2013

STUART Expert System Shell User Manual

Stuart Ednie, Braedon Carter, Ryan McKercher, Willim Olding, Leo kerkslake.
UTAS

1/1/2013

Introduction

Thank you for purchasing the STUART inference engine, you have made the first step towards simplifying your interactions with intelligent systems. STUART stands from (System To Utilise Artificial Reasoning Techniques). The STUART can help you quickly and easily create expert system for virtually any purpose.

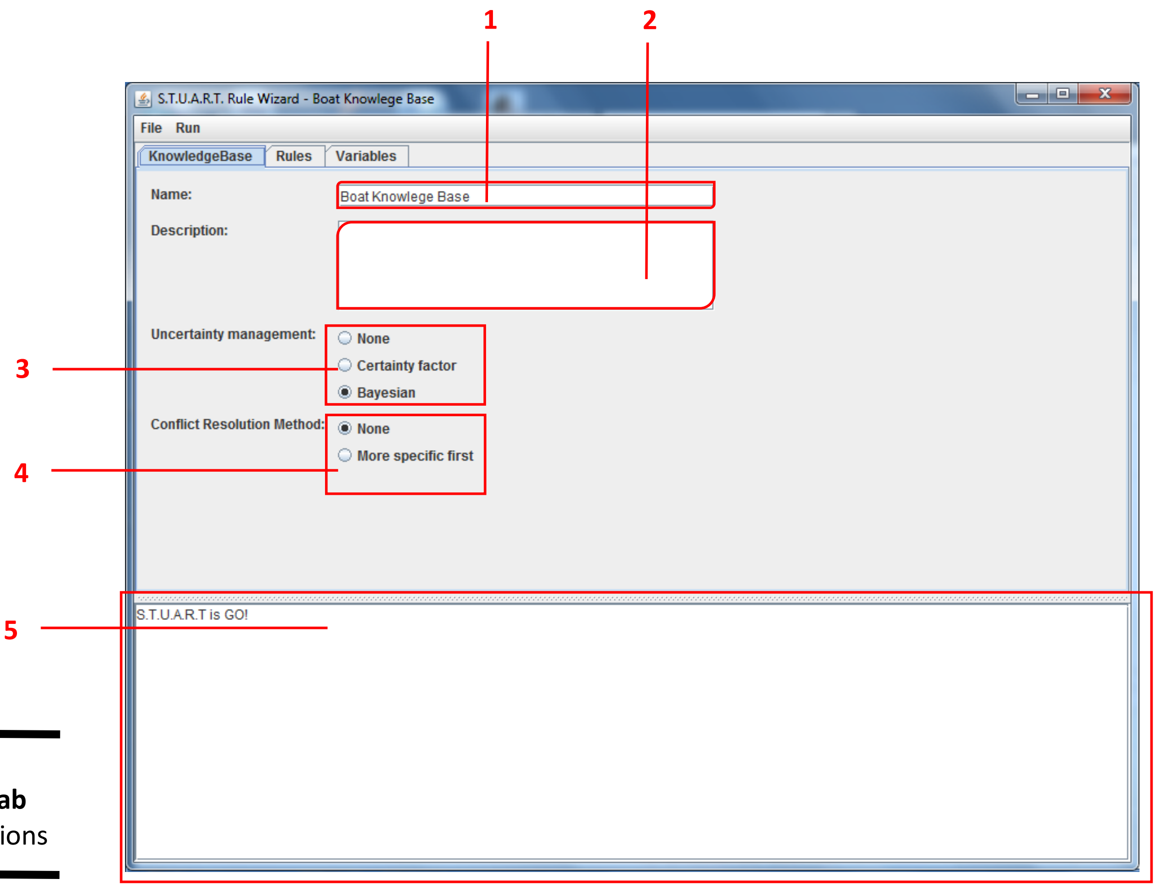
The STUART expert system shell can utilise either forward or backward changing algorithms to reach a conclusion. This can be done with no uncertainty management, using certainly factors or Bayesian reasoning.

Considerably complex expert systems can be created with the STUART and systems can be run within the program without the user needing to interact with the knowledge base.

This manual will explain the various functions of the STUART rule editing and creation software, followed by a simple traffic light problem as a worked example. The actual algorithms and technical documentation are also given at the end in appendix one.

Knowledge base

The first of the three tabs in S.T.U.A.R.T allows the Knowledge Engineer to make decisions affect the whole of the current Knowledge Base.



Name: Boat Knowlege Base

Description:

Uncertainty management: None / Certainty factor / [✓ Bayesian]

Conflict Resolution Method: [✓ None] / More specific first

S.T.U.A.R.T is GO!

Figure
Knowledge Tab
Field descriptions

1. ***Name***; this field allows the current knowledge base to be named. This name will be the default file name when saving the knowledge base.

2. ***Description***; this field is a text field that has no effect on the operation of the knowledge base. Useful information about the current knowledge base can be included here if desired, alternatively it may be left blank.

3. ***Uncertainty management***; The Knowledge Engineer has three options for dealing with uncertainty.

    a. ***None;*** Rule antecedents are evaluated as;
        i. either True or False (no confidence values required by the Operator),
        ii. definitive with the result of the rule evaluation providing only one definitive value (the probability that the value is correct is 100%).

    b. ***Certainty factor***; Rules are evaluated with Certainty factors given by the Operator. When the Operator is prompted for information they will also be prompted to provide an estimate of the confidence they have in the answer they provided.

c. ***Bayesian***; Rules are evaluated using Bayesian probability. The Knowledge Engineer assigns 'Logical Sufficiency' (LS), 'Logical Necessity'(LN) and 'Prior Probability' values when creating a rule.

