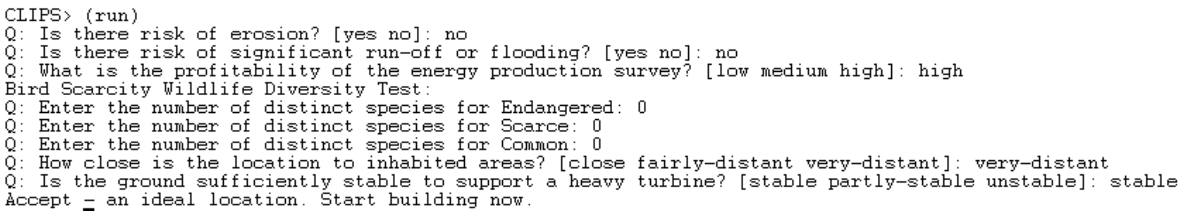
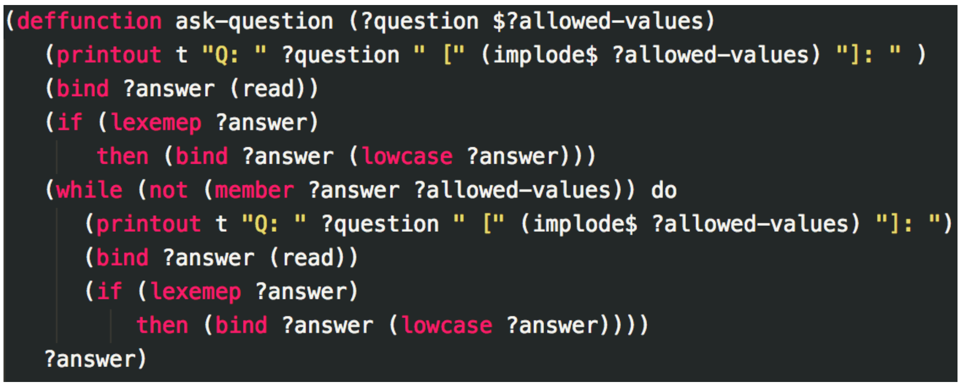
Windy-ES Knowledge-based System

The knowledge-base proposed uses input from the user to help consult a decision over the suitability of a location for a wind farm. Depending on the input given, the rule-base will fire the specific rules that correspond with the user's input (e.g. if distance from population was very far, there will be no need to ask about the sound impact). Therefore, the user is taken through multiple questions with a selection of answers until a verdict is reached. Those being an ideal location which states the wind farm has no clear issues, a second-best location which states what can be done to improve the location i.e. stabilise the land and any rejections that explains why the location is unsuitable.

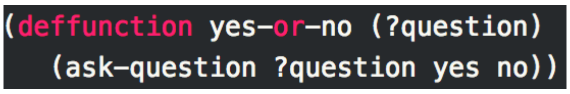


```
CLIPS> (run)
Q: Is there risk of erosion? [yes no]: no
Q: Is there risk of significant run-off or flooding? [yes no]: no
Q: What is the profitability of the energy production survey? [low medium high]: high
Bird Scarcity Wildlife Diversity Test:
Q: Enter the number of distinct species for Endangered: 0
Q: Enter the number of distinct species for Scarce: 0
Q: Enter the number of distinct species for Common: 0
Q: How close is the location to inhabited areas? [close fairly-distant very-distant]: very-distant
Q: Is the ground sufficiently stable to support a heavy turbine? [stable partly-stable unstable]: stable
Accept - an ideal location. Start building now.
```

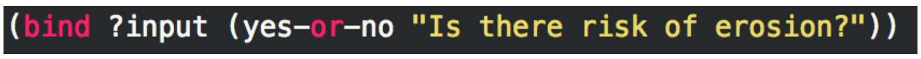
As the knowledge-base is using input from the user, it was important that input validation was included to prevent any potential errors or incorrect rules firing. Each question uses one of two deffunctions to validate input that is either of type string or integer depending on the type of question asked. It also prevents incorrect answers being submitted by being passed 'allowed-values'. For example, if the question needed a yes or no answer, it would only accept these two string values otherwise if any other word was inputted, the system will ask the user again for the correct input.



