebook will be returned in class on 5 March or can be picked up ahead if prior arrangements are made.

6. **Do one of the following.**
(outline due 2 April 2008, completed task due 30 April 2006)

   i. **Blackboards (Multiple Agendas)**
   Design of a blackboard system (or one using multiple agendas) that could be used in conjunction with your term project, *using the steps outlined in Ex.7-1*.

   ii. **Certainty Factors and/or Vagueness**
   Describe how uncertainty management could be incorporated into your term project, complete with examples. Select the most appropriate uncertainty method, and justify your choice. How would uncertain data be represented? How would two or more pieces of uncertain data be combined? How would inferences be drawn using uncertain data? Design an uncertainty management mechanism for your project, and work through an example. Integrate uncertainty management into your project, and demonstrate its effect on execution.

   See 7-1, 7-3, 7-4, and 7-5 Gonzalez and Dankel for additional detail.

7. **Progress report #2** **(due 2 April 2008)**
Prepare a written report incorporating the information developed to date. Turn it in along with the previously submitted task writeups (ALL revised *and* originally submitted versions), and an updated schedule (*and* ALL previously submitted schedules, if the schedule has been rewritten rather than just updated).

   Prepare a brief oral status. Again, the *oral* report should list tasks, achievements, and risks ahead, *without* going into detail about the knowledge in your system. This is in contrast to the written report, which should be comprehensive.

   Your project notebook will be returned in class on 9 April or can be picked up ahead if prior arrangements are made.

8. **Complete Implementation** **(due 30 April 2008)**
This task includes two phases: the *knowledge implementation* phase of the project followed by the *testing* phase. Implementation produces a *working program*, while testing *produces refinements of implementation, redesigns of formalization, formulations of*

conceptualization, and *reformulations of identification.* Here you formulate rules to embody knowledge and specify both flow of *control* and the details of *information* flow. You express the rules in some executable form under a chosen control regime, design data structures and define the degree of independence between different modules of the program.

Testing validates the rules that organize knowledge in the system. To complete this phase, you will need to run the program on a *large* and *representative* sample of test cases. Be alert for common sources of error, such as rules that are *missing*, *incomplete*, or *wholly incorrect* as well as competition between related rules.

The completed implementation deliverable includes all of the rules, facts, procedures, and test drivers developed during the course of the project, as well as a `readme` file which explains how to run the code. The implementation should be delivered on a CD unless other arrangements are made prior to the due date.

9. Demo (due 30 April 2008)

Demonstrate your project for the class. Present a running commentary of your code as it executes.

10. Evaluation of the system (Technical review). (due 30 April 2008)

How successful were you in implementing the desired system? What proved to be the simplest aspect of the project? What was the most challenging aspect? Could your system be successfully scaled up? Explain.

11. Lessons learned (Management review) (due 30 April 2008)

What would you do differently next time? What knowledge representation techniques should you have used, or do you wish had been available? How many hours did you put in? How useful was the project in helping you learn the course material? What would facilitated completing the project?

Note: This is ***not*** an optional component of the project. You ***will*** lose points if you do not include this section.

12. Final project report (due 30 April 2008)

The completed project notebook must contain all of the information described in the above tasks, as well as all interim reports and a hard copy of your code. The code itself should be included on a CD in your notebook. Your report should include a project title page, table of contents, annotated trace of execution, and bibliography. Number and labels figures (figure title belongs below figure) and titles (table title belongs above table). Pages should be numbered. Use 12 point font and one-inch margins.