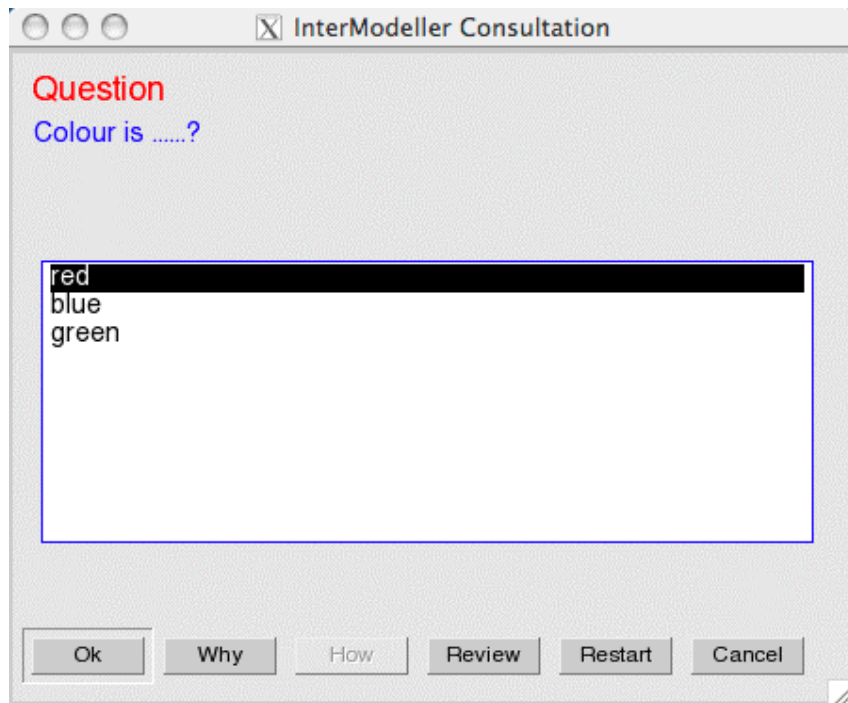


# Expert System Shells

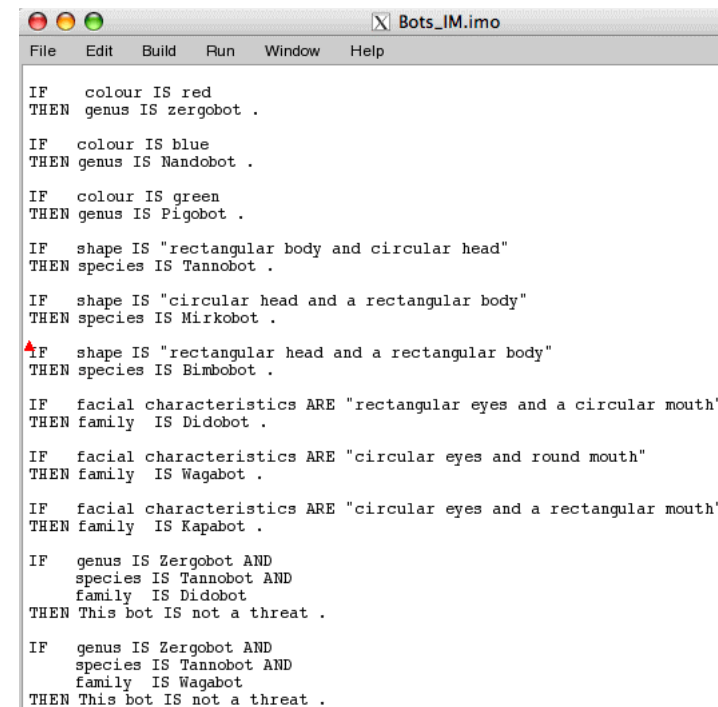
An **expert system shell** (such as Intermodellar) has a ready made **user interface** and contains an **inference engine**.

Information in the form of facts and rules is added to the **knowledge base**.

## User Interface



## Knowledge Base



We do not see the inference engine, but it has a very powerful role to play in the function of the expert system.