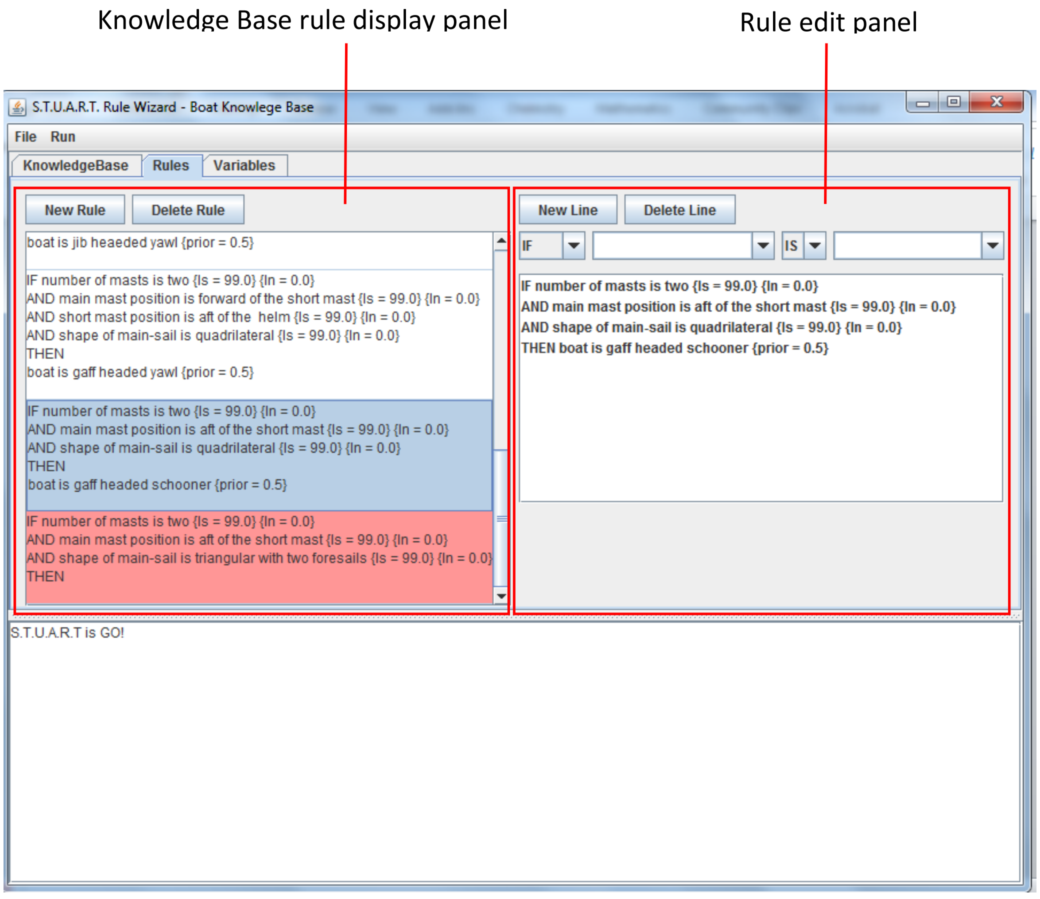
4. ***Conflict resolution***; Conflict resolution strategies seek to resolve apparently conflicting rules that may appear in the Knowledge base. The default value is '***None***', that is, rules will be evaluated in cyclical order from top to bottom, changing this to '***More specific first***' allows a rule conflict to be resolved by using the more specific rule i.e. the rule with the greater number of antecedents

Only advanced users, aware of the system algorithms, are advised to alter this default setting.

5. ***Console***; the final field shows all outputs from the S.T.U.A.R.T. This can be the result of running the program, displaying a summary etc. Note that this window is permanent and not tab dependant.

Rules Tab

The second of the three tabs in S.T.U.A.R.T allows the Knowledge Engineer to create and edit rules. The Rules Tab is divided into two main areas.



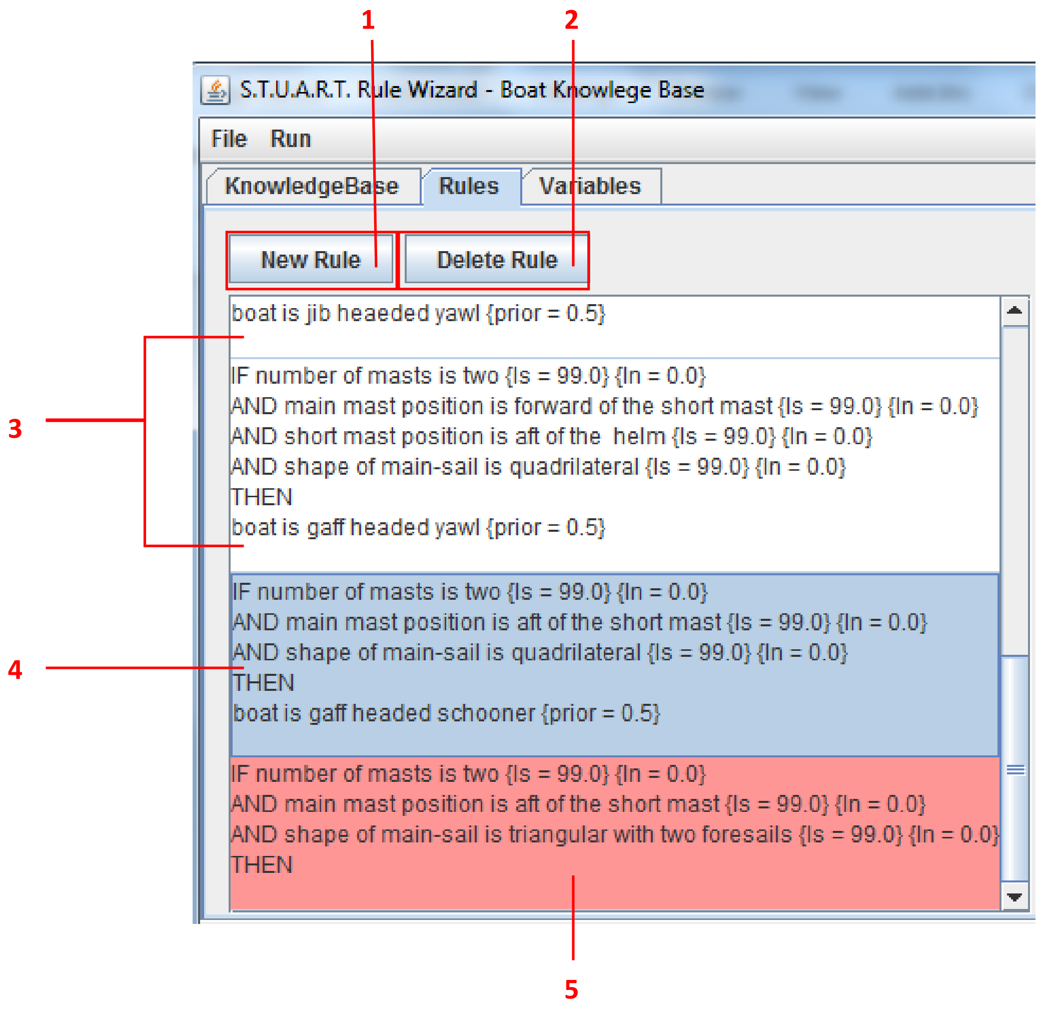
```
boat is jib heaeded yawl {prior = 0.5}

IF number of masts is two {ls = 99.0} {ln = 0.0}
AND main mast position is forward of the short mast {ls = 99.0} {ln = 0.0}
AND short mast position is aft of the  helm {ls = 99.0} {ln = 0.0}
AND shape of main-sail is quadrilateral {ls = 99.0} {ln = 0.0}
THEN
boat is gaff headed yawl {prior = 0.5}

IF number of masts is two {ls = 99.0} {ln = 0.0}
AND main mast position is aft of the short mast {ls = 99.0} {ln = 0.0}
AND shape of main-sail is quadrilateral {ls = 99.0} {ln = 0.0}
THEN
boat is gaff headed schooner {prior = 0.5}

IF number of masts is two {ls = 99.0} {ln = 0.0}
AND main mast position is aft of the short mast {ls = 99.0} {ln = 0.0}
AND shape of main-sail is triangular with two foresails {ls = 99.0} {ln = 0.0}
THEN
```

```
IF number of masts is two {ls = 99.0} {ln = 0.0}
AND main mast position is aft of the short mast {ls = 99.0} {ln = 0.0}
AND shape of main-sail is quadrilateral {ls = 99.0} {ln = 0.0}
THEN boat is gaff headed schooner {prior = 0.5}
```

```
S.T.U.A.R.T is GO!
```

Figure
Rules Tab
Field descriptions

Knowledge Base rule display panel. This panel is used to display the rules in the current knowledge base. Additional rule operations are detailed below.



