

SOPHIEs made easy: Part 6



Dr Robert Treharne Jones

GP,
Torquay

Welcome back, and a special greeting to those gritty individuals who have stuck to the course so far! I hope that the earlier work has paid off and you are beginning to see some glimmer of light at the end of the tunnel. I'm also pleased to get positive e-mail responses to earlier parts of the course, but if your experience is still on the thin side then I really would encourage you to go back to the point at which you may have 'lost the plot' and start again! Just to repeat the underlying message, you really have to grasp the fundamentals before you can move on.

Last time we dealt with the intricacies of the \$FETCH question and I posed a puzzling question right at the end of the article. The editorial offices have not been overwhelmed by the postman staggering under the weight of the bag containing your answers so I shall start by putting you out of your misery!

The puzzle, as you may remember, was based on the schematic SOPHIE:

1. \$AGE	Less than 25	→	GOTO Q.2
	More than 25	→	GOTO Q.3
2. \$FETCH	Find height within last 12 months		
	YES	- calculate patient's BMI using values	
	NO	- prompt user to measure patient's height	
3. \$FETCH	Find height within last 10 years		
	YES	- calculate patient's BMI using values	
	NO	- prompt user to measure patient's height	

On the face of it this seems quite straightforward. Most people stop growing, vertically at least, by age 25. Before that age it seems appropriate to check for a height value within the last year, while for fully mature adults a ten-year look-back should suffice.

But what about the adult aged, for instance, 26 years? This SOPHIE stands a chance of checking back and using a height value from when this individual was aged 16, which is likely to be inappropriate, especially if the pubertal growth spurt was incomplete.

The answer is either to have 35 years as the cut-off point to start the ten-year look-back or, more elegantly, include an intermediate branch on the age question to cover those patients who

are just on the transition. The end result might then look like this:

1. \$AGE	Less than 25	→	GOTO Q.2
	Between 25 and 35	→	GOTO Q.3
	More than 35	→	GOTO Q.4
2. \$FETCH	Find height within last 12 months		
	YES	- calculate patient's BMI using values	
	NO	- prompt user to measure patient's height	
3. \$FETCH	Find height within last 3 years		
	YES	- calculate patient's BMI using values	
	NO	- prompt user to measure patient's height	
4. \$FETCH	Find height within last 10 years		
	YES	- calculate patient's BMI using values	
	NO	- prompt user to measure patient's height	

The value I have used here is entirely arbitrary and you may have your own ideas about which values you would like to use for your own guidelines. The important principle to be learned here has repercussions for other SOPHIE authors—unless you are careful it is possible for some patients to 'fall down the hole' and the guideline won't allow them to follow the flow chart as you had intended. We shall come across more examples in the main topic for this issue which is the \$VAL question¹.

'Hang on a moment,' I hear you say, 'you promised us a feature on the \$CALC question this time!' Yes, I did say that, such was my eagerness to continue the database questions in full—but it wouldn't be right to tackle \$CALC before we had experienced the rite of passage through the wonders of \$VAL. **\$VAL** This question differs from all those others so far, because it requires input from the

user—a numerical value, in fact! We haven't yet completed the database questions but \$CALC, which I had hoped to discuss, is a database question which manipulates values, and it would be confusing to do things the other way round.

Let's click on the 'Value' icon on the palette, then drag the object open in the usual way and this is what we get (Fig. 1), overleaf.

Don't ask me why there's a figure '30' included in the icon but no doubt some Torex person will be able to tell us one day. The important thing is to double click on the icon in the top left hand corner of the object to open up a window like Fig. 2.

You now see two tabs open up at the

top of the new window, labelled 'Prompt' and 'Values', and the default view brings you straight into the 'Prompt' tab, which itself contains two fields.

The 'Prompt' field is where you enter the words that you want the user to read at this stage in your guideline, so for this reason it's compulsory to enter something. Let's imagine we are including some questions on alcohol consumption as part of a screening guideline; we

might enter the words 'Pints of beer per week?' at this point.