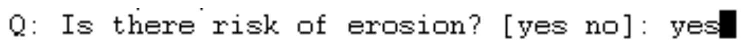
```
(deffunction ask-question (?question $?allowed-values)
   (printout t "Q: " ?question " [" (implode$ ?allowed-values) "]: " )
   (bind ?answer (read))
   (if (lexemep ?answer)
       then (bind ?answer (lowcase ?answer)))
   (while (not (member ?answer ?allowed-values)) do
      (printout t "Q: " ?question " [" (implode$ ?allowed-values) "]: ")
      (bind ?answer (read))
      (if (lexemep ?answer)
          then (bind ?answer (lowcase ?answer))))
   ?answer)
```



```
(deffunction yes-or-no (?question)
   (ask-question ?question yes no))
```



```
(bind ?input (yes-or-no "Is there risk of erosion?"))
```

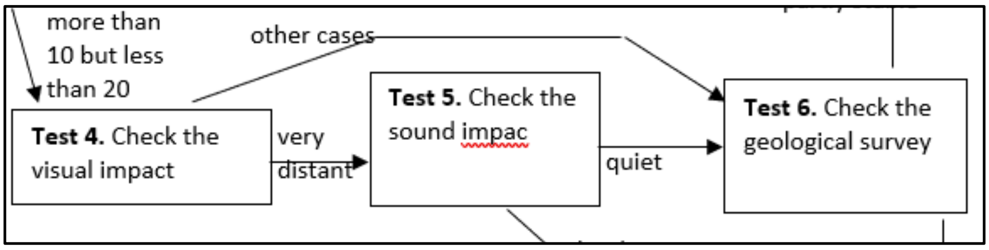


```
Q: Is there risk of erosion? [yes no]: yes
```

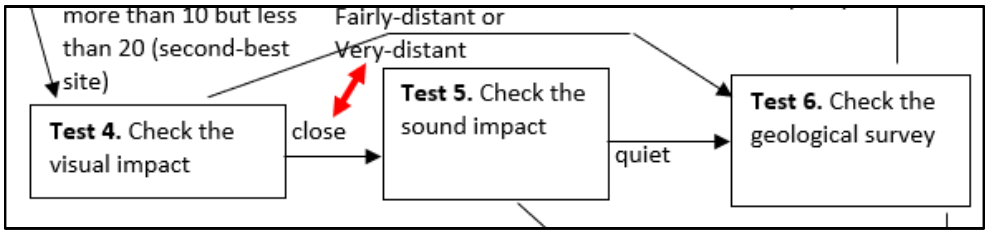
This is beneficial as it allows the rules to have both stricter and less complicated conditions preventing any accidental firing and longer, repeated code.
Input and output dialog were chosen as it allowed the user to feel as if an actual consultation was taking place whilst also ensuring that any input was meaningful (not asking unnecessary questions). This is important as it prevents any misinformation (facts) entering the rule-base and keeps the user informed as they're constantly updated with new questions which also helps them understand why the rule-based system comes to a certain decision. To ensure an actual consultation was occurring, questions and answers (reasons of suitability) had to be detailed (if necessary) meaning that instead of saying "What was the visual-impact?", use "How close is the location to inhabited areas?".

A change was also made to the decision tree provided which I believed to be inaccurate and either a mistake/unintended. This was the 'Check the visual impact' test where for an ideal site, it would require a case of 'very-distant' where it would also bypass asking for the sound impact as the site is already faraway from local towns etc. Whereas, another test for 'Check the visual impact' had the 'very-distant' case firing the 'check for sound impact' test which shows there is a conflict. Therefore, I changed the case to require either 'close' or 'fairly-distant' (known as 'other cases' on the decision tree) so that the system uses the three visual-impact cases accurately and correctly.

Original Decision Tree


more than 10 but less than 20
other cases
Test 4. Check the visual impact
very distant
Test 5. Check the sound impac
quiet
Test 6. Check the geological survey

Revised Decision Tree


more than 10 but less than 20 (second-best site)
Fairly-distant or Very-distant
Test 4. Check the visual impact
close
Test 5. Check the sound impact
quiet
Test 6. Check the geological survey