boat is jib heaeded yawl {prior = 0.5}

IF number of masts is two {Is = 99.0} {In = 0.0}
AND main mast position is forward of the short mast {Is = 99.0} {In = 0.0}
AND short mast position is aft of the helm {Is = 99.0} {In = 0.0}
AND shape of main-sail is quadrilateral {Is = 99.0} {In = 0.0}
THEN
boat is gaff headed yawl {prior = 0.5}

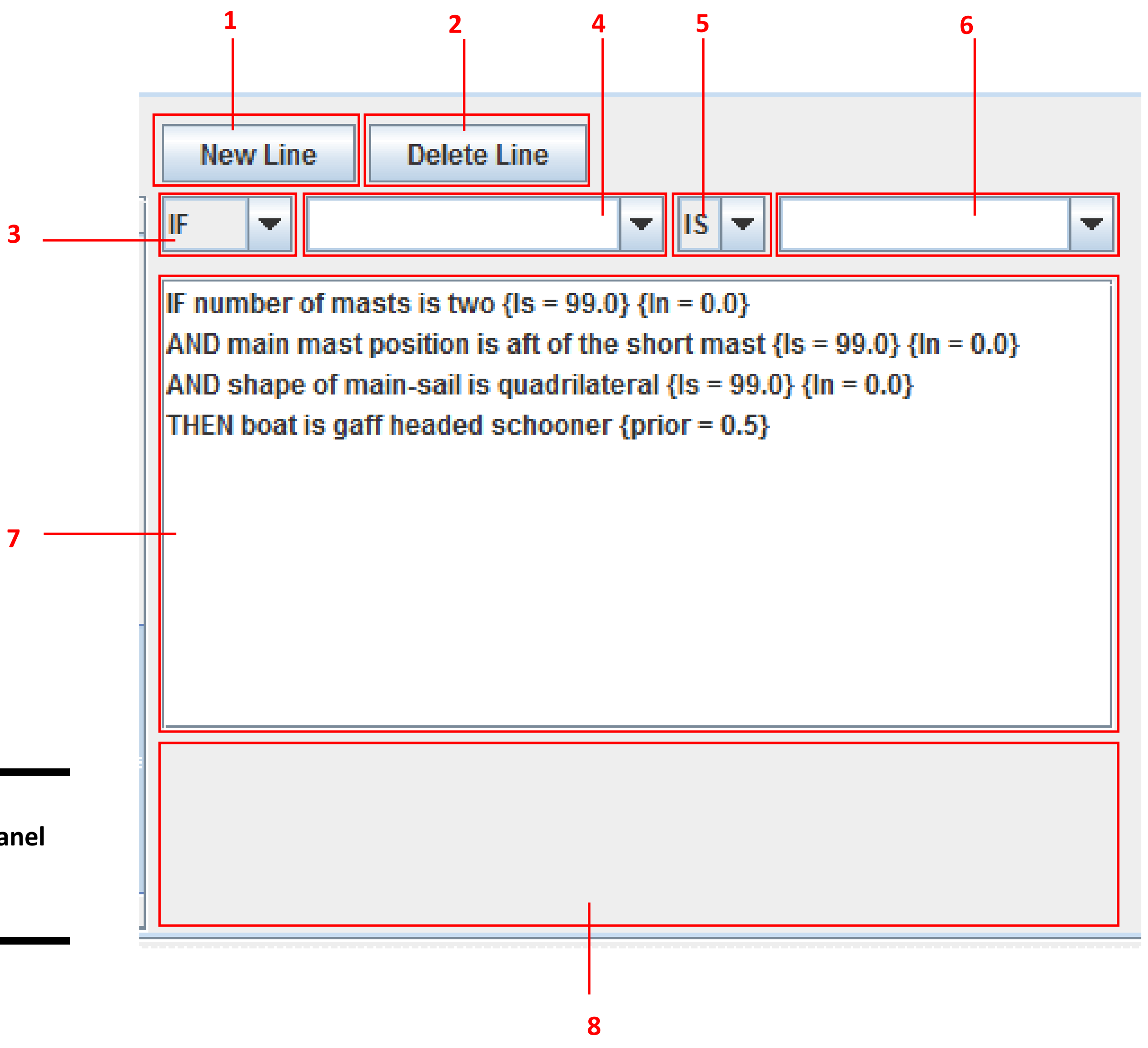
IF number of masts is two {Is = 99.0} {In = 0.0}
AND main mast position is aft of the short mast {Is = 99.0} {In = 0.0}
AND shape of main-sail is quadrilateral {Is = 99.0} {In = 0.0}
THEN
boat is gaff headed schooner {prior = 0.5}

IF number of masts is two {Is = 99.0} {In = 0.0}
AND main mast position is aft of the short mast {Is = 99.0} {In = 0.0}
AND shape of main-sail is triangular with two foresails {Is = 99.0} {In = 0.0}
THEN

Figure **Knowledge Base rule display panel** Field descriptions

1. ***New Rule***; New (blank) rules are added to the Knowledge Base when this button is selected(*clicked*).
2. ***Delete Rule***; The highlighted existing rules in the Knowledge Base will be deleted from the Knowledge Base when this button is selected (*clicked*).
3. Existing rules appear in this window.
4. *Current Rule*; When a rule from the Knowledge Base is selected the entire rule is highlighted BLUE and appears in the Rule Edit panel.
5. Rule error; A rule appears highlighted in red when it lacks a vital element. The rule in Figure xx, for example, has been highlighted because it lacks a valid ‘Then’ statement.

Rule edit panel. This panel allows the *current rule*, the rule selected in the ***Knowledge Base rule display panel***, to be edited.



IF number of masts is two {Is = 99.0} {In = 0.0}
AND main mast position is aft of the short mast {Is = 99.0} {In = 0.0}
AND shape of main-sail is quadrilateral {Is = 99.0} {In = 0.0}
THEN boat is gaff headed schooner {prior = 0.5}

Figure
Rules Edit panel
Field descriptions

1. ***New Line***; This button always a new line to be added to the rule. The added line may be an antecedent or a consequence.

2. ***Delete Line***; this button allows a Knowledge Engineer to remove a line from the *current rule*. The line to be removed is first highlighted then the ***Delete Line*** button is selected (*clicked*) to delete the line. Once deleted the line is not recoverable.

Rule Structure; Each rule;

a. must contain an IF conditional statement on the first line,
b. may contain additional conditional statements of the AND/OR type
c. AND/OR type condition statements are not permitted to be mixed
d. must have at least one consequence (THEN) statement
e. may contain additional consequence (THEN) statements
f. must have its conditional statement/s preceding the consequence statement/s

3. ***Antecedent/Consequence assignment***; a line's conditional/consequence type is assigned from the dropdown box. Any assignment must comply with the previously stated rule structure above.

4. ***Object field;*** this field contains the object of the antecedent/consequence present in the line of the rule. Objects are text 'strings' which may be any text as they are not evaluated mathematically or with a Boolean method. Once an object has been defined, by use in a previous rule, it will appear in the dropdown list for ease of entering future antecedences/consequences.

5. ***Mathematical/Boolean operator;*** one of several operators may be assigned in this field, (***IS, =, !=, >, <, <=, >=***).