In the Help field, which is optional, it is useful to enter some text to indicate to the novice user exactly what the question is all about. You know what it's all about because you wrote it, but not all the rest of your team may think the way you do, so supportive comments are useful. Even though the field is optional it makes good sense to use it whenever you can to explain the question more fully. Something that might fit the bill here is 'Please enter a number to indicate the average number of pints of beer consumed by your patient each week. If they don't drink beer then enter zero'.

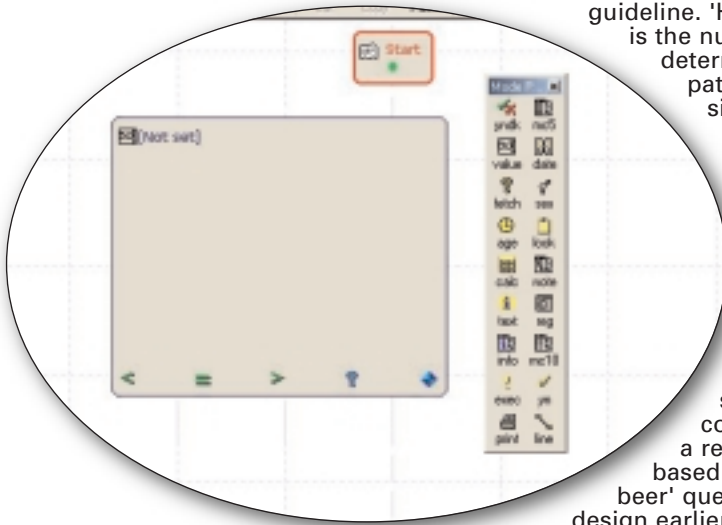
Now click on the 'Values' tab and let's see what happens (Fig. 3).

It looks bit complicated, doesn't it, but never fear, the Wizard is here to guide you delicately through this potential minefield.

This section is simply trying to assess the value you have entered and decide whether that value is appropriate for this particular question. It then gives you the option to save this numerical value to be used in some special way later in the guideline. Next it wants to know where to go, based on the value you have entered.

The field labelled 'High validation limit' and 'Low validation limit' are where you enter the upper and lower numbers to be accepted by this question. Any number which is outside the range you set will be rejected completely by the question. In the case of pints of beer this may not be important because everyone knows that 1000 pints of beer a week is

¹ Interestingly, there's a second issue in the original algorithm, of importance to programmers. Those under 25 go to Q.2; those over 25 to Q.3. But what happens to those who are just 25? The original algorithm question really needs to be rewritten in the form of '25 and over' and '24 and under', which takes care of the > and < output paths from the \$AGE question, leaving the = path redundant.



guideline. 'High decision value' is the number which determines the outward path you take via the > sign, 'Low decision value' determines the path via the < sign, and if the value entered lies between these two values the guideline will take the user along the = path. That all may sound over-complex so let's use a real life example based on the 'pints of beer' question we started to design earlier.

Fig. 1

probably inappropriate, even for a medical student.

But what about a \$VAL question which your receptionists might be using to enter your lab results from the hospital? Supposing they misread the

Let's assume that 28 units of alcohol is currently the upper recommended safe limit for males¹: that equates to 14 pints of beer. So why not put 14 in the 'low decision value' and something like 21 in the 'high decision value'. The net result of this design means that if your patient drinks within the safe recommended limits they might move

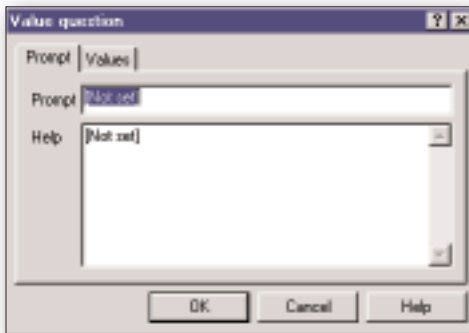


Fig. 2

sodium and potassium values, swapping them round, and have no clinical knowledge to override their mistake? In this sort of instance it makes sense to have fairly strict upper and lower absolute values to set the range, in the certain knowledge that your patient would be dead if they had a potassium greater than 10, for example.