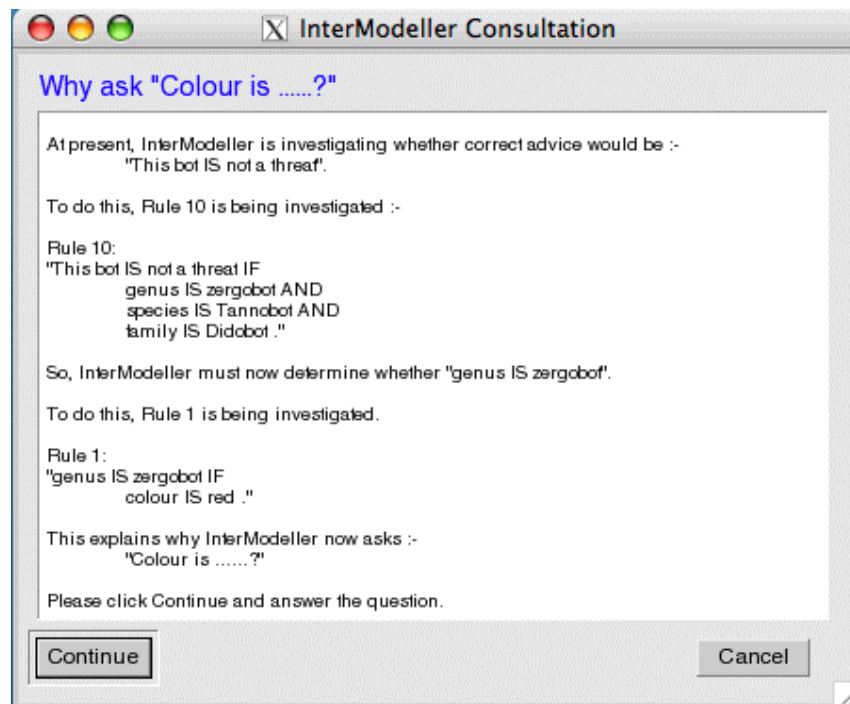
# Expert System Shells

An **expert system shell** asks questions and provides **advice** and **justification** (all via the user interface).

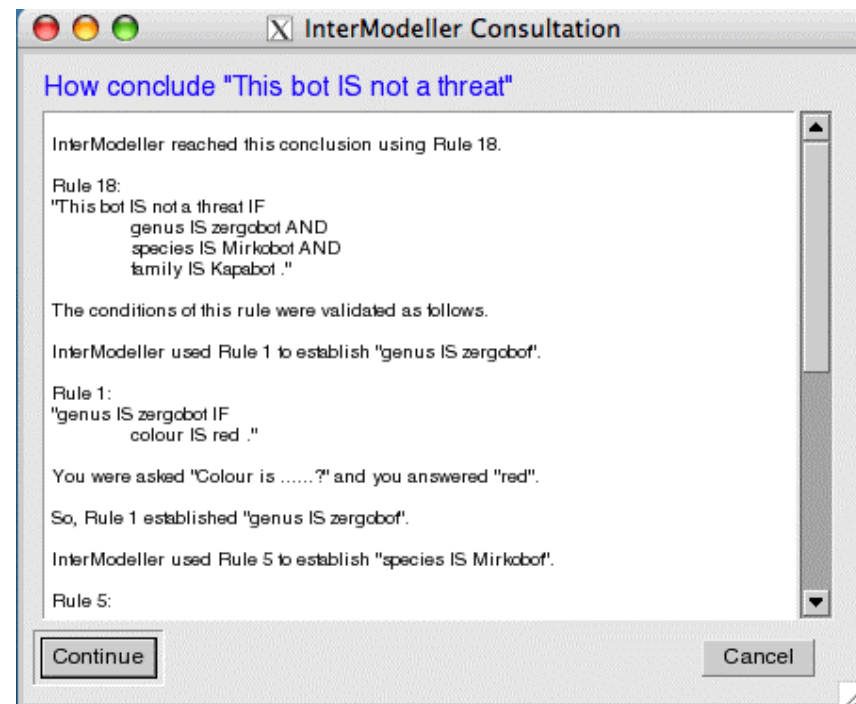
Justification comes in two parts:

**Why** - explanation of the reasoning for asking a particular question.

**How** - explanation of the reasoning in providing a particular piece of advice.



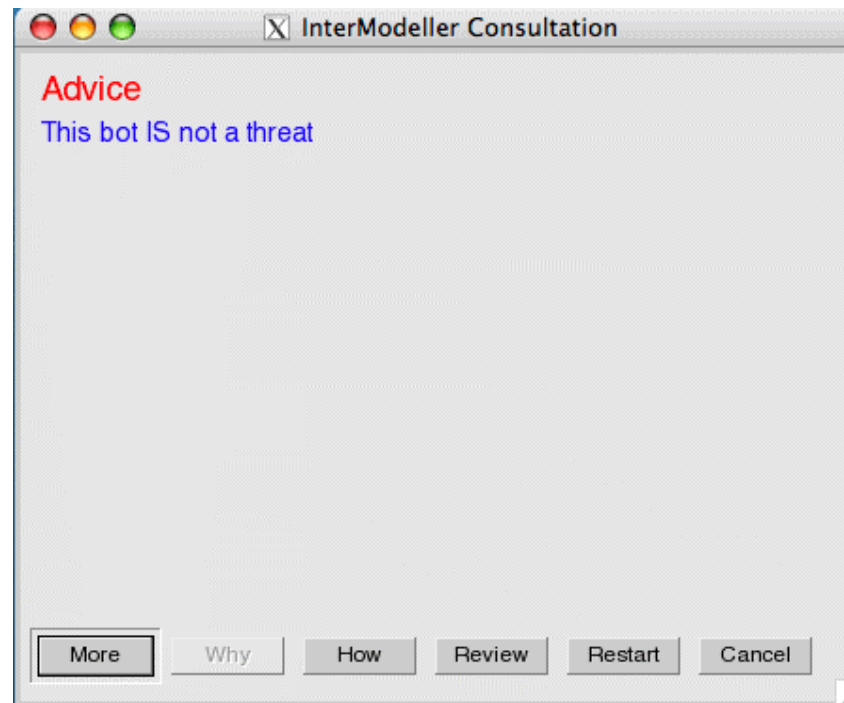
**Why Justification**



**How Justification**

# Expert System Shells

An **expert system shell** asks questions and provides **advice** and **justification** (all via the user interface).



