

## EXPERT SYSTEMS COMPUTER SHELL DEMONSTRATION

TONY STARFIELD: The expert system shell we are going to demonstrate is called "WINEXP." And when one loads WINEXP, one sees the icon with Sherlock Holmes in it, and that means we now have the shell in the computer. When you click OK, up comes this screen.

Now, the shell loaded into the computer is kind of turning the computer into a brain that knows how to interact and think but has nothing to think about. So, basically, what we have to do is make it an expert in a particular subject.

So if I click on the left-hand icon here, up comes various examples of knowledge bases. And we're going to choose the one called "Burn" because this is the one that relates to the fire problem in the Kruger Park. We click OK on that, and the first thing that the shell does is take us into what's called the "Context," which tells us what this particular knowledge base does. And I'm not going to scroll through this, but it basically gives you a bit of the history of fire and fire management in the Kruger Park that I talked about just before this demonstration.

Once you've made sure that the shell is going to meet your needs, you close that. And now you can say, "I want advice." So you click on the button that says, "Ask," and the computer starts asking you questions.

And the very first question it asks is, "Is there old, moribund vegetation that needs to be removed?" And as an intelligent user, you don't just answer questions; you need to ask, "Why?" So you click on the "Why" button, and up comes an explanation. And the first part of the explanation is that old vegetation may inhibit the growth of new grass. So it's giving you a rationale for why it's asking the question. And it also says, "All material provides fuel for a fire."

And then when you look to see more about the explanation, it tells you that it's actually trying to validate a decision. And that decision says, "One cannot sustain a fire with insufficient fuel." In other words, it's trying to make sure that there is enough fuel for a fire before it gives you advice about how to burn or when to burn.

So now you know where the question is coming from. You close the explanation. And we are going to answer, "Yes." Click OK, and up comes the next question.

"Has the area recently been burnt accidentally?" Well, if you look at the "why" here, it says that if it has, there is unlikely to be a good reason to burn again. It's also unlikely there will be sufficient fuel to sustain a fire. And it is still trying to validate the first decision about fuel.

Now, you might say, "Gee, this is odd. I've just told the computer that there is sufficient fuel, but it's double checking on me." And this raises an interesting point. When you develop an expert system, you're not only modeling the expertise, you also have to have in mind a model of your user and how competent your user is and how capable your user is of asking questions. And so this little second check on whether or not there is enough fuel basically tells you that this system was designed for a user that you didn't have a lot of confidence in. So the computer is double checking the first answer.

We're going to say, "No, the area hasn't recently been burnt."

And the next question it asks is, "Are the shrubs in the area becoming dense or growing high?" And again you ask, "Why?" And it tells you that bush encroachment can be a problem in African savannas. And we're looking for evidence of bush encroachment.

And then the next part of the explanation is that it is trying to validate Decision 2. And this Decision 2 is a fairly complicated decision. But basically, what it's looking for is evidence that you need to burn because of bush encroachment. So you go back, and you look around, and maybe you have some pictures which tell you what bush encroachment looks like. And you say, "No, not a problem."

And then the next question is kind of enigmatic. It says, "Is the density of *Bothriochloa* higher in recently burnt neighboring areas?" And again, to get to what's behind this question, a user really needs to click on "Why." And what it tells you is that *Bothriochloa* is a pioneer species that thrives in degraded areas. It's a sign that too frequent burning is having a negative effect on the vegetation. So here, the designer of the knowledge base has recognized that you can't just ask the user to tell you whether or not he or she thinks that the area is becoming degraded. So it uses the presence or absence of *Bothriochloa* as a cue to enable the computer to decide whether or not it's becoming degraded.

And suppose we say, "Yes." Seems to be *Bothriochloa* all over the place. You click on OK again, and it tells you it has reached the decision. And the decision is to defer burning for a year.

And you ask, "Why?" And it says, "If there is no urgent reason to burn, then we want to avoid degradation of the vegetation. It is likely to accelerate soil erosion."

So I think you get a sense of how the computer was thinking here. Now this particular shell, once it's reached the decision, can go on and reach more decisions. So if we say, "Look for additional decisions," it tells us that, in this case, there aren't any. And that is the conversation that a user might have with a shell.

Now, there are two things worth looking at in this particular example. The first is if we click on the "Trail" button, and what we get is a kind of replay of the questions that you were asked, how you answered, and what decisions were discarded along the way to reach the decision that it eventually came up with. So you see it discarded the decision that said, "Do not burn." It discarded the decision that said, "Burn after the first rains," and so on until it eventually reached the decision, "Defer burning for a year." So you have a record of your conversation with the computer.

The second thing that is useful to look at here is to click on the word "List" and actually have a look at the knowledge base

that is behind this particular interaction. And, as I said at the beginning, first, it goes through the context, telling you what the knowledge base does. And then it lists the possible decisions that can be reached by this particular expert system. And, as you can see, there are only five decisions. So if you look at the decisions in the knowledge base, that tells you very quickly what the scope of the expert system is. So we kick off with a list of the decisions.

And then the next part of the knowledge base is a list of the questions. And the questions are all numbered. And you see the first question was, "Is there old, moribund vegetation that needs to be removed?" Now, the numbering of the questions is unimportant because the computer's going to figure out on the basis of your answers which questions it wants to ask. But you can see the format of a question. There's a question, then there is the explanation behind the question.

And that explanation behind the question serves two purposes. The one is to, perhaps, help the user to answer the question, and the other is to explain to the user why that question is being asked. And then the question can have any number of answers. In this case, it's "yes" or "no." But as we scroll down, we find that there's a question, for example, that has three answers

where the answers are "Long grass grazers, short grass grazers, and no priority." And all told, this particular knowledge base has only six questions.

The last part of the knowledge base is the set of rules. And the rules are basically the logic, which enables the computer to reach a decision. And in this particular case, the format of the rules is "If," and then there's a logical expression relating to answers to questions. And the "Then" part of the rule is a decision number.

So if you had to say this rule, "If question 1, answer 2 or question 2, answer 1, then decision 1" in words, you would say, "If the answer to question 1 is either 2 or 1, whatever that is, then decision 1 is true." And the explanation on the rule is meant to spell out what that rule is saying and to justify the rule if necessary. And we've already seen the explanation, "One cannot sustain a fire with insufficient fuel." And this particular knowledge base has seven rules.

So when we talk about small expert systems, and we talk about the expert system experts telling us that knowledge bases need to be large to be effective, they are talking about hundreds, maybe even thousands of rules. Seven rules seems to me to be

about as simple as one can get. The question we have to ask is,  
"Is this trivial, or is it useful?"

[END OF VIDEO]

Document: blm0212.doc

Transcribed by: <http://www.hiredhand.com>