   In the case of an ***Antecedent***, the existing Value of the Object in the particular line is compared, using the assigned Mathematical/Boolean operator, to the value present in the line.

   e.g. **IF Temperature [C°] >= 50,**

   this antecedent is true if the Object 'Temperature [C°]' has a value of 50 or more.

   For a ***Consequence***, if the rule is Fired, the Object's value becomes that of the Value in the line.

   e.g. ***THEN COOLING* IS *ON***,

   If the rule is fired the *Object* 'COOLING' will be assigned the *Value* of ***ON***.

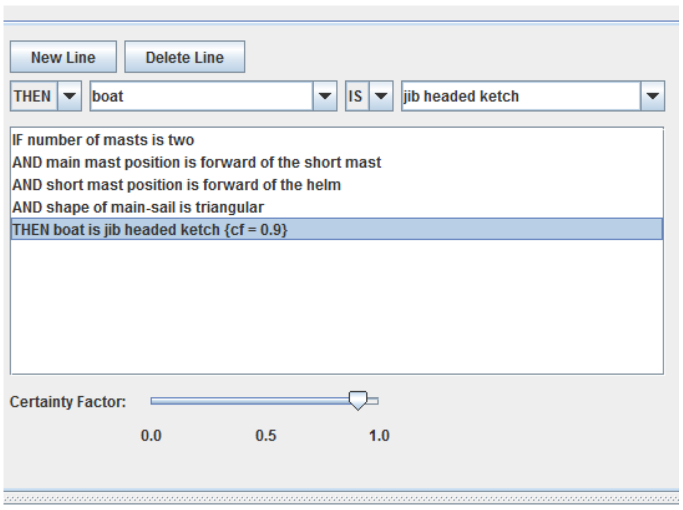
6. ***Value***; this field contains the **Value** to be compared to the Object's value or in the case of a Consequence, the Value to be assigned to the Object.

7. ***Rule window***; the current rule (new or existing) is displayed here.

8. ***Probability/Confidence assignment***; Depending on the type of ***Uncertainty management***, selected in the ***Knowledge Base*** tab, differing probability entries can be made.

   a. ***None***; this area is left blank as indicated in Figure xxxx

   b. ***Certainty factor***; if the Certainty factor radio button is selected, area 7 of Figure XXXX will remain blank when an Antecedent line is selected. However if the/a Consequence line is selected area 7 of Figure XXXX will display a slider for the input of the current rule Certainty factor.

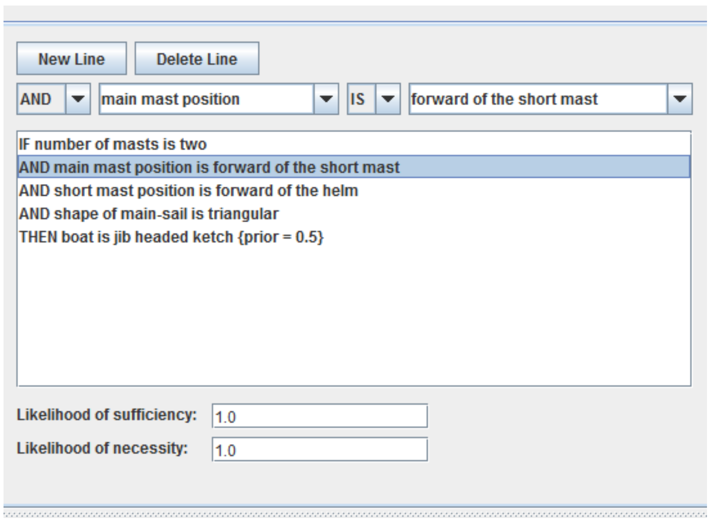


IF number of masts is two
AND main mast position is forward of the short mast
AND short mast position is forward of the helm
AND shape of main-sail is triangular
THEN boat is jib headed ketch {cf = 0.9}

Figure **Certainty Factor** Slider

c. ***Bayesian;*** if the Bayesian radio button is selected and an Antecedent is also selected from the current rule, area 7 of Figure XXXX will display fields for the entry of;

   i. Likelihood of sufficiency (LS) and
   ii. Likelihood of necessity (LN)



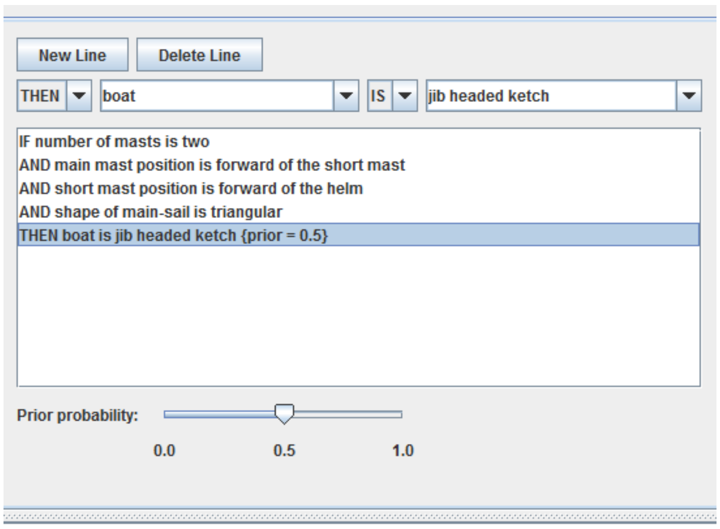
IF number of masts is two
AND main mast position is forward of the short mast
AND short mast position is forward of the helm
AND shape of main-sail is triangular
THEN boat is jib headed ketch {prior = 0.5}