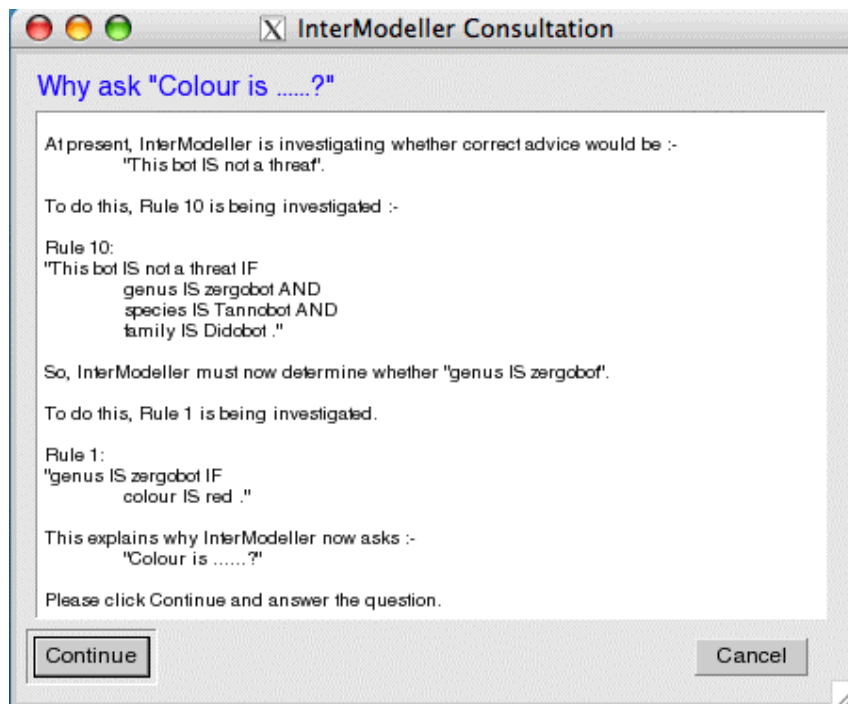
**Advice**

# Expert System Shells

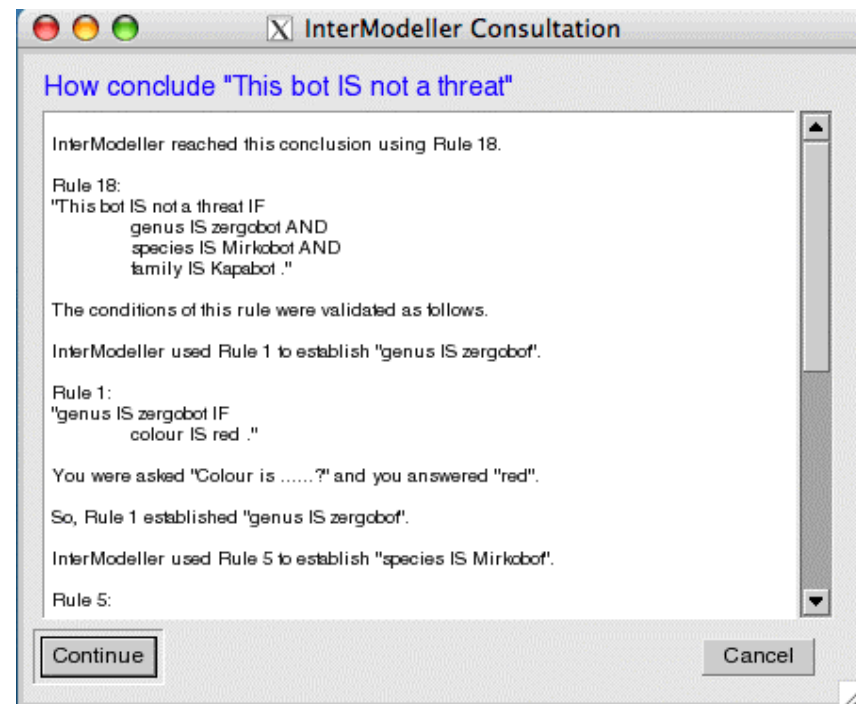
Justification allows the programmer of an expert system to test for faults in the system before it is made publicly available.

It can pinpoint an unexpected course of events for instance when considering the how justification.

It also allows users of the system peace of mind as they can see exactly how a particular conclusion is arrived at.



**Why Justification**



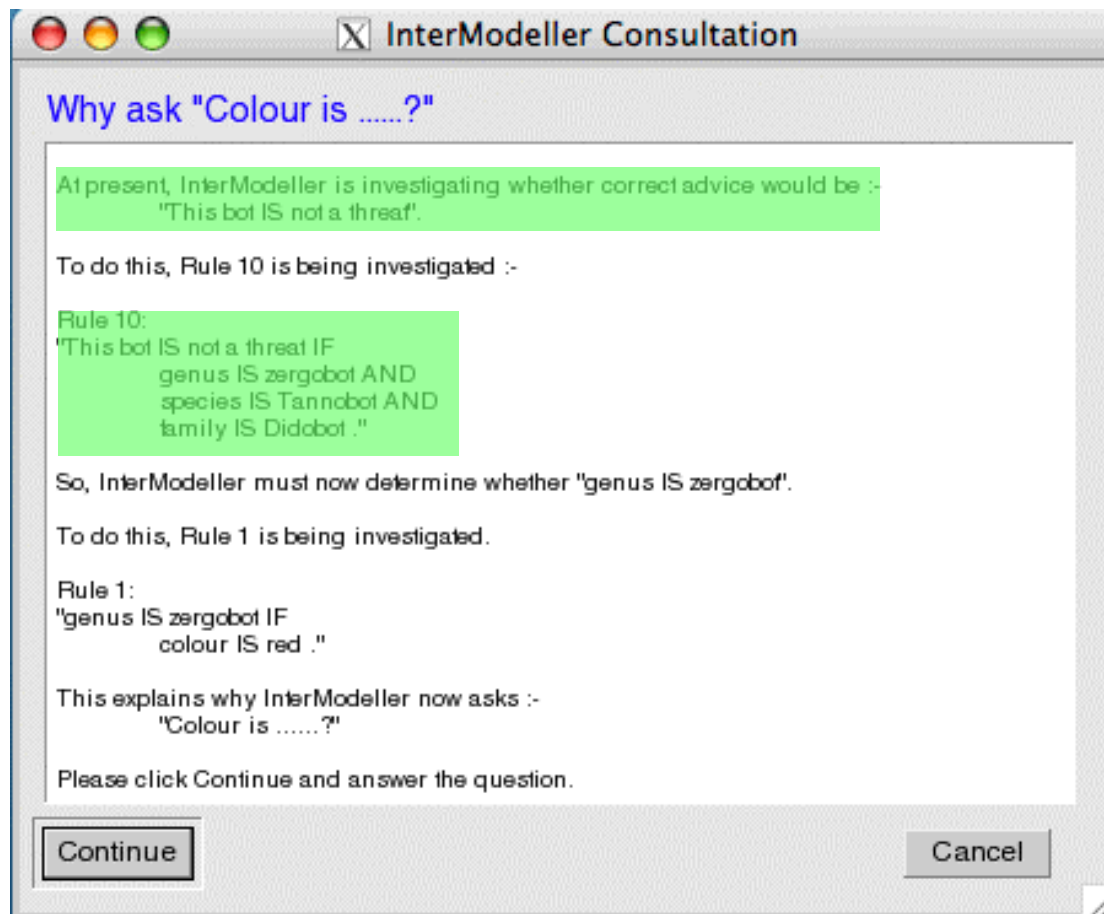
**How Justification**

# Expert System Shells

Consider the why justification shown from the first question asked in a consultation.