If you read Part 5 of this series in the last issue, all about the \$FETCH question, then the 'Assign to variable' field will hold no fear for you! This is the point at which \$VAL gives you the option of saving the numeric value you enter in answer to the question, and attaching it to a letter of the alphabet for possible use later in the guideline.

In this instance the field defaults to value 'V' (compared with 'F' in the \$FETCH question) but these letters are entirely arbitrary, and the same rules apply. Just to remind you, 'A' must always be used for Age, 'S' for Systolic pressure and 'D' for Diastolic pressure, and although there is no rule against it, in practice 'H' and 'W' are always used for Height and Weight.

So that just leaves us with the two remaining fields which \$VAL will use to determine the next step in the

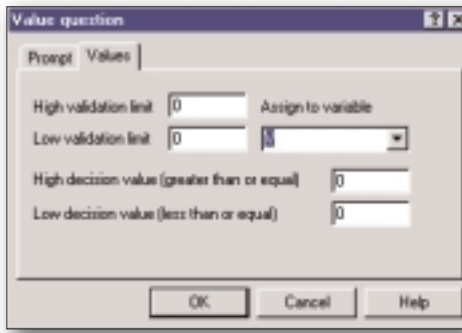


Fig. 3

on swiftly through the rest of the guideline. If they consume more than 14 pints, but less than 21, then they might be given the option of watching their intake, whereas patients who drink more than 21 pints a week might be offered more pertinent advice!

And that's the \$VAL question. \$VAL has to be used whenever a numeric needs to be entered.

Perhaps a few more practical examples might help set the scene.

1. \$VAL	Systolic BP > 160 Systolic BP 140 - 160 Systolic BP < 140	GOTO Q.2 GOTO Q.3 GOTO Q.4
2. \$VAL	Diastolic BP > 100 Diastolic BP 90-100 Diastolic BP < 90	Code BP as HIGH Code BP as HIGH Code BP as HIGH
3. \$VAL	Diastolic BP > 100 Diastolic BP 90-100 Diastolic BP < 90	Code BP as HIGH Code BP as BORDERLINE Code BP as BORDERLINE
4. \$VAL	Diastolic BP > 100 Diastolic BP 90-100 Diastolic BP < 90	Code BP as HIGH Code BP as BORDERLINE Code BP as NORMAL

Fig. 4

One of the first extensive SOPHIEs I ever wrote was a laboratory results guideline to help our receptionists enter results on a daily basis. We still haven't got Path Links in our area so they are still using it ten years later! Basically it's a whole string of \$VAL questions strung together one after the other, although there are some short cuts involving \$CALC which we shall talk about next time.

It's not just lab tests which lend themselves to the \$VAL approach. Smoking habits, heights, weights and blood pressure are all commonly included in guidelines and will need this approach.

One thing you won't use \$VAL for is a patient's age. Why not? Because, as we found two issues ago, the patient's age is stored on the database and the \$AGE question will extract it automatically without the need for any user input.

Let's finish as we started, with a puzzle, but this time I'll explain what's what. Blood pressure, as I have just said, lends itself particularly well to the \$VAL question but of course there are two numerics you have to enter—systolic and diastolic. How do you structure your questions to cater for each one?

The answer is to have two \$VAL questions, one after the other, the first for systolic and the second for diastolic. The essential thing here is that you must save the values in each question using the variables S and D and then the guideline will know exactly what to do with them.

The second thing to remember is that the output paths from the \$VAL question dealing with systolic pressure should contain no Read codes! By all means have different output paths for different systolic BPs pointing to different \$VAL questions for the diastolic pressure, but it is at the diastolic \$VAL question where the coding actually occurs.

Here's another schematic diagram (Fig. 4) to show what I mean (but once again note that the figures I have used are entirely arbitrary).

Next time we really will be looking at the \$CALC question before moving on to the questions which require patient input!

¹ The exact value seems to vary from authority to authority, and from time to time