Likelihood of sufficiency: 1.0
Likelihood of necessity: 1.0

Figure **Bayesian** LS and LN Sliders

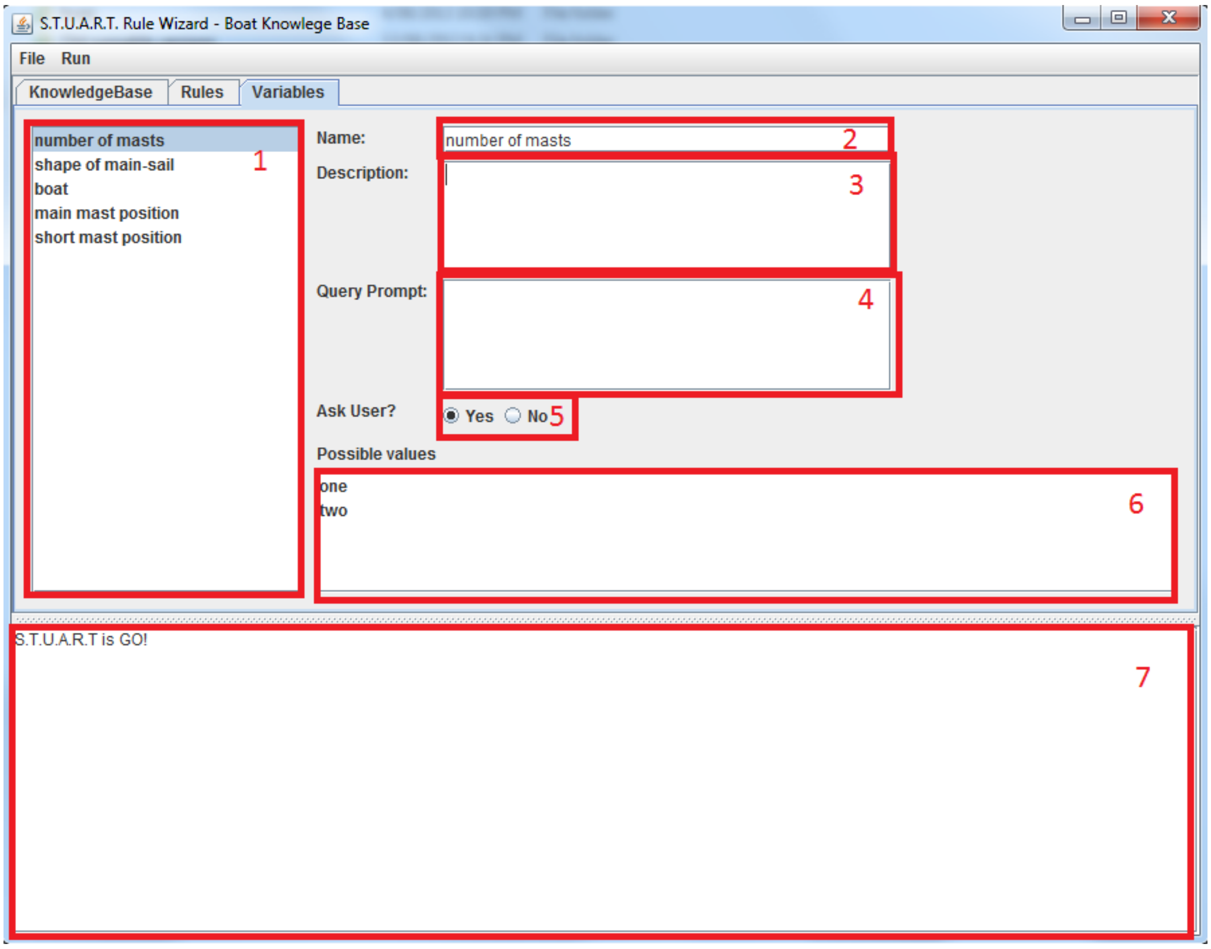
However if the/a *Consequence* line is selected area 7 of Figure XXXX will display a slider for the input of the current rule ***Prior probability***.

Figure
Bayesian
Prior probability slider



IF number of masts is two
AND main mast position is forward of the short mast
AND short mast position is forward of the helm
AND shape of main-sail is triangular
THEN boat is jib headed ketch {prior = 0.5}

Variables Tab


S.T.U.A.R.T. Rule Wizard - Boat Knowlege Base

File Run

KnowledgeBase | Rules | Variables

number of masts
shape of main-sail
boat
main mast position
short mast position

Name: number of masts

Description:

Query Prompt:

Ask User? [✓ Yes] No

Possible values
one
two

S.T.U.A.R.T is GO!

1. Variable List; This field lists all variables (antecedents and consequents) that have been defined by the user while writing rules in the Rules tab. You cannot add new variables directly to this list, it exists for the sol purpose of selecting which variable is currently selected (highlighted blue).

2. Name; this field is un-editable and displays the name of the currently selected variable.

3. Description; This field can be left black by the user if desired. It provides a location where as description of the type or purpose of a variable can by typed. This could be used effectively as comment space.

4. Query prompt; this is an optional field where the user can specify the question that will be asked when and if the system decides to query the user over a variable. This field is not compulsory and if left blank a default question "Input value of…." Will be used.

5. Ask user; this if where the user can specify whether the program should quire the user about an unknown variable or wait until it is defined from another rule. If this field is set incorrectly in such a way that the program is un-runnable the theses will be set to “best guess” values by the code. However setting them correctly the first time is highly recommended.

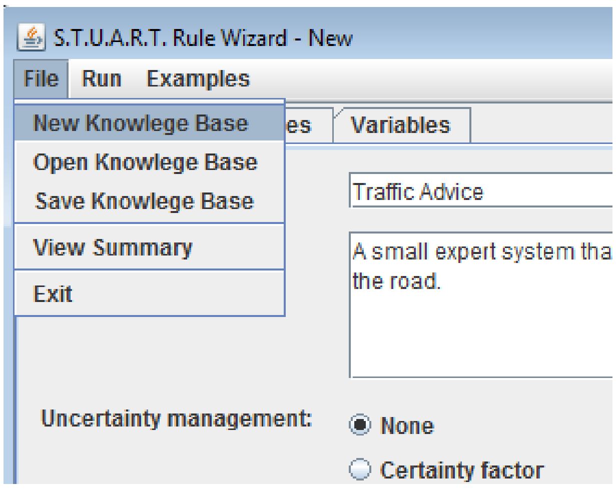
6. Possible values; here the possible values of the currently selected variable are displayed, this is only included for troubleshooting and examination purposes. In most cased this field can be ignored.

7. Console; this final field of tab shows all output from the S.T.U.A.R.T. This can be the result of running the program, displaying a summary etc. Note that this window is permanent and not tab dependant.

Worked example

Here we will work through the example of creating an expert system designed to tell someone when to cross the road.

First open the STUART ESS (Expert system shell) and create a new knowledge base by clicking File -> New Knowledge base



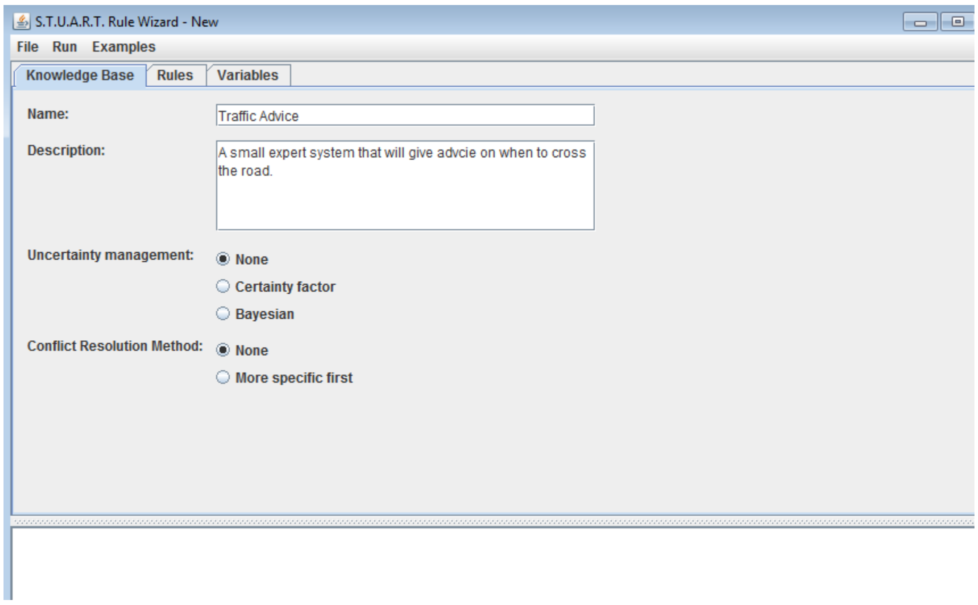
Traffic Advice

A small expert system tha
the road.

Uncertainty management: [✓ None] Certainty factor

Once the new knowledge base has been created you should now populate the global parameters.