The system is trying to provide the advice provided in rule 10 (rather than rules 1, 2, 3 etc).

Which part of the expert system is responsible in making the decision to seek answers justifying rule 10's advice?



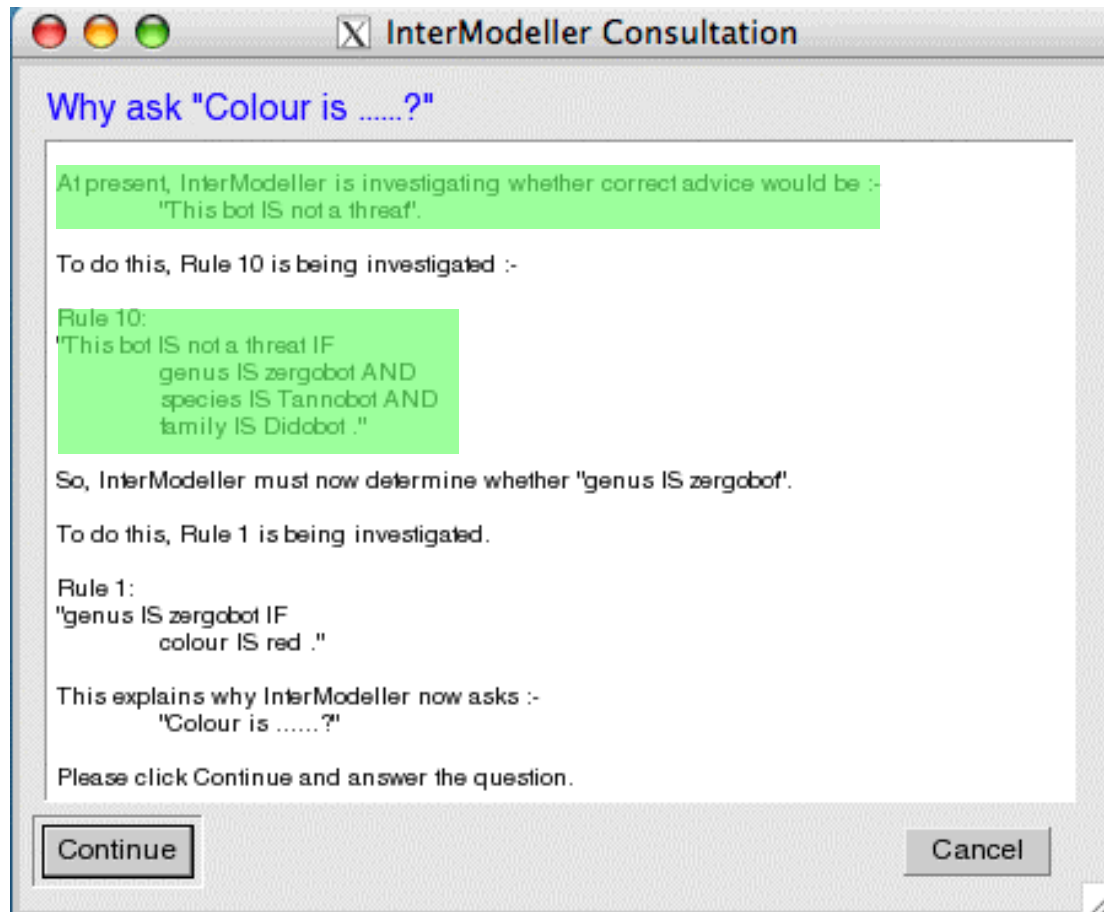
The role of the **inference engine** is to decide how to apply the facts to the rules and decide the order that questions are asked in the user interface.

# Expert System Shells - Methods of Inferencing

Expert systems have 2 ways of inferencing.

**Forward chaining** - Where the system gathers facts from the user and only draws conclusions once enough facts have been collected to identify a conclusion.

**Backward chaining** - Where the system starts with a potential conclusion and works backwards to find evidence that justifies this conclusion.



In the example shown, it would appear that the system uses **backward chaining**, as the system is trying to prove that the bot is not a threat before any questions have been asked (and therefore no facts are known).