

COMPSCI 367 Assignment 3 Briefing

This assignment will involve designing and implementing a prototype **knowledge-based system** (in the domain of your choice) using a KBS development language, environment or shell of your choice (e.g., CLIPS or JESS). It will also involve a **written report** describing the KBS you create.

The assignment is worth **15%** of the total marks for CompSci.367 and it will be graded out of 100 marks.

Deadline: Saturday 21st October 2006 11.59 pm (last day of lectures)

Frequently Asked Questions

What should I do it on?

Optimally, there should be some “*depth*” to the problem domain you choose for this project; that is, some chains of reasoning should be necessary to solve the problem you have chosen. This means that you should focus tightly on a particular problem. Good examples are diagnosing types of problems with computers, deciding on which computer or peripheral to purchase (e.g., product selection), or something technical like choosing the correct pesticide to spray insect infected plants.

It should also be primarily a *heuristic* domain, for which no simple formulaic rules or algorithms exist. It should probably also be one for which you have an easily accessible *source* for your expertise.

It does not have to be about computers, engineering, science, etc., in fact, some of the best assignments I have received have been about things like music, cooking or sport. Pick a topic that you both have an interest in and are perhaps expert in.

Do **NOT** chose a game like tic tac toe (noughts & crosses) or chess.

If you have questions about a particular topic, please contact me (ian@cs.auckland.ac.nz).

How large does it have to be?

This will be a prototype KBS, so I don't expect hundreds of rules. Don't panic if the problem you choose turns out to be larger and harder than you first anticipated -- in that event, you can simplify the problem by narrowing the domain as much as necessary. Focus in on a *specific* part of the problem, rather than the whole problem.

Remember the assignment is worth 15% of the marks for COMPSCI 367 so you should schedule your time accordingly.

What else should I do?

It is important that you document how you acquired the knowledge for your expert system. Did it come from a book or user manual or perhaps a website? If so give a reference. Did you elicit the knowledge from a friend who is expert in the problem domain or did you have the expertise yourself? If you elicited knowledge from a person, what elicitation techniques did you use? It is very important in your report that you document the knowledge using a suitable intermediate knowledge level representation, like decision trees, decision tables or inference nets, and produce easily understandable English-like pseudocode for your rules.

Where should I get my knowledge?

You should consult some sort of verifiable **reference** for the rules you create. This source may be a human expert, books, user manuals etc... Multiple sources (the more varied the better) are strongly encouraged. These sources should be listed in a *bibliography* section in your written report.

In addition, the **test problems** you use to test your system should also come from one of your references, so that you will have some **desired results** for those tests to verify your system.

Can I use Java or .Net?

Yes you can use Java or any other programming language you like, but you will have to create a declarative method of representing your rules (perhaps as a .TXT file that can be parsed in by Java) and you will need to programme an inference engine to reason over your rules. I suggest you look at Jess (available on the resources page of the course website) which provides a Java wrapper for CLIPS.

What kind of written report do you want?

Your report should cover the following subjects:

- briefly introducing the problem domain;
- describe the sources that you acquired your knowledge from (these should also be listed as references in the bibliography), and the methods you used to elicit the knowledge;
- it is essential that you describe the knowledge using an intermediate knowledge level representation;
- then describe your rules in pseudocode (i.e., IF THEN rules) and then list your rules in the knowledge representation of your chosen implementation language,
- list the **test problems** that you used to validate your system along with the required inputs and expected outputs; and
- finally, include installation instructions – e.g., does your system need to be installed into a particular directory or run on a particular operating system. Note it is **strongly** recommended that you test that your system will run on a computer other than the one on which you developed it. Markers will not spend long trying to get your system to install and run.

You should also describe any problems that you encountered and observations that you made in this process. Make sure that you mention anything that is *extra* or *interesting* about the system you have created (particularly if you think it might get you extra credit!).

This report does not have to be a major piece of work, as long as it covers the material listed above -- It probably won't be longer than 3-4 pages (aside from the reference materials).

What do I need to hand in?

The following are due by the deadline

- Your **written report**.
- The file(s) containing your KBS program code.
- Clear installation instructions (e.g. readme.txt)
- Submit all your files in a single zip file using the web assignment dropbox (<https://adb.ec.auckland.ac.nz/>).

Marking Criteria

Marking will be based on:

- The complexity of the **domain** you choose, and the success of your rules in representing the domain (10%)
- The number and quality of the **reference sources** you use (5%)
- The **intermediate knowledge level model** of the knowledge you have elicited (20%)
- The **correct functioning** of your programme (20%)
- as well as the completeness and validity of your **test problems**. (10%)
- The quality of your **written report**, particularly with respect to knowledge elicitation and modelling, references, and explanations of the problem and your approach, (30%)
- A maximum of 5% of the total marks will be available for the **interface** to the programme.