First the knowledge base should be given a name, this is not important and only for user reference. Next the method of uncertainty management and conflict resolution methods need be chosen. For this example no uncertainty management and no conflict resolution will be used. Once this has been done the screen should look as follows, note a system description can also be entered.



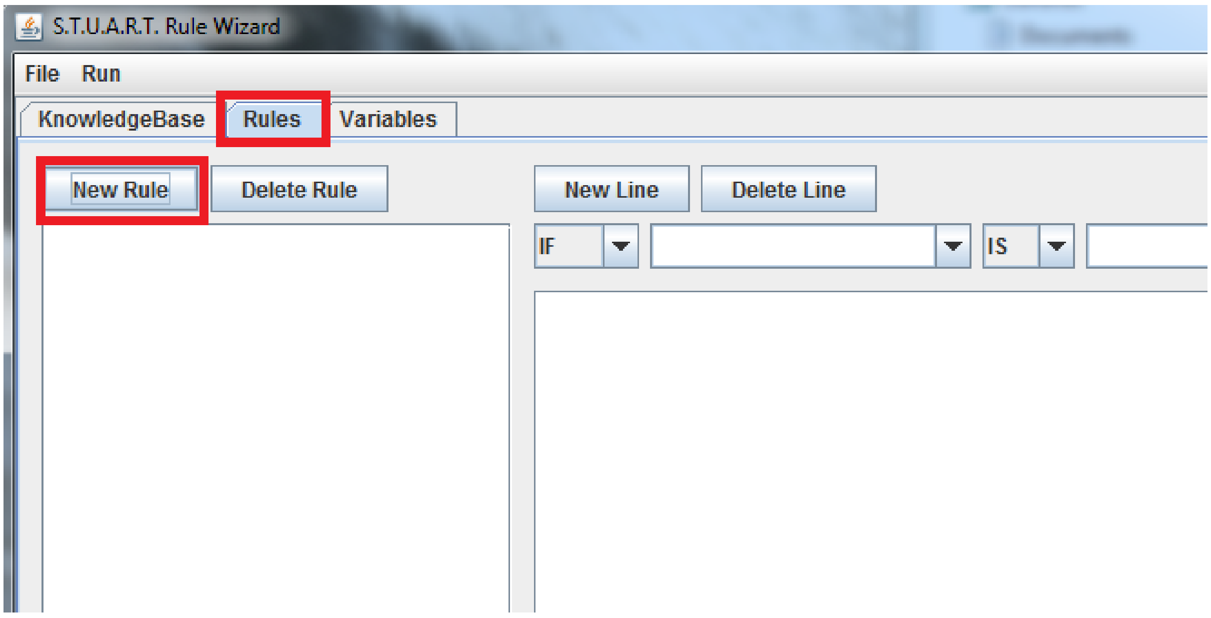
Name: Traffic Advice

Description: A small expert system that will give advcie on when to cross the road.

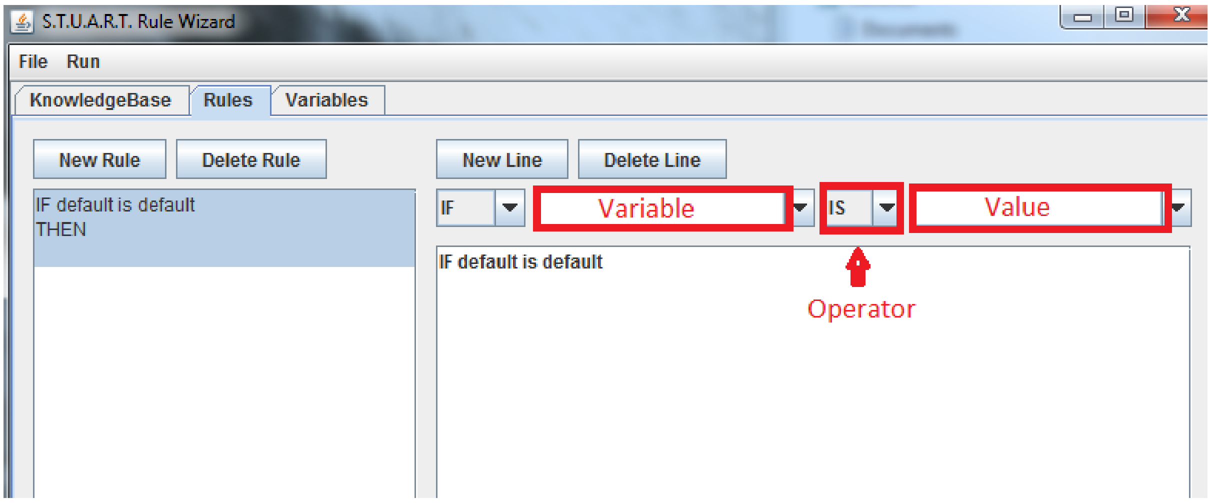
Uncertainty management: [✓ None] Certainty factor Bayesian

Conflict Resolution Method: [✓ None] More specific first

Now that the global data has been set the Rules can now be written, first select the rules tab. It should look as follows.



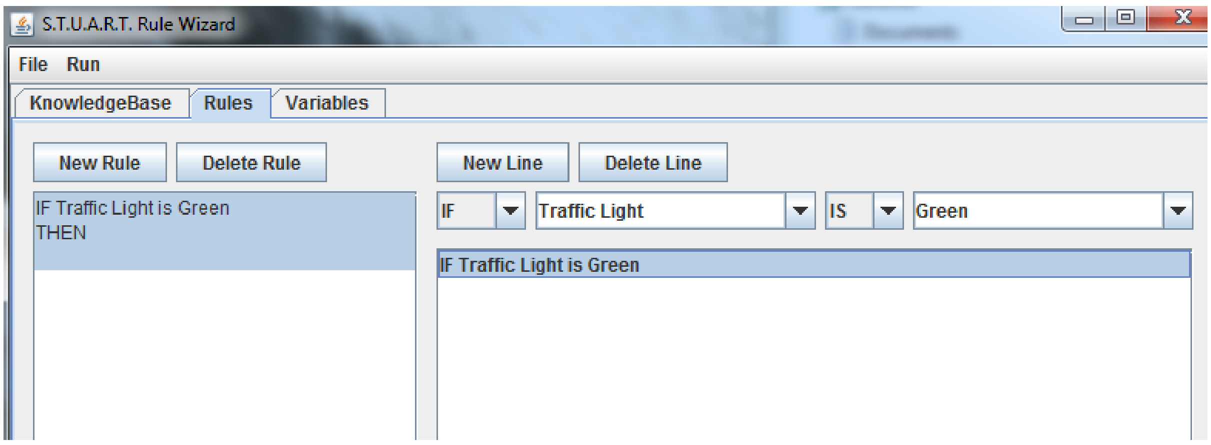
To add the first rule click the new rule button, this will also add the first line of the new rule.



IF default is default
THEN

IF default is default

You can now enter the Variable, and value for each line. To begin with, enter the line "If Light is green". The completed line should now look as shown below.



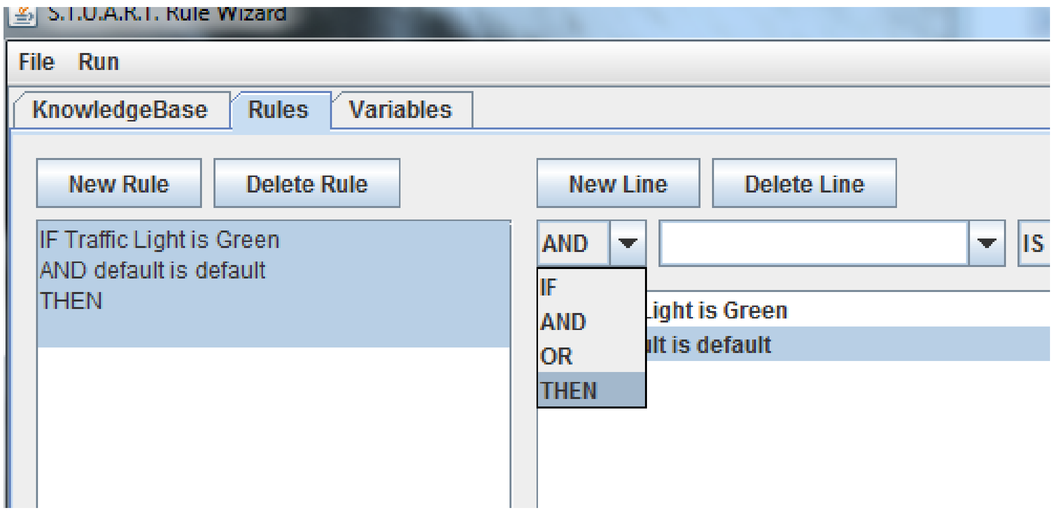
IF Traffic Light is Green
THEN

IF | Traffic Light | IS | Green

IF Traffic Light is Green

Now click new line to enter the next line in the current rule.

For the next line we will need to change the function from AND, which is the default for the second line of any rule, to THEN, which is the concluding statement on any rule. This is done from the left hand drop down list, as shown below.

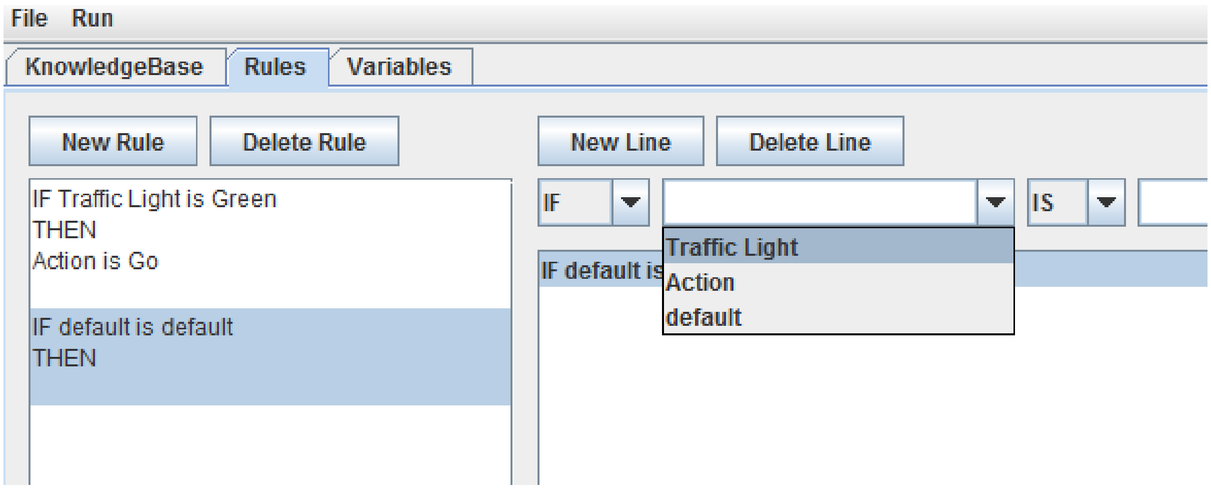


IF Traffic Light is Green
AND default is default
THEN

AND | | IS

Once the is selected we can enter the consequent “Then action is go”.

Now we will enter the next rule in much the same manner as the first one except we do not need to type the variable “traffic light” as is now appears in the drop down list. This is shown below.



IF Traffic Light is Green
THEN
Action is Go

IF default is default
THEN

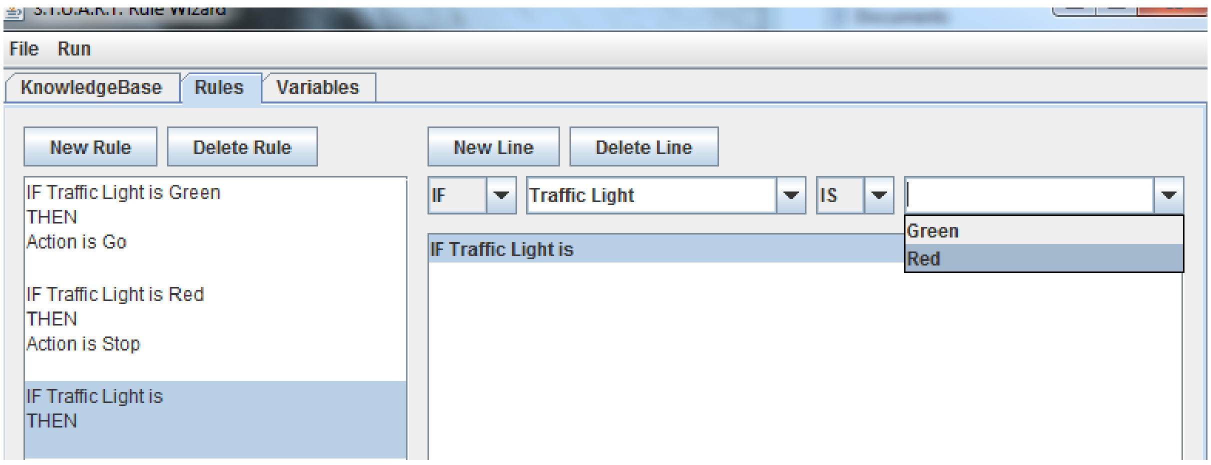
IF default is

Now enter the rest of the rule the same way. The Rule entered here was "If Traffic light is Red THEN action is stop".

Before the program is finished one final rule will be created:

"IF Traffic Light is Red AND Cars are not present AND policemen are not present THEN action is Go"

The First line of this rule is the same as the first line of the second rule, so it can be constructed in much the way using the drop lists. See below



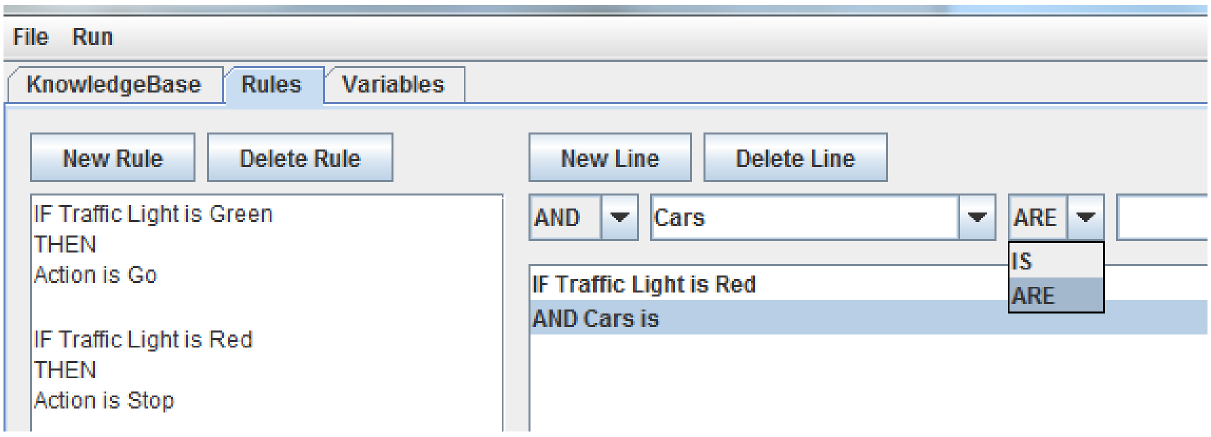
IF Traffic Light is Green
THEN
Action is Go

IF Traffic Light is Red
THEN
Action is Stop

IF Traffic Light is
THEN

IF Traffic Light is

For the next two lines the function will need to be set to AND, as this allows the addition of more antecedents. The operator can also be changed from IS to ARE if the user should prefer. This is only a semantic change and does not affect the program at all. The selection of this is shown below, however the example here has left it as is.

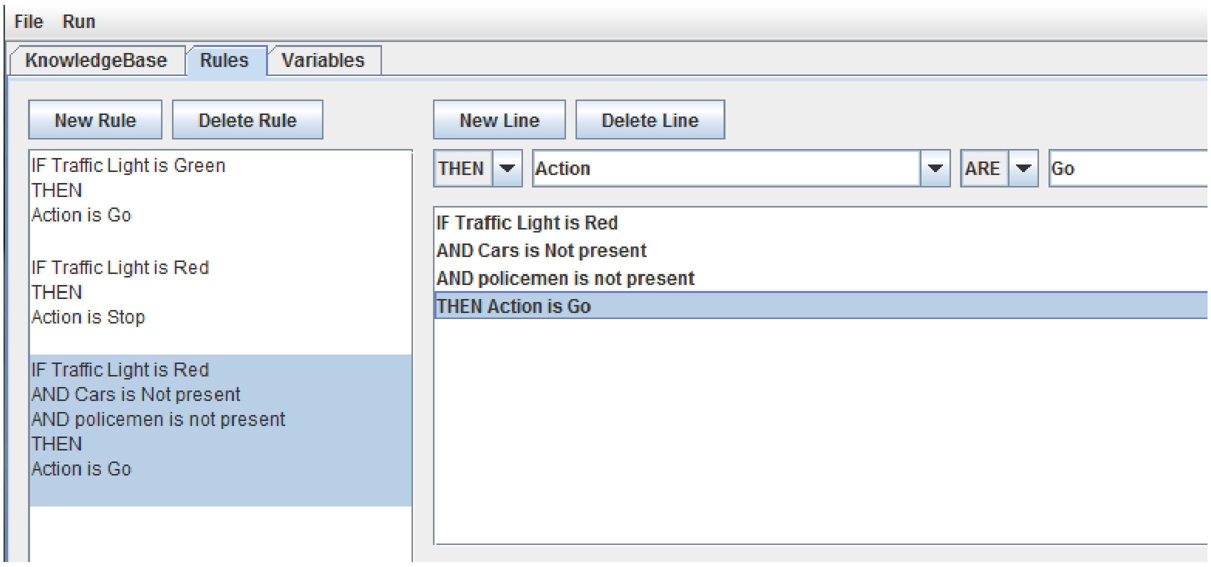


IF Traffic Light is Green
THEN
Action is Go

IF Traffic Light is Red
THEN
Action is Stop

IF Traffic Light is Red
AND Cars is

Once the final rule has been completed the rule base is now finished and should look as shown here;



IF Traffic Light is Green
THEN
Action is Go

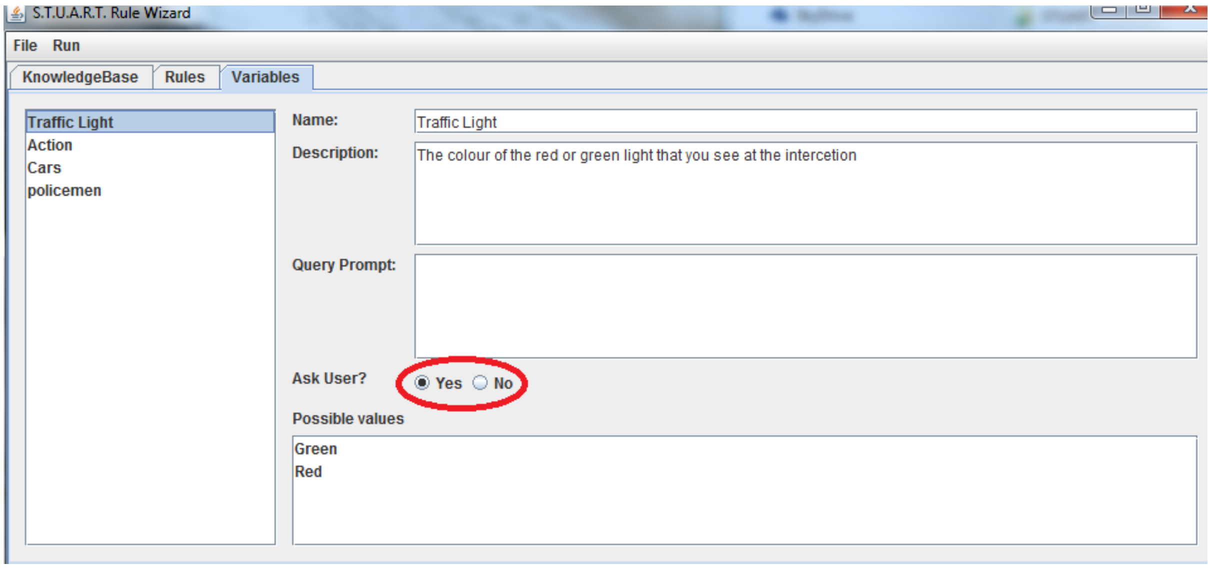
IF Traffic Light is Red
THEN
Action is Stop

IF Traffic Light is Red
AND Cars is Not present
AND policemen is not present
THEN
Action is Go

IF Traffic Light is Red
AND Cars is Not present
AND policemen is not present
THEN Action is Go

Before more work is carried out the current ESS should be saved, this can be accomplished by clicking File -> Save Knowledge base.

Before the expert system can be run the nature of each variable needs to be defined in the program. This is done in the variables tab. Once in the variables tab you will see a list of variables that you have used in your rules. Here you can enter a variable description, write a Query prompt and view a list of the variables possible values as defined by your rules. However you should set the ask user option so that only data you intend to gather from the operator is queried, although if you fill this in incorrectly the program will assume user inputs and most of the time run correctly.



Traffic Light
Action
Cars
policemen

Name: Traffic Light

Description: The colour of the red or green light that you see at the intercetion

Query Prompt:

Ask User? [✓ Yes] No

Possible values
Green
Red

Lastly none of the fields in the tab need be filled in, but should be seen as housekeeping tasks that make the database more complete.

Lastly, save the data base before trying